



ВАРИАНТ 1

Задача 1 (10 баллов)

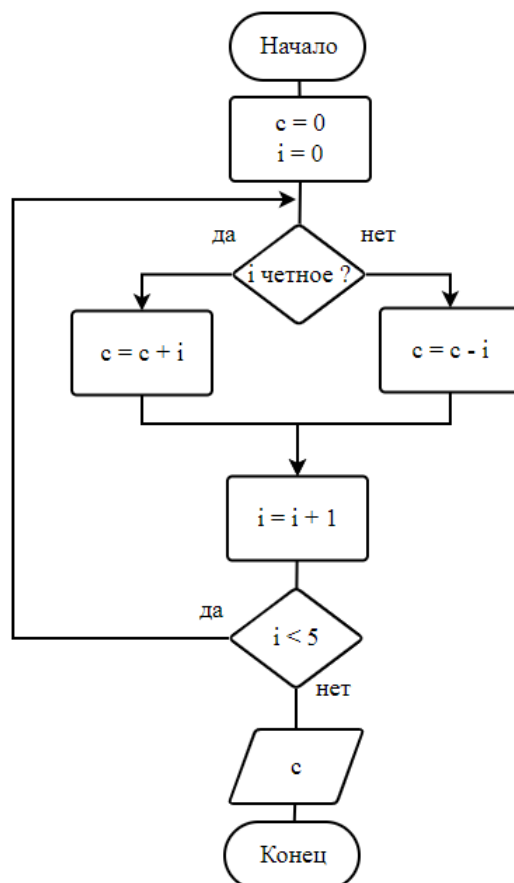
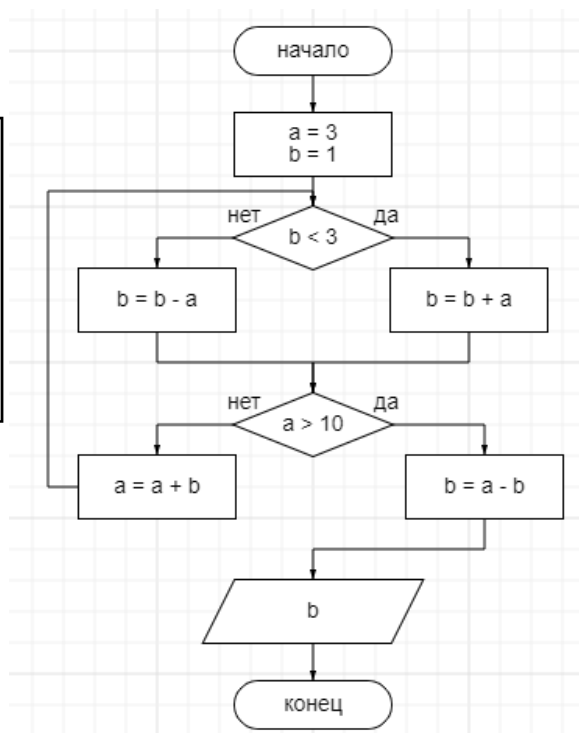
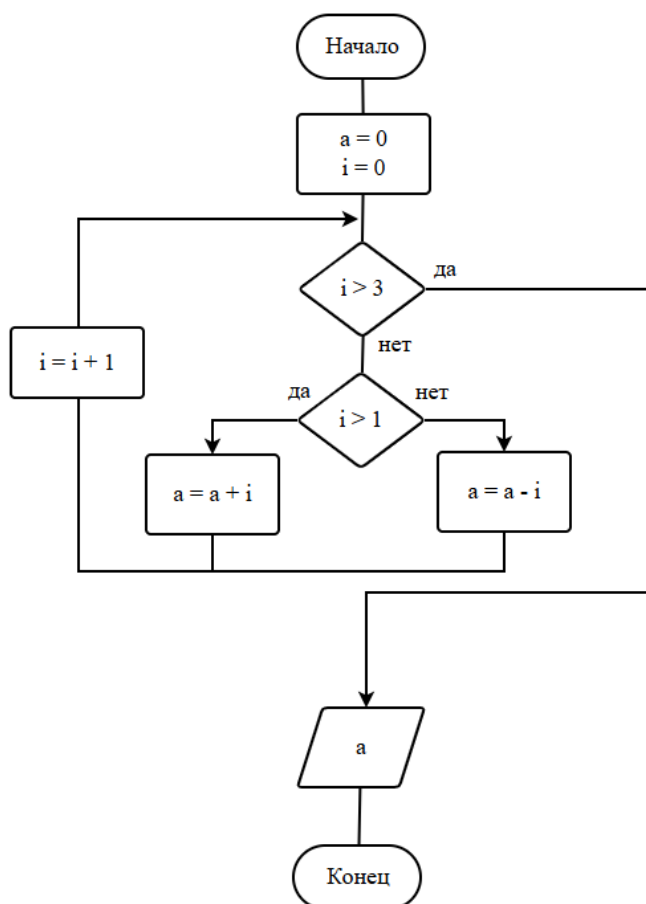
Разведчики

Нашими разведчиками на линии боевого соприкосновения были обнаружены средства противника, сведения о которых они передали в виде закодированной электронной таблицы на командный пункт. Определите количество этих технических средств, значение которых расположены в ячейках B2:F2. В ответе запишите значения ячеек в одну строку, через запятую, начиная с B2 и заканчивая F2.

	A	B	C	D	E	F
1		Танки	БМП	Орудия	Установки РЭБ	Опорные пункты
2						
3						
4	a =				1) =СЧЁТЕСЛИ(B2:F2;B4)	
5	b =				2) =СЧЁТЕСЛИ(E2;B4)+B5	
6	c =				3) =СУММЕСЛИ(D2:E2;B4)-B6	
7	d =				4) =СУММ(D2:F2)-B6	
8					5) =ЕСЛИ(B7>5;B2-C2;СУММ(B2;C2))	
9					6) =ЕСЛИ(B7<=5;B2-C2;СУММ(B2;C2))	
10						

	A	B	C	D	E	F	G
1		Танки	БМП	Орудия	Установки РЭБ	Опорные пункты	
2							
3							
4	a =				1) 2		
5	b =				2) 17		
6	c =				3) 6		
7	d =				4) 8		
8					5) 3		
9					6) -1		
10							

Значения a, b и c определите, выполнив соответствующие алгоритмы по блок-схемам, d (ячейка B7) – неизвестное натуральное число.





Решение

Определяем значения коэффициентов a , b , c выполнив соответствующие алгоритмы по представленным блок-схемам. В результате получаем следующие значения.

	A	B	C
3			
4	a =	4	
5	b =	16	
6	c =	2	
7	d =		
8			

1) Значение формулы в ячейке E4 «=СЧЁТЕСЛИ(B2:F2;B4)» равно 2, следовательно в ячейках B2:F2 находятся две 4.

2) Значение формулы в ячейке E5 «=СЧЁТЕСЛИ(E2;B4)+B5» равно 17, откуда значение «=СЧЁТЕСЛИ(E2;B4)» равно $17 - B5 = 17 - 16 = 1$. Следовательно в ячейке E2 находится 4.

3) Значение формулы в ячейке E6 «=СУММЕСЛИ(D2:E2;B4)-B6» равно 6, откуда значение «=СУММЕСЛИ(D2:E2;B4)» равно $6 + B6 = 6 + 2 = 8$. Следовательно в ячейке D2, также как и в E2, находится значение 4.

4) Значение формулы в ячейке E7 «=СУММ(D2:F2)-B6» равно 8, откуда значение «=СУММ(D2:F2)» равно $8 + B6 = 8 + 2 = 10$. Значения ячеек D2 и E2 известны, находим значение ячейки F2. $F2 = 10 - 4 - 4 = 2$.

5) Анализируя формулы в ячейке E8 «=ЕСЛИ(B7>5;B2-C2;СУММ(B2;C2))» и ячейке E9 «=ЕСЛИ(B7<=5;B2-C2;СУММ(B2;C2))» понимаем, что если значение в ячейке B7 будет больше 5, то в ячейке E8 будет $B2 - C2$, а в ячейке E9: $B2 + C2$, в противном случае в ячейке E8 будет $B2 + C2$, а в ячейке E9: $B2 - C2$. Так как, по условию, значения в ячейках B2:F2 натуральные числа (количество объектов), то разность всегда меньше суммы, получаем:

$$\begin{cases} B2 + C2 = 3 \\ B2 - C2 = -1 \end{cases} \Rightarrow \begin{cases} C2 - 1 + C2 = 3 \\ B2 = C2 - 1 \end{cases} \Rightarrow \begin{cases} C2 = 2 \\ B2 = 1 \end{cases}$$

Ответ: 1, 2, 4, 4, 2.



Задача 2 (10 баллов)

Чертёжник

Исполнитель Чертёжник перемещается на координатной плоскости, оставляя след в виде линии. Чертёжник может выполнять следующие команды:

1) Сместиться на (a, b) , где a, b – целые числа. Эта команда перемещает исполнителя из точки с координатами (x, y) в точку с координатами $(x + a; y + b)$. Например, если Чертёжник находится в точке с координатами $(1, 2)$, то команда сместиться на $(2, -1)$ переместит Чертёжника в точку $(3, 1)$.

2) Цикл:

ПОКА <условие>

последовательность команд

КОНЕЦ ПОКА

Этот цикл означает, что последовательность команд будет выполняться до тех пор, пока условие <условие> выполняется.

Исполнителю Чертёжник был дан для исполнения следующий алгоритм (буквами a, b обозначены неизвестные числа):

НАЧАЛО

 сместиться на $(2, -3)$

ПОКА $x + y$ не равно 5

 сместиться на (a, b)

 сместиться на $(-1.5, 4.5)$

КОНЕЦ ПОКА

 сместиться на $(-8, -3)$

КОНЕЦ

После выполнения программы смещение Чертёжника по оси y будет в три раза меньше смещения по оси x . Сколько раз выполнялась последовательность команд цикла ПОКА, если известно, что команды цикла ПОКА были выполнены более одного раза и сумма значений координат по осям x и y до выполнения программы была равна 6

Решение

Значение координат Чертёжника в результате выполнения цикла ПОКА:

$$x_n = x_0 + \Delta x = x_0 + 2 + N(a - 1.5)$$

$$y_n = y_0 + \Delta y = y_0 - 3 + N(b + 5.5)$$

В этом уравнении N – количество повторений команд цикла ПОКА, которое является искомым значением.

Из условия завершения цикла $x_n + y_n = 5$, получим:

$$x_0 + 2 + N(a - 1.5) + y_0 - 3 + N(b + 5.5) = 5$$

Из условия $x_0 + y_0 = 6$, получим:



$$N(a + b + 4) = 6$$

Общее изменение координат Чертёжника в результате выполнения этого алгоритма:

$$x - x_0 = 2 + N(a - 1,5) - 8$$

$$y - y_0 = -3 + N(b + 5,5) - 3$$

Так как изменение координат по оси x будет превышать изменение координат по оси y в три раза, получим:

$$(2 + N(a - 1,5) - 8) = 3(-3 + N(b + 5,5) - 3)$$

$$N(a - 1,5) - 6 = N(3b + 16,5) - 18$$

$$N(a - 3b - 18) = 12$$

Получим систему уравнений:

$$\begin{cases} N(a + b + 4) = 6 \\ N(a - 3b - 18) = 12 \end{cases}$$

Нужно найти натуральное N , при котором эта система уравнений разрешима в целых числах относительно a и b . Для этого число N должно быть одновременно делителем чисел 6 и 12. Общими делителями этих чисел являются числа 1, 2, 3 и 6, при этом 1 не подходит по условию, что цикл повторялся более 1 раза, а числа 2 и 6 не подходят из условия, что a , b – целые числа, при подстановке в систему уравнений выше.

Ответ: последовательность команд цикла ПОКА выполнилась 3 раза.

**Задача 3 (10 баллов)****Шкатулка**

Николай купил новую цифровую шкатулку с четырехзначным кодом. Он придумал код. Чтобы лучше его запомнить, он решил преобразовать каждую цифру этого 4-значного числа в двоичный код. Далее он составил логическое выражение. В результирующем столбце таблицы истинности к этому выражению отобразилась та самая двоичная последовательность для всего кода.

$$(a \vee \neg b \wedge c) \rightarrow a \wedge (b \vee \neg c)$$

Строки в таблице истинности идут последовательно, 000, 001, 010 и т.д.

Какой шифр на шкатулке у Николая, если известно, что первая цифра кода – 2?

Решение

Составим таблицу истинности

a	b	c	$\neg b$	$\neg b \wedge c$	$a \vee \neg b \wedge c$	$\neg c$	$b \vee \neg c$	$a \wedge (b \vee \neg c)$	$a \vee \neg b \wedge c \rightarrow a \wedge (b \vee \neg c)$
0	0	0	1	0	0	1	1	0	1
0	0	1	1	1	1	0	0	0	0
0	1	0	0	0	0	1	1	0	1
0	1	1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	1	1	1
1	0	1	1	1	1	0	0	0	0
1	1	0	0	0	1	1	1	1	1
1	1	1	0	0	1	0	1	1	1

Двоичный код в последнем столбце можно разбить на двоичные числа следующим образом: $10_2 11_2 101_2 1_2$

Таким образом получаем следующий десятичный код: 2 3 5 1

Ответ: 2 3 5 1



Задача 4 (20 баллов)

Добыча урана

Шахта «Уран-2» использует систему шифрования для передачи сообщения о добыче урана в головную организацию. Для дешифровки данного сообщения каждый элемент закрытого ключа, хранящегося в головной организации, умножается на n^{-1} – мультипликативное обратное к значению n по модулю m (остаток от деления $n \cdot n^{-1}$ на m равен 1). Результат вычисляется как остаток от деления полученного числа на m . Полученный модифицированный закрытый ключ используется для дешифрования, а 8-битный ключ определяет, какие элементы модифицированного ключа суммируются для получения расшифрованного сообщения.

В понедельник добыча урана на шахте составила $3B_{16}$ килограмм. Ежедневно добываемая масса урана увеличивалась согласно арифметической прогрессии, состоящей из последовательности четных чисел, начиная со второго члена (например, количество добытого урана во второй день равно $3B_{16} + 4_{16}$ кг и т.д.). С шахты в головную организацию в течение рабочей недели поступило пять сообщений с количеством добытого урана, которые преобразовывались в головной организации в 8-битный двоичный код и служили ключом для дешифрования. Закрытый ключ, хранящийся в организации, представляет собой следующую последовательность $\{2, 3, 9, 17, 28, 47, 81, 131\}$, значения n^{-1} и m равны 91 и 320 соответственно.

Расшифруйте полученные сообщения, зная, что они представляют номер склада, где хранится добытый уран в каждый из дней рабочей недели. В ответе укажите последовательность пяти чисел через запятую, начиная с номера склада, где хранился добытый уран в понедельник, заканчивая номером склада, где хранился добытый уран в пятницу.

Решение

На первом этапе необходимо определить сообщения, которые были переданы в административное управление со вторника по пятницу. В условии сказано, что ежедневно добываемая масса урана (шифротекст) увеличивалась согласно арифметической прогрессии, состоящей из последовательности четных чисел, начиная со второго члена, т.е. во второй день количество добытого урана увеличилось на 4_{16} кг, в третий день – на 6_{16} кг, в четвертый день – на 8_{16} кг и в пятый день – на 10_{16} кг. Для нахождения переданных сообщений произведем сложение чисел в шестнадцатеричной системе.

Сложение чисел в шестнадцатеричной системе счисления выполняется поразрядно, начиная с младшего разряда. Если при сложении двух чисел одинакового разряда получается двузначное число, то значение его старшего разряда (единицу) добавляют в старший разряд. Следовательно:



Второе сообщение: $3V_{16} + 4_{16} \Rightarrow V + 4 = F \Rightarrow 3F_{16}$

Третье сообщение: $3F_{16} + 6_{16} \Rightarrow F + 6 = 15 \Rightarrow 3 + 1 = 4 \Rightarrow 45_{16}$

Четвертое сообщение: $45_{16} + 8_{16} \Rightarrow 5 + 8 = D \Rightarrow 4D_{16}$

Пятое сообщение: $4D_{16} + 10_{16} \Rightarrow D + 0 = D \Rightarrow 4 + 1 = 5 \Rightarrow 5D_{16}$

Таким образом, последовательность переданных сообщений в течение рабочей недели: $3V_{16}, 3F_{16}, 45_{16}, 4D_{16}, 5D_{16}$

Далее необходимо найти ключ для дешифрования каждого сообщения. По условию задачи он представляет собой преобразованное значение переданного сообщения в 8-битном двоичном виде. Для перевода числа из шестнадцатеричной системы счисления в двоичную можно воспользоваться несколькими способами, мы воспользуемся переводом сначала в десятичную систему счисления, затем в двоичную.

Каждую цифру шестнадцатеричного числа умножим на 16 в степени, равной позиции этой цифры, и затем сложим все полученные произведения. Далее разделим полученное число в десятичной системе на 2 и запишем остаток до тех пор, пока неполное частное не будет равно нулю. Полученные остатки записываем в обратном порядке.

Число в 16-ой системе счисления	Перевод в 10-ую систему счисления	Число в 10-ой системе счисления	Перевод в 2-ую систему счисления	Число в 2-ой системе счисления
$3V_{16}$	$3 \times 16^1 + V \times 16^0$ $= 3 \times 16^1 + 11$ $\times 16^0 = 48 + 11$ $= 59$	59_{10}	1) $\frac{59}{2} = 29$ (остаток 1) 2) $\frac{29}{2} = 14$ (остаток 1) 3) $\frac{14}{2} = 7$ (остаток 0) 4) $\frac{7}{2} = 3$ (остаток 1) 5) $\frac{3}{2} = 1$ (остаток 1) 6) $\frac{1}{2} = 0$ (остаток 1)	111011_2
$3F_{16}$	$3 \times 16^1 + F \times 16^0$ $= 3 \times 16^1 + 15$ $\times 16^0 = 48 + 15$ $= 63$	63_{10}	1) $\frac{63}{2} = 31$ (остаток 1) 2) $\frac{31}{2} = 15$ (остаток 1) 3) $\frac{15}{2} = 7$ (остаток 1) 4) $\frac{7}{2} = 3$ (остаток 1) 5) $\frac{3}{2} = 1$ (остаток 1) 6) $\frac{1}{2} = 0$ (остаток 1)	111111_2



45_{16}	$4 \times 16^1 + 5 \times 16^0$ $= 64 + 5 = 69$	69_{10}	1) $\frac{69}{2} = 34$ (остаток 1) 2) $\frac{34}{2} = 17$ (остаток 0) 3) $\frac{17}{2} = 8$ (остаток 1) 4) $\frac{8}{2} = 4$ (остаток 0) 5) $\frac{4}{2} = 2$ (остаток 0) 6) $\frac{2}{2} = 1$ (остаток 0) 7) $\frac{1}{2} = 0$ (остаток 1)	1000101_2
$4D_{16}$	$4 \times 16^1 + D \times 16^0$ $= 4 \times 16^1 + 13$ $\times 16^0 = 64 + 13$ $= 77$	77_{10}	1) $\frac{77}{2} = 38$ (остаток 1) 2) $\frac{38}{2} = 19$ (остаток 0) 3) $\frac{19}{2} = 9$ (остаток 1) 4) $\frac{9}{2} = 4$ (остаток 1) 5) $\frac{4}{2} = 2$ (остаток 0) 6) $\frac{2}{2} = 1$ (остаток 0) 7) $\frac{1}{2} = 0$ (остаток 1)	1001101_2
$5D_{16}$	$5 \times 16^1 + D \times 16^0$ $= 5 \times 16^1 + 13$ $\times 16^0 = 80 + 13$ $= 93$	93_{10}	1) $\frac{93}{2} = 46$ (остаток 1) 2) $\frac{46}{2} = 23$ (остаток 0) 3) $\frac{23}{2} = 11$ (остаток 1) 4) $\frac{11}{2} = 5$ (остаток 1) 5) $\frac{5}{2} = 2$ (остаток 1) 6) $\frac{2}{2} = 1$ (остаток 0) 7) $\frac{1}{2} = 0$ (остаток 1)	1011101_2

Мы нашли 8-битный двоичный код для каждого сообщения, который служит ключом для дешифрования. Далее исходя из условия задачи для дешифровки сообщений требуется модифицированный закрытый ключ, который рассчитывается как каждый элемент закрытого ключа умноженный на n^{-1} и найденный остаток от деления полученного числа на m . В нашем случае закрытый ключ представляет последовательность {2, 3, 9, 17, 28, 47,



81, 131}, значения n^{-1} и m равны 91 и 320 соответственно.

Модифицированный закрытый ключ:

- 1) (2×91) делится на 320 с остатком 182
- 2) (3×91) делится на 320 с остатком 273
- 3) (9×91) делится на 320 с остатком 179
- 4) (17×91) делится на 320 с остатком 267
- 5) (28×91) делится на 320 с остатком 308
- 6) (47×91) делится на 320 с остатком 117
- 7) (81×91) делится на 320 с остатком 11
- 8) (131×91) делится на 320 с остатком 81

Модифицированный закрытый ключ

{182, 273, 179, 267, 308, 117, 11, 81}.

Найденный ранее 8-битный ключ определяет, какие элементы модифицированного ключа суммируются для получения расшифрованного сообщения. При нехватке элементов 8-битного ключа при выборе элементов модифицированного ключа, недостающие элементы приравниваются к 0). Таким образом, произведем дешифровку принятых в административном управлении сообщений:

Первое сообщение:

$$\begin{array}{cccccccc} 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 182 & 273 & 179 & 267 & 308 & 117 & 11 & 81 \end{array} = 179 + 267 + 308 + 11 + 81 = 846$$

Второе сообщение:

$$\begin{array}{cccccccc} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 182 & 273 & 179 & 267 & 308 & 117 & 11 & 81 \end{array} = 179 + 267 + 308 + 117 + 11 + 81 = 963$$

Третье сообщение:

$$\begin{array}{cccccccc} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 182 & 273 & 179 & 267 & 308 & 117 & 11 & 81 \end{array} = 273 + 117 + 81 = 471$$

Четвертое сообщение:

$$\begin{array}{cccccccc} 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 182 & 273 & 179 & 267 & 308 & 117 & 11 & 81 \end{array} = 273 + 308 + 117 + 81 = 779$$

Пятое сообщение:

$$\begin{array}{cccccccc} 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 182 & 273 & 179 & 267 & 308 & 117 & 11 & 81 \end{array} = 273 + 267 + 308 + 117 + 81 = 1046$$

В ответе необходимо указать последовательность пяти чисел через запятую, начиная с номера склада, где хранился добытый уран в понедельник, заканчивая номером склада, где хранился добытый уран в пятницу. Следовательно, ответ задачи выглядит следующим образом 846, 963, 471, 779, 1046.

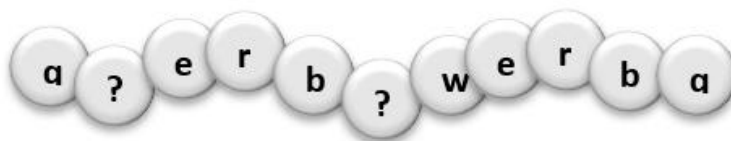
Ответ: 846, 963, 471, 779, 1046.



Задача 5 (25 баллов)

Задача об удивительных последовательностях

Профессор Тимофей Чудоген, известный биолог, проводит интересное исследование, в котором изучает особый искусственный генетический код. Этот код состоит из последовательности искусственных нуклеотидов, которые обозначаются строчными латинскими буквами. Однако в этой последовательности есть вопросы — знаки, которые нужно заменить на определенные буквы, чтобы завершить анализ.



Профессор считает, что этот код имеет удивительную особенность, которую он называет А-периодичность. Периодичность означает, что если взять два искусственных нуклеотида, расположенных на расстоянии, кратном заданному числу А, то они должны быть одинаковыми. Это значит, что все нуклеотиды, стоящие на тех же позициях, кратных А, должны совпадать.

Профессор Чудоген уверен, что понимание этих особенностей поможет в будущем в таких науках, как медицина, экология и биология. Его задача заключается в том, чтобы найти правильные буквы для замены знаков вопроса и определить можно ли соответствующую строку сделать А-периодичной путем замены всех знаков вопроса. Помогите профессору.

Формат входных данных

В первой строке содержится два числа через пробел: N — общее количество строк, которые вам нужно проверить на А-периодичность и число A ($1 \leq A \leq 5 \cdot 10^3$). Далее содержится N непустых строк, состоящих из малых букв латиницы и знаков вопроса $1 \leq N \leq 5 \cdot 10^4$, суммарная длина всех строк не превосходит $2 \cdot 10^5$.

Формат выходных данных

Вывести N строк, в каждой из которых вывести либо "YES", если соответствующую строку можно сделать А-периодичной путем замены всех знаков вопроса на некоторые буквы и "NO" в противном случае. Будем считать последовательность А-периодичной, если для любых двух символов, расстояние между которыми кратно А верно, что они совпадают.



Примеры

стандартный ввод	стандартный вывод
8 1	NO
abacabaca	NO
abracadabra	YES
aa?aaa?aaaaaaaaaaa	YES
q	YES
???????	NO
q?er?w?	NO
q?erw??	NO
q?er?wer?werw?er	

Решите указанную задачу, используя один из языков программирования, и укажите, какой именно язык был выбран. Предложите эффективное решение, учитывая минимизацию времени выполнения программы и объема памяти, необходимого для её работы. Постарайтесь использовать алгоритмы и структуры данных, которые обеспечивают оптимальную производительность.

Реализуйте решение в виде программного кода, обязательно добавив комментарии, которые поясняют ключевые шаги алгоритма и переменные, если это важно для понимания проверяющему. Допускается использование только стандартных библиотек, встроенных в язык. Подробно опишите алгоритм решения, предоставив текстовое описание или блок-схему, объяснив логику работы программы.

Приведите результат выполнения программы для заданных исходных данных. Результат выполнения для указанных исходных данных должен включать вывод программы, а также, если соответствующую строку можно сделать A-периодичной заменой всех знаков вопроса, привести полученную строку.

Исходные данные для выполнения задания

стандартный ввод	стандартный вывод
2 4	?
q?er?wer?wer?w	
ad?fg?def?ad?f	

Ваше решение должно быть аккуратно оформлено, читабельно и соответствовать стандартам выбранного языка программирования.

Решение

Чтение входных данных

Программа сначала считывает количество строк и значение, которое определяет период, по которому будет происходить проверка.

Инициализация

Для каждой строки создается флаг ok, который по умолчанию устанавливается в true. Этот флаг отслеживает, соответствует ли текущая



строка необходимым критериям А-периодичности.

Проверка А-периодичности

Для каждого индекса от 0 до $p-1$ (внешний цикл) инициализируется символ s , устанавливаемый в '#'. Он будет использоваться для отслеживания первого ненулевого символа в строке.

Внутренний цикл проходит по строке с шагом p , начиная с текущего индекса. Если текущий символ не равен '?', проверяется:

Если s все еще '#', это значит, что найден первый ненулевой символ, и s устанавливается на этот символ.

Если s уже установлен, проверяется, совпадает ли текущий символ с s . Если нет — флаг ok устанавливается в `false`, и цикл прерывается.

Сохранение результата

После завершения проверки для каждой строки результат ("YES" или "NO") добавляется в список `results`.

Вывод результата

В конце программа выводит результаты всех тестов.

Реализация на C++

```
#include <bits/stdc++.h>
#define int long

using namespace std;

signed main() {
    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);

    int u, p;
    cin >> u >> p;
    vector<string> results; // Вектор для хранения результатов

    while (u--) {
        string s;
        cin >> s;
        bool ok = true; // Флаг, отслеживает правильность строк

        for (int i = 0; i < p; i++) {
            char c = '#'; // Устанавливаем символ по умолчанию
            for (int j = 0; i + j < (int)(s).size(); j += p) {
                if (s[i + j] != '?') {
                    if (c == '#') {
                        c = s[i + j]; // Устанавливаем 1-ый ненулевой
                    } else {
                        if (c != s[i + j]) {
                            ok = false; // Устанавливаем флаг в false
                        }
                    }
                }
            }
        }

        if (ok) {
            results.push_back("YES"); // Сохраняем результат
        } else {
            results.push_back("NO"); // Сохраняем результат
        }
    }
}
```



```
    }  
}  
  
// Выводим все результаты после завершения всех проверок  
for (int i = 0; i < (int) (results).size(); i++) {  
    cout << results[i] << "\n";  
}  
}
```

Реализация на Python

```
# Считываем с клавиатуры количество строк и значение p  
u, p = map(int, input().split())  
  
# Создаём список для хранения результатов  
results = []  
  
# Обрабатываем u строк  
for _ in range(u):  
    s = input().strip() # Считываем текущую строку  
    ok = True # Флаг, указывающий на A-периодичность  
  
    # Проверяем A-периодичность строки  
    for i in range(p):  
        c = '#' # Инициализируем переменную для отслеживания символа  
        for j in range(i, len(s), p):  
            if s[j] != '?':  
                if c == '#':  
                    c = s[j] # Устанавливаем 1-ый ненулевой  
                elif c != s[j]:  
                    ok = False # Устанавливаем флаг в False  
                    break  
  
        if not ok:  
            break  
  
    # Сохраняем результат в список  
    if ok:  
        results.append("YES")  
    else:  
        results.append("NO")  
  
# Выводим все результаты после завершения обработки строк  
for result in results:  
    print(result)
```

Временную сложность алгоритма можно определить следующим образом:

Пусть длина строки n . Внутренний цикл выполняется для каждого i от 0 до $p-1$. В каждой итерации внутреннего цикла мы проходим элементы строки с шагом p . Таким образом, количество итераций внутреннего цикла составляет примерно n/p для каждого i . В итоге, временная сложность будет: $O(u \times (p + n/p))$, где u — количество строк для проверки. Так как n значительно больше p , то можно упростить до: $O(u \times n)$.

Основные потребности в памяти включают вектор для хранения результатов и саму строку.



Результат выполнения алгоритма для указанных исходных данных

стандартный ввод	стандартный вывод
2 4 q?er?wer?wer?w ad?fg?def?ad?f	YES NO

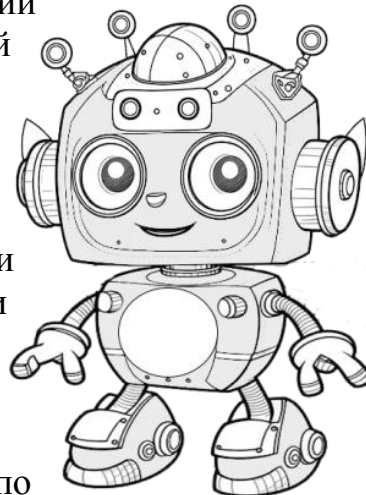
Для первого слова можем получить A-периодичную последовательность **qwerqwerqwerqw**.



Задача 6 (25 баллов)

РобоМакс и тайна цветных деталей

На производственном предприятии «Механический Мастер» функционирует длинный конвейер, на котором детали пронумерованы от 1 до N (где $N \leq 10^5$). Эти детали движутся по кругу, таким образом, перед первой деталью оказывается деталь с номером N , а перед любой другой — деталь с номером, на единицу меньшим. Детали с четными номерами предназначены для зеленых машин и окрашены в яркий зеленый цвет, в то время как нечетные детали окрашены в насыщенный синий цвет и предназначены для синих машин.



На этом заводе трудится наш герой — робот по имени РобоМакс. РобоМакс — маленький, но невероятно умный робот с большими глазами и яркой светодиодной панелью. Его задача состоит в том, чтобы забирать детали для дальнейшего производства. Однако у него есть особое правило: он может забрать только те детали, у которых обе соседние детали (та, что перед ней, и та, что за ней) имеют одинаковый цвет.

РобоМакс внимательно осматривает конвейер. Если обе соседние детали имеют одинаковый цвет, он забирает центральную деталь, в то время как соседние продолжают двигаться по кругу, не меняя своих положений. Если же одна деталь зеленая, а другая синяя, РобоМакс не может забрать нужную деталь, и она остается на месте, вызывая у него легкую грусть.

Представьте себе, как быстро и эффективно работает РобоМакс, вызывая восхищение рабочих. Однако постепенно становится заметно, что некоторые детали остаются без внимания. Теперь возникает вопрос: сколько деталей останется на конвейере после нескольких успешных операций РобоМакса? Ваша задача заключается в том, чтобы помочь ему выяснить, сколько деталей в итоге останется на конвейере, учитывая правила его работы.

Формат входных данных

В первой строке два натуральных числа N и K ($3 \leq N \leq 50000$, $1 \leq K \leq N-2$). Во второй строке через пробел K чисел, каждое из интервала от 1 до N — номера деталей в том порядке, в котором их пытается взять РобоМакс. Если детали с указанным номером к моменту, когда за ней приходит РобоМакс, уже нет, ничего не происходит.

Формат выходных данных



Одно число — ответ на вопрос.

Примеры

стандартный ввод	стандартный вывод
13 11 1 2 1 13 2 6 12 3 13 5 3	8

Решите указанную задачу, используя один из языков программирования, и укажите, какой именно язык был выбран. Предложите эффективное решение, учитывая минимизацию времени выполнения программы и объема памяти, необходимого для её работы. Постарайтесь использовать алгоритмы и структуры данных, которые обеспечивают оптимальную производительность.

Реализуйте решение в виде программного кода, обязательно добавив комментарии, которые поясняют ключевые шаги алгоритма и переменные, если это важно для понимания проверяющему. Допускается использование только стандартных библиотек, встроенных в язык. Подробно опишите алгоритм решения, предоставив текстовое описание или блок-схему, объяснив логику работы программы.

Приведите результат выполнения программы для заданных исходных данных.

Исходные данные для выполнения задания

стандартный ввод	стандартный вывод
15 10 1 2 10 11 8 3 5 12 4 7	?

Ваше решение должно быть аккуратно оформлено, читабельно и соответствовать стандартам выбранного языка программирования.

Решение

Чтение входных данных

Сначала считываются значения n (количество деталей) и k (количество операций).

Инициализация массивов

Создаются два массива `pred` и `next`, которые представляют индексы предыдущей и следующей деталей соответственно. Массив `lab` инициализируется единицами, показывая, что изначально все детали находятся на конвейере.

Обработка операций

Считывается номер детали и проверяется, можно ли забрать деталь по



заданным правилам. Если обе соседние детали одного цвета и текущая деталь не была забрана, она забирается, и связи между соседними деталями обновляются.

Подсчет оставшихся деталей

В конце программы производится подсчет оставшихся на конвейере деталей, после чего результат выводится на экран.

Реализация на C++

```
#include<bits/stdc++.h>
#define int long
using namespace std;
int n, k, a; // Объявляем глобальные переменные: n (количество
деталей), k (количество запросов), a (номер детали запроса)

signed main(){
    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
    // Чтение входных данных: количество деталей и количество
запросов
    cin >> n >> k;
    // Массивы для предшествующих, последующих и меток деталей
    vector<int> pred(n), next(n), lab(n, 1);

    // Заполняем массив предшествующих, указывая,
    // что предыдущая деталь у первой – последняя
    for(int i = 0; i < n; i++) pred[i] = (i-1 + n) % n;
    // Заполняем массив последующих, указывая,
    // что следующая деталь у последней – первая
    for(int i = 0; i < n; i++) next[i] = (i+1) % n;
    // Обработка запросов РобоМакса
    for(int i = 0; i < k; i++){
        cin >> a; // Читаем номер детали
        a--;
        // Переход к нулевой индексации (т.к. вводятся номера от 1 до N)
        if(pred[a]%2 == next[a]%2 && lab[a] == 1){
            lab[a] = 0; // Помечаем деталь как убранную
            next[pred[a]] = next[a];
            // Обновляем следующую деталь для предшествующей
            pred[next[a]] = pred[a];
            // Обновляем предшествующую для следующей детали
        }
    }
    // Подсчет оставшихся деталей на конвейере
    int ans = 0;
    for(auto q : lab) ans += q;
    cout << ans; // Выводим результат
}
```

Реализация на Python

```
def main():
    n, k = map(int, input().split())
```



```
# Создаем массивы предшествующих и последующих деталей
pred = [(i - 1 + n) % n for i in range(n)]
next = [(i + 1) % n for i in range(n)]
# Массив, забрана ли деталь (1 - нет, 0 - да)
lab = [1] * n
for _ in range(k):
    # Индекс детали, которую РобоМакс собирается проверить
    a = int(input()) - 1
    # Проверяем, могут ли соседние детали быть одного цвета
    if pred[a] % 2 == next[a] % 2 and lab[a] == 1:
        lab[a] = 0 # Забираем деталь
        next[pred[a]] = next[a] # меняем соседние
        pred[next[a]] = pred[a]
# Подсчитываем количество оставшихся деталей
ans = sum(lab)
print(ans)

if __name__ == "__main__":
    main()
```

Давайте проанализируем временную и пространственную сложность решения. Создание массивов `pred`, `next` и `lab` требует $O(n)$ времени, поскольку мы проходимся по всем деталям. Цикл по запросам выполняется k раз. время выполнения всей программы будет суммироваться: $O(n)+O(k)=O(n+k)$.

Мы используем три массива: `pred`, `next` и `lab`, каждый из которых занимает $O(n)$ пространства. Следовательно, потребление памяти для этих массивов составляет $O(n)$.

Результат выполнения алгоритма для указанных исходных данных

стандартный ввод	стандартный вывод
15 10 1 2 10 11 8 3 5 12 4 7	10



ВАРИАНТ 2

Задача 1 (10 баллов)

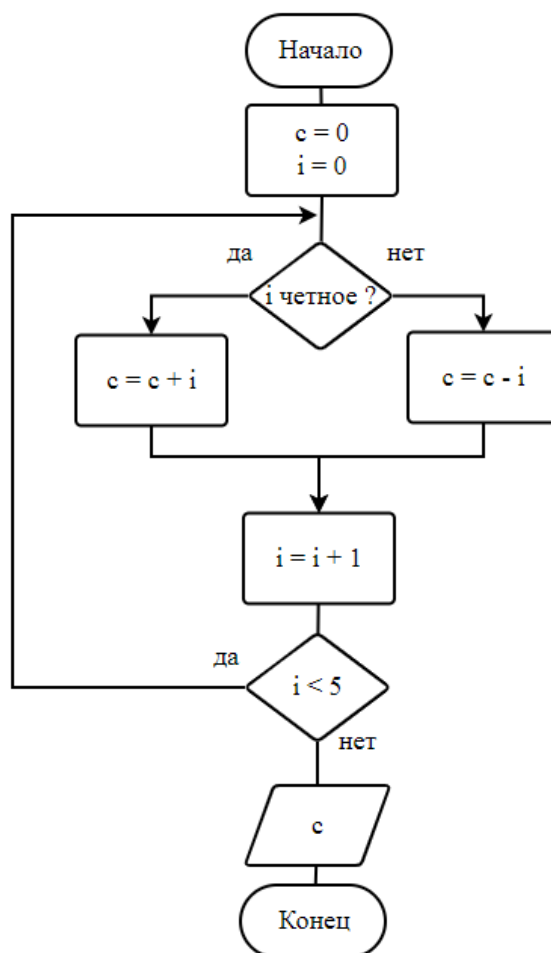
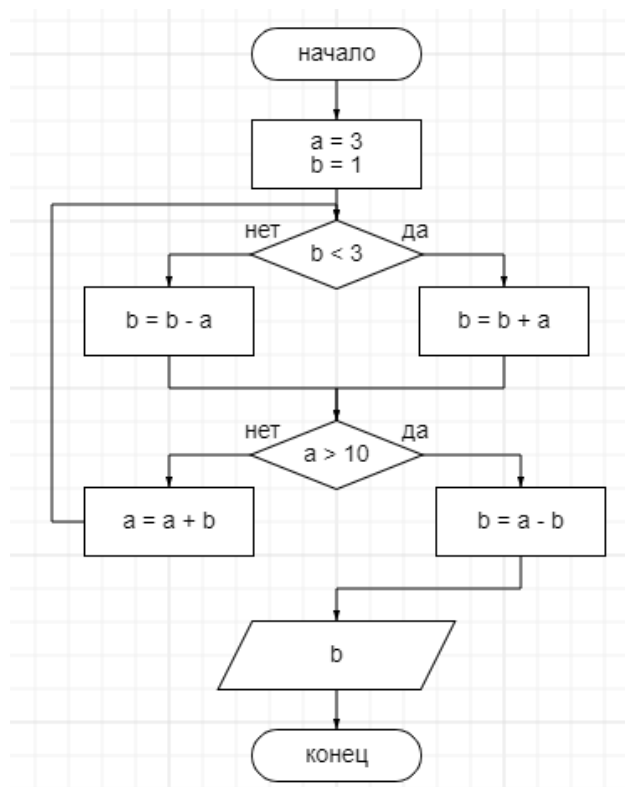
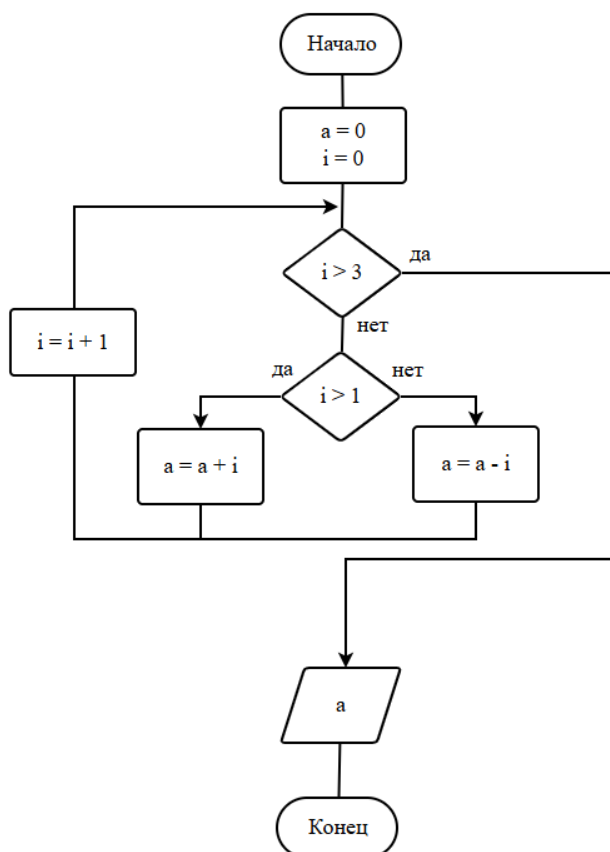
Разведчики

Нашими разведчиками на линии боевого соприкосновения были обнаружены средства противника, сведения о которых они передали в виде закодированной электронной таблицы на командный пункт. Определите количество этих технических средств, значение которых расположены в ячейках B2:F2. В ответе запишите значения ячеек в одну строчку, через запятую, начиная с B2 и заканчивая F2.

	A	B	C	D	E	F
		Танки	БМП	Орудия	Установки РЭБ	Опорные пункты
1						
2						
3						
4	a =				1) =СЧЁТЕСЛИ(B2:F2;B4)	
5	b =				2) =СЧЁТЕСЛИ(E2;B4)+B5	
6	c =				3) =СУММЕСЛИ(D2:E2;B4)-B6	
7	d =				4) =СУММ(D2:F2)-B6	
8					5) =ЕСЛИ(B7>5;B2-C2;СУММ(B2;C2))	
9					6) =ЕСЛИ(B7<=5;B2-C2;СУММ(B2;C2))	
10						

	A	B	C	D	E	F
		Танки	БМП	Орудия	Установки РЭБ	Опорные пункты
1						
2						
3						
4	a =				1)	2
5	b =				2)	17
6	c =				3)	6
7	d =				4)	11
8					5)	8
9					6)	2
10						

Значения a, b и c определите, выполнив соответствующие алгоритмы по блок-схемам, d (ячейка B7) – неизвестное натуральное число.





Решение

Определяем значения коэффициентов a , b , c выполнив соответствующие алгоритмы по представленным блок-схемам. В результате получаем следующие значения.

	A	B	C
3			
4	a =	4	
5	b =	16	
6	c =	2	
7	d =		
8			

1) Значение формулы в ячейке E4 «=СЧЁТЕСЛИ(B2:F2;B4)» равно 2, следовательно в ячейках B2:F2 находятся две 4.

2) Значение формулы в ячейке E5 «=СЧЁТЕСЛИ(E2;B4)+B5» равно 17, откуда значение «=СЧЁТЕСЛИ(E2;B4)» равно $17 - B5 = 17 - 16 = 1$. Следовательно в ячейке E2 находится 4.

3) Значение формулы в ячейке E6 «=СУММЕСЛИ(D2:E2;B4)-B6» равно 6, откуда значение «=СУММЕСЛИ(D2:E2;B4)» равно $6 + B6 = 6 + 2 = 8$. Следовательно в ячейке D2, также как и в E2, находится значение 4.

4) Значение формулы в ячейке E7 «=СУММ(D2:F2)-B6» равно 11, откуда значение «=СУММ(D2:F2)» равно $11 + B6 = 11 + 2 = 13$. Значения ячеек D2 и E2 известны, находим значение ячейки F2. $F2 = 13 - 4 - 4 = 5$.

5) Анализируя формулы в ячейке E8 «=ЕСЛИ(B7>5;B2-C2;СУММ(B2;C2))» и ячейке E9 «=ЕСЛИ(B7<=5;B2-C2;СУММ(B2;C2))» понимаем, что если значение в ячейке B7 будет больше 5, то в ячейке E8 будет $B2 - C2$, а в ячейке E9: $B2 + C2$, в противном случае в ячейке E8 будет $B2 + C2$, а в ячейке E9: $B2 - C2$. Так как, по условию, значения в ячейках B2:F2 натуральные числа (количество объектов), то разность всегда меньше суммы, получаем:

$$\begin{cases} B2 + C2 = 8 \\ B2 - C2 = 2 \end{cases} \Rightarrow \begin{cases} C2 + 2 + C2 = 8 \\ B2 = C2 + 2 \end{cases} \Rightarrow \begin{cases} C2 = 3 \\ B2 = 5 \end{cases}$$

Ответ: 5, 3, 4, 4, 5.



Задача 2 (10 баллов)

Чертёжник

Исполнитель Чертёжник перемещается на координатной плоскости, оставляя след в виде линии. Чертёжник может выполнять следующие команды:

1) Сместиться на (a, b) , где a, b – целые числа. Эта команда перемещает исполнитель из точки с координатами (x, y) в точку с координатами $(x + a; y + b)$. Например, если Чертёжник находится в точке с координатами $(1, 2)$, то команда сместиться на $(2, -1)$ переместит Чертёжника в точку $(3, 1)$.

2) Цикл:

ПОКА <условие>

последовательность команд

КОНЕЦ ПОКА

Этот цикл означает, что последовательность команд будет выполняться до тех пор, пока условие <условие> выполняется.

Исполнителю Чертёжник был дан для исполнения следующий алгоритм (буквами a, b обозначены неизвестные числа):

НАЧАЛО

сместиться на $(3, -1)$

ПОКА $x + y$ не равно 12

сместиться на (a, b)

сместиться на $(-1.5, 7.5)$

КОНЕЦ ПОКА

сместиться на $(-14, -8)$

КОНЕЦ

После выполнения программы смещение Чертёжника по оси y будет в два раза больше смещения по оси x . Сколько раз выполнялась последовательность команд цикла ПОКА, если известно, что команды цикла ПОКА были выполнены более одного раза и сумма значений координат по осям x и y до выполнения программы была равна 4

Решение

Значение координат Чертёжника в результате выполнения цикла ПОКА:

$$x_n = x_0 + \Delta x = x_0 + 3 + N(a - 1.5)$$

$$y_n = y_0 + \Delta y = y_0 - 1 + N(b + 7.5)$$

В этом уравнении N – количество повторений команд цикла ПОКА, которое является искомым значением.

Из условия завершения цикла $x_n + y_n = 12$, получим:



$$x_0 + 3 + N(a + 1,5) + y_0 - 1 + N(b - 7,5) = 12$$

Из условия $x_0 + y_0 = 4$, получим:

$$N(a + b - 6) = 6$$

Общее изменение координат Чертёжника в результате выполнения этого алгоритма:

$$x - x_0 = 3 + N(a - 1) - 14$$

$$y - y_0 = -1 + N(b + 2) - 8$$

Так как изменение координат по оси y будет превышать изменение координат по оси x в три раза, получим:

$$3(3 + N(a - 1,5) - 14) = -1 + N(b - 7,5) - 8$$

$$N(3a + 4,5) - 33 = N(b - 7,5) - 9$$

$$N(3a - b + 12) = 24$$

Получим систему уравнений:

$$\begin{cases} N(a + b - 6) = 6 \\ N(3a - b + 12) = 24 \end{cases}$$

Нужно найти натуральное N , при котором эта система уравнений разрешима в целых числах относительно a и b . Для этого число N должно быть одновременно делителем чисел 6 и 24. Общими делителями этих чисел являются числа 1, 2, 3 и 6, при этом 1 не подходит по условию, что цикл повторялся более 1 раза, а числа 2 и 6 не подходят из условия, что a , b – целые числа, при подстановке в систему уравнений выше.

Ответ: последовательность команд цикла ПОКА выполнилась 3 раз.

**Задача 3 (10 баллов)****Шкатулка**

Николай купил новую цифровую шкатулку с четырехзначным кодом. Он придумал код. Чтобы лучше его запомнить, он решил преобразовать каждую цифру этого 4-значного числа в двоичный код. Далее он составил логическое выражение. В результирующем столбце таблицы истинности к этому выражению отобразилась та самая двоичная последовательность для всего кода.

$$\neg(b \wedge c \wedge \neg a) \rightarrow (b \vee \neg a)$$

Строки в таблице истинности идут последовательно, 000, 001, 010 и т.д.

Какой шифр на шкатулке у Николая, если известно, что первая и последняя цифры в коде одинаковые?

Решение

Составим таблицу истинности

a	b	c	$b \wedge c$	$\neg a$	$b \wedge c \wedge \neg a$	$\neg b \wedge c \wedge \neg a$	$b \vee \neg a$	$\neg b \wedge c \wedge \neg a \rightarrow (b \vee \neg a)$
0	0	0	0	1	0	1	1	1
0	0	1	0	1	0	1	1	1
0	1	0	0	1	0	1	1	1
0	1	1	1	1	1	0	1	1
1	0	0	0	0	0	1	0	0
1	0	1	0	0	0	1	0	0
1	1	0	0	0	0	1	1	1
1	1	1	1	0	0	1	1	1

Двоичный код в последнем столбце можно разбить на двоичные числа следующим образом: $11_2 1_2 100_2 11_2$

Таким образом получаем следующий десятичный код: 3 1 4 3

Ответ: 3 1 4 3

**Задача 4 (20 баллов)****Добыча урана**

Шахта «Уран-2» использует систему шифрования для передачи сообщения о добыче урана в головную организацию. Для дешифровки данного сообщения каждый элемент закрытого ключа, хранящегося в головной организации, умножается на n^{-1} – мультипликативное обратное к значению n по модулю m (остаток от деления $n \cdot n^{-1}$ на m равен 1). Результат вычисляется как остаток от деления полученного числа на m . Полученный модифицированный закрытый ключ используется для дешифрования, а 8-битный ключ определяет, какие элементы модифицированного ключа суммируются для получения расшифрованного сообщения.

В понедельник добыча урана на шахте составила $2D_{16}$ килограмм. Ежедневно добываемая масса урана увеличивалась согласно арифметической прогрессии, состоящей из последовательности полных квадратов, начиная со второго члена (например, количество добытого урана во второй день равно $2D_{16} + 1_{16}$ кг и т.д.). С шахты в головную организацию в течение рабочей недели поступило пять сообщений с количеством добытого урана, которые преобразовывались в головной организации в 8-битный двоичный код и служили ключом для дешифрования. Закрытый ключ, хранящийся в организации, представляет собой следующую последовательность $\{1, 4, 6, 13, 21, 45, 69, 119\}$, значения n^{-1} и m равны 87 и 280 соответственно.

Расшифруйте полученные сообщения, зная, что они представляют номер склада, где хранится добытый уран в каждый из дней рабочей недели. В ответе укажите последовательность пяти чисел через запятую, начиная с номера склада, где хранился добытый уран в понедельник, заканчивая номером склада, где хранился добытый уран в пятницу.

Решение

На первом этапе необходимо определить сообщения, которые были переданы в административное управление со вторника по пятницу. В условии сказано, что ежедневно добываемая масса урана (шифротекст) увеличивалась согласно арифметической прогрессии, состоящей из последовательности полных квадратов, начиная со второго члена, т.е. во второй день количество добытого урана увеличилось на 1_{16} кг, в третий день – на 4_{16} кг, в четвертый день – на 9_{16} кг и в пятый день – на 16_{16} кг. Для нахождения переданных сообщений произведем сложение чисел в шестнадцатеричной системе.

Сложение чисел в шестнадцатеричной системе счисления выполняется поразрядно, начиная с младшего разряда. Если при сложении двух чисел одинакового разряда получается двузначное число, то значение его старшего разряда (единицу) добавляют в старший разряд. Следовательно:



Второе сообщение: $2D_{16} + 1_{16} \Rightarrow D + 1 = E \Rightarrow 2E_{16}$

Третье сообщение: $2E_{16} + 4_{16} \Rightarrow E + 6 = 12 \Rightarrow 2 + 1 = 3 \Rightarrow 32_{16}$

Четвертое сообщение: $32_{16} + 9_{16} \Rightarrow 2 + 9 = B \Rightarrow 3B_{16}$

Пятое сообщение: $3B_{16} + 16_{16} \Rightarrow B + 6 = 11 \Rightarrow 3 + 1 + 1 = 5 \Rightarrow 5116$

Таким образом, последовательность переданных сообщений в течение рабочей недели: $2D_{16}, 2E_{16}, 32_{16}, 3B_{16}, 51_{16}$

Далее необходимо найти ключ для дешифрования каждого сообщения. По условию задачи он представляет собой преобразованное значение переданного сообщения в 8-битном двоичном виде. Для перевода числа из шестнадцатеричной системы счисления в двоичную можно воспользоваться несколькими способами, мы воспользуемся переводом сначала в десятичную систему счисления, затем в двоичную.

Каждую цифру шестнадцатеричного числа умножим на 16 в степени, равной позиции этой цифры, и затем сложим все полученные произведения. Далее разделим полученное число в десятичной системе на 2 и запишем остаток до тех пор, пока неполное частное не будет равно нулю. Полученные остатки записываем в обратном порядке.

Число в 16-ой системе счисления	Перевод в 10-ую систему счисления	Число в 10-ой системе счисления	Перевод в 2-ую систему счисления	Число в 2-ой системе счисления
$2D_{16}$	$2 \times 16^1 + D \times 16^0$ $= 2 \times 16^1 + 13$ $\times 16^0 = 32 + 13$ $= 45$	45_{10}	1) $\frac{45}{2} = 22$ (остаток 1) 2) $\frac{22}{2} = 11$ (остаток 0) 3) $\frac{11}{2} = 5$ (остаток 1) 4) $\frac{5}{2} = 2$ (остаток 1) 5) $\frac{2}{2} = 1$ (остаток 0) 6) $\frac{1}{2} = 0$ (остаток 1)	101101_2
$2E_{16}$	$2 \times 16^1 + E \times 16^0$ $= 2 \times 16^1 + 14$ $\times 16^0 = 32 + 14$ $= 46$	46_{10}	1) $\frac{46}{2} = 23$ (остаток 0) 2) $\frac{23}{2} = 11$ (остаток 1) 3) $\frac{11}{2} = 5$ (остаток 1) 4) $\frac{5}{2} = 2$ (остаток 1) 5) $\frac{2}{2} = 1$ (остаток 0) 6) $\frac{1}{2} = 0$ (остаток 1)	101110_2



32_{16}	$3 \times 16^1 + 2 \times 16^0$ $= 48 + 2 = 50$	50_{10}	$\begin{aligned} 1) \frac{50}{2} &= 25 \text{ (остаток 0)} \\ 2) \frac{25}{2} &= 12 \text{ (остаток 1)} \\ 3) \frac{12}{2} &= 6 \text{ (остаток 0)} \\ 4) \frac{6}{2} &= 3 \text{ (остаток 0)} \\ 5) \frac{3}{2} &= 1 \text{ (остаток 1)} \\ 6) \frac{1}{2} &= 0 \text{ (остаток 1)} \end{aligned}$	110010_2
$3B_{16}$	$3 \times 16^1 + B \times 16^0$ $= 3 \times 16^1 + 11$ $\times 16^0 = 48 + 11$ $= 59$	59_{10}	$\begin{aligned} 1) \frac{59}{2} &= 29 \text{ (остаток 1)} \\ 2) \frac{29}{2} &= 14 \text{ (остаток 1)} \\ 3) \frac{14}{2} &= 7 \text{ (остаток 0)} \\ 4) \frac{7}{2} &= 3 \text{ (остаток 1)} \\ 5) \frac{3}{2} &= 1 \text{ (остаток 1)} \\ 6) \frac{1}{2} &= 0 \text{ (остаток 1)} \end{aligned}$	111011_2
51_{16}	$5 \times 16^1 + 1 \times 16^0$ $= 80 + 1 = 93$	81_{10}	$\begin{aligned} 1) \frac{81}{2} &= 40 \text{ (остаток 1)} \\ 2) \frac{40}{2} &= 20 \text{ (остаток 0)} \\ 3) \frac{20}{2} &= 10 \text{ (остаток 0)} \\ 4) \frac{10}{2} &= 5 \text{ (остаток 0)} \\ 5) \frac{5}{2} &= 2 \text{ (остаток 1)} \\ 6) \frac{2}{2} &= 1 \text{ (остаток 0)} \\ 7) \frac{1}{2} &= 0 \text{ (остаток 1)} \end{aligned}$	1010001_2

Мы нашли 8-битный двоичный код для каждого сообщения, который служит ключом для дешифрирования. Далее исходя из условия задачи для дешифровки сообщений требуется модифицированный закрытый ключ, который рассчитывается как каждый элемент закрытого ключа умноженный на n^{-1} и найденный остаток от деления полученного числа на m . В нашем случае закрытый ключ представляет последовательность $\{1, 4, 6, 13, 21, 45, 69, 119\}$, значения n^{-1} и m равны 87 и 280 соответственно.

Модифицированный закрытый ключ:

- 1) (1×87) делится на 280 с остатком 87
- 2) (4×87) делится на 280 с остатком 68
- 3) (6×87) делится на 280 с остатком 242
- 4) (13×87) делится на 280 с остатком 11



5) (21×87) делится на 280 с остатком 147

6) (45×87) делится на 280 с остатком 275

7) (69×87) делится на 280 с остатком 123

8) (119×87) делится на 280 с остатком 273

Модифицированный закрытый ключ {87, 68, 242, 11, 147, 275, 123, 273}.

Найденный ранее 8-битный ключ определяет, какие элементы модифицированного ключа суммируются для получения расшифрованного сообщения. При нехватке элементов 8-битного ключа при выборе элементов модифицированного ключа, недостающие элементы приравниваются к 0). Таким образом, произведем дешифровку принятых в административном управлении сообщений:

Первое сообщение:

$$\begin{array}{cccccccc} 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 87 & 68 & 242 & 11 & 147 & 275 & 123 & 273 \end{array} = 242 + 147 + 275 + 273 = 937$$

Второе сообщение:

$$\begin{array}{cccccccc} 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 87 & 68 & 242 & 11 & 147 & 275 & 123 & 273 \end{array} = 242 + 147 + 275 + 123 = 787$$

Третье сообщение:

$$\begin{array}{cccccccc} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 87 & 68 & 242 & 11 & 147 & 275 & 123 & 273 \end{array} = 242 + 11 + 123 = 376$$

Четвертое сообщение:

$$\begin{array}{cccccccc} 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 87 & 68 & 242 & 11 & 147 & 275 & 123 & 273 \end{array} = 242 + 11 + 147 + 123 + 273 = 796$$

Пятое сообщение:

$$\begin{array}{cccccccc} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 87 & 68 & 242 & 11 & 147 & 275 & 123 & 273 \end{array} = 68 + 11 + 273 = 352$$

В ответе необходимо указать последовательность пяти чисел через запятую, начиная с номера склада, где хранился добытый уран в понедельник, заканчивая номером склада, где хранился добытый уран в пятницу. Следовательно, ответ задачи выглядит следующим образом 937, 787, 376, 796, 352.

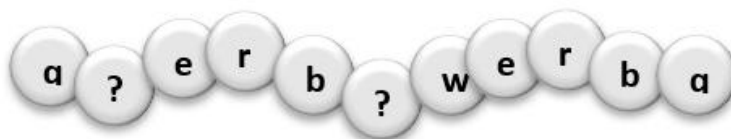
Ответ: 937, 787, 376, 796, 352



Задача 5 (25 баллов)

Задача об удивительных последовательностях

Профессор Тимофей Чудоген, известный биолог, проводит интересное исследование, в котором изучает особый искусственный генетический код. Этот код состоит из последовательности искусственных нуклеотидов, которые обозначаются строчными латинскими буквами. Однако в этой последовательности есть вопросы — знаки, которые нужно заменить на определенные буквы, чтобы завершить анализ.



Профессор считает, что этот код имеет удивительную особенность, которую он называет А-периодичность. Периодичность означает, что если взять два искусственных нуклеотида, расположенных на расстоянии, кратном заданному числу А, то они должны быть одинаковыми. Это значит, что все нуклеотиды, стоящие на тех же позициях, кратных А, должны совпадать.

Профессор Чудоген уверен, что понимание этих особенностей поможет в будущем в таких науках, как медицина, экология и биология. Его задача заключается в том, чтобы найти правильные буквы для замены знаков вопроса и определить можно ли соответствующую строку сделать А-периодичной путем замены всех знаков вопроса. Помогите профессору.

Формат входных данных

В первой строке содержится два числа через пробел: N — общее количество строк, которые вам нужно проверить на А-периодичность и число А ($1 \leq A \leq 5 \cdot 10^3$). Далее содержится N непустых строк, состоящих из малых букв латиницы и знаков вопроса $1 \leq N \leq 5 \cdot 10^4$, суммарная длина всех строк не превосходит $2 \cdot 10^5$.

Формат выходных данных

Вывести N строк, в каждой из которых вывести либо "YES", если соответствующую строку можно сделать А-периодичной путем замены всех знаков вопроса на некоторые буквы и "NO" в противном случае. Будем считать последовательность А-периодичной, если для любых двух символов, расстояние между которыми кратно А верно, что они совпадают.



Примеры

стандартный ввод	стандартный вывод
8 1	NO
abacabaca	NO
abracadabra	YES
aa?aaa?aaaaaaaaaaa	YES
q	YES
???????	NO
q?er?w?	NO
q?erw??	NO
q?er?wer?werw?er	

Решите указанную задачу, используя один из языков программирования, и укажите, какой именно язык был выбран. Предложите эффективное решение, учитывая минимизацию времени выполнения программы и объема памяти, необходимого для её работы. Постарайтесь использовать алгоритмы и структуры данных, которые обеспечивают оптимальную производительность.

Реализуйте решение в виде программного кода, обязательно добавив комментарии, которые поясняют ключевые шаги алгоритма и переменные, если это важно для понимания проверяющему. Допускается использование только стандартных библиотек, встроенных в язык. Подробно опишите алгоритм решения, предоставив текстовое описание или блок-схему, объяснив логику работы программы.

Приведите результат выполнения программы для заданных исходных данных. Результат выполнения для указанных исходных данных должен включать вывод программы, а также, если соответствующую строку можно сделать A-периодичной заменой всех знаков вопроса, привести полученную строку.

Исходные данные для выполнения задания

стандартный ввод	стандартный вывод
2 5 ad?fg?def?ad?f mn?a?c?noabc?n	?

Ваше решение должно быть аккуратно оформлено, читабельно и соответствовать стандартам выбранного языка программирования.

Решение

Чтение входных данных

Программа сначала считывает количество строк и значение, которое определяет период, по которому будет происходить проверка.

Инициализация

Для каждой строки создается флаг ok, который по умолчанию устанавливается в true. Этот флаг отслеживает, соответствует ли текущая



строка необходимым критериям А-периодичности.

Проверка А-периодичности

Для каждого индекса от 0 до $p-1$ (внешний цикл) инициализируется символ s , устанавливаемый в '#'. Он будет использоваться для отслеживания первого ненулевого символа в строке.

Внутренний цикл проходит по строке с шагом p , начиная с текущего индекса. Если текущий символ не равен '?', проверяется:

Если s все еще '#', это значит, что найден первый ненулевой символ, и s устанавливается на этот символ.

Если s уже установлен, проверяется, совпадает ли текущий символ с s . Если нет — флаг ok устанавливается в `false`, и цикл прерывается.

Сохранение результата

После завершения проверки для каждой строки результат ("YES" или "NO") добавляется в список `results`.

Вывод результата

В конце программа выводит результаты всех тестов.

Реализация на C++

```
#include <bits/stdc++.h>
#define int long

using namespace std;

signed main() {
    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);

    int u, p;
    cin >> u >> p;
    vector<string> results; // Вектор для хранения результатов

    while (u--) {
        string s;
        cin >> s;
        bool ok = true; // Флаг, отслеживает правильность строк

        for (int i = 0; i < p; i++) {
            char c = '#'; // Устанавливаем символ по умолчанию
            for (int j = 0; i + j < (int)(s).size(); j += p) {
                if (s[i + j] != '?') {
                    if (c == '#') {
                        c = s[i + j]; // Устанавливаем 1-ый ненулевой
                    } else {
                        if (c != s[i + j]) {
                            ok = false; // Устанавливаем флаг в false
                        }
                    }
                }
            }
        }

        if (ok) {
            results.push_back("YES"); // Сохраняем результат
        } else {
            results.push_back("NO"); // Сохраняем результат
        }
    }
}
```




```
    }  
}  
  
// Выводим все результаты после завершения всех проверок  
for (int i = 0; i < (int) (results).size(); i++) {  
    cout << results[i] << "\n";  
}  
}
```

Реализация на Python

```
# Считываем с клавиатуры количество строк и значение p  
u, p = map(int, input().split())  
  
# Создаём список для хранения результатов  
results = []  
  
# Обрабатываем u строк  
for _ in range(u):  
    s = input().strip() # Считываем текущую строку  
    ok = True # Флаг, указывающий на A-периодичность  
  
    # Проверяем A-периодичность строки  
    for i in range(p):  
        c = '#' # Инициализируем переменную для отслеживания символа  
        for j in range(i, len(s), p):  
            if s[j] != '?':  
                if c == '#':  
                    c = s[j] # Устанавливаем 1-ый ненулевой  
                elif c != s[j]:  
                    ok = False # Устанавливаем флаг в False  
                    break  
  
        if not ok:  
            break  
  
    # Сохраняем результат в список  
    if ok:  
        results.append("YES")  
    else:  
        results.append("NO")  
  
# Выводим все результаты после завершения обработки строк  
for result in results:  
    print(result)
```

Временную сложность алгоритма можно определить следующим образом:

Пусть длина строки n . Внутренний цикл выполняется для каждого i от 0 до $p-1$. В каждой итерации внутреннего цикла мы проходим элементы строки с шагом p . Таким образом, количество итераций внутреннего цикла составляет примерно n/p для каждого i . В итоге, временная сложность будет: $O(u \times (p + n/p))$, где u — количество строк для проверки. Так как n значительно больше p , то можно упростить до: $O(u \times n)$.

Основные потребности в памяти включают вектор для хранения результатов и саму строку.



Результат выполнения алгоритма для указанных исходных данных

стандартный ввод	стандартный вывод
2 5 ad?fg?def?ad?f mn?a?c?noabc?n	YES NO

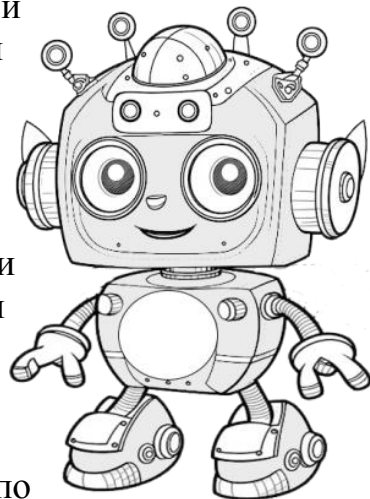
Для первого слова можем получить A-периодичную последовательность **adefgadefgadef**.



Задача 6 (25 баллов)

РобоМакс и тайна цветных деталей

На производственном предприятии «Механический Мастер» функционирует длинный конвейер, на котором детали пронумерованы от 1 до N (где $N \leq 10^5$). Эти детали движутся по кругу, таким образом, перед первой деталью оказывается деталь с номером N , а перед любой другой — деталь с номером, на единицу меньшим. Детали с четными номерами предназначены для зеленых машин и окрашены в яркий зеленый цвет, в то время как нечетные детали окрашены в насыщенный синий цвет и предназначены для синих машин.



На этом заводе трудится наш герой — робот по имени РобоМакс. РобоМакс — маленький, но невероятно умный робот с большими глазами и яркой светодиодной панелью. Его задача состоит в том, чтобы забирать детали для дальнейшего производства. Однако у него есть особое правило: он может забрать только те детали, у которых обе соседние детали (та, что перед ней, и та, что за ней) имеют одинаковый цвет.

РобоМакс внимательно осматривает конвейер. Если обе соседние детали имеют одинаковый цвет, он забирает центральную деталь, в то время как соседние продолжают двигаться по кругу, не меняя своих положений. Если же одна деталь зеленая, а другая синяя, РобоМакс не может забрать нужную деталь, и она остается на месте, вызывая у него легкую грусть.

Представьте себе, как быстро и эффективно работает РобоМакс, вызывая восхищение рабочих. Однако постепенно становится заметно, что некоторые детали остаются без внимания. Теперь возникает вопрос: сколько деталей останется на конвейере после нескольких успешных операций РобоМакса? Ваша задача заключается в том, чтобы помочь ему выяснить, сколько деталей в итоге останется на конвейере, учитывая правила его работы.

Формат входных данных

В первой строке два натуральных числа N и K ($3 \leq N \leq 50000$, $1 \leq K \leq N-2$). Во второй строке через пробел K чисел, каждое из интервала от 1 до N — номера деталей в том порядке, в котором их пытается взять РобоМакс. Если детали с указанным номером к моменту, когда за ней приходит РобоМакс, уже нет, ничего не происходит.

Формат выходных данных



Одно число — ответ на вопрос.

Примеры

стандартный ввод	стандартный вывод
13 11 1 2 1 13 2 6 12 3 13 5 3	8

Решите указанную задачу, используя один из языков программирования, и укажите, какой именно язык был выбран. Предложите эффективное решение, учитывая минимизацию времени выполнения программы и объема памяти, необходимого для её работы. Постарайтесь использовать алгоритмы и структуры данных, которые обеспечивают оптимальную производительность.

Реализуйте решение в виде программного кода, обязательно добавив комментарии, которые поясняют ключевые шаги алгоритма и переменные, если это важно для понимания проверяющему. Допускается использование только стандартных библиотек, встроенных в язык. Подробно опишите алгоритм решения, предоставив текстовое описание или блок-схему, объяснив логику работы программы.

Приведите результат выполнения программы для заданных исходных данных.

Исходные данные для выполнения задания

стандартный ввод	стандартный вывод
12 11 1 4 3 12 10 9 8 1 4 7 9	?

Ваше решение должно быть аккуратно оформлено, читабельно и соответствовать стандартам выбранного языка программирования.

Решение

Чтение входных данных

Сначала считываются значения n (количество деталей) и k (количество операций).

Инициализация массивов

Создаются два массива `pred` и `next`, которые представляют индексы предыдущей и следующей деталей соответственно. Массив `lab` инициализируется единицами, показывая, что изначально все детали находятся на конвейере.

Обработка операций



Считывается номер детали и проверяется, можно ли забрать деталь по заданным правилам. Если обе соседние детали одного цвета и текущая деталь не была забрана, она забирается, и связи между соседними деталями обновляются.

Подсчет оставшихся деталей

В конце программы производится подсчет оставшихся на конвейере деталей, после чего результат выводится на экран.

Реализация на C++

```
#include<bits/stdc++.h>
#define int long
using namespace std;
int n, k, a; // Объявляем глобальные переменные: n (количество
деталей), k (количество запросов), a (номер детали запроса)

signed main(){
    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
    // Чтение входных данных: количество деталей и количество
запросов
    cin >> n >> k;
    // Массивы для предшествующих, последующих и меток деталей
    vector<int> pred(n), next(n), lab(n, 1);

    // Заполняем массив предшествующих, указывая,
    // что предыдущая деталь у первой – последняя
    for(int i = 0; i < n; i++) pred[i] = (i-1 + n) % n;
    // Заполняем массив последующих, указывая,
    // что следующая деталь у последней – первая
    for(int i = 0; i < n; i++) next[i] = (i+1) % n;
    // Обработка запросов РобоМакса
    for(int i = 0; i < k; i++){
        cin >> a; // Читаем номер детали
        a--;
        // Переход к нулевой индексации (т.к. вводятся номера от 1 до N)
        if(pred[a]%2 == next[a]%2 && lab[a] == 1){
            lab[a] = 0; // Помечаем деталь как убранную
            next[pred[a]] = next[a];
            // Обновляем следующую деталь для предшествующей
            pred[next[a]] = pred[a];
            // Обновляем предшествующую для следующей детали
        }
    }
    // Подсчет оставшихся деталей на конвейере
    int ans = 0;
    for(auto q : lab) ans += q;
    cout << ans; // Выводим результат
}
```

Реализация на Python

```
def main():
    n, k = map(int, input().split())
```



```
# Создаем массивы предшествующих и последующих деталей
pred = [(i - 1 + n) % n for i in range(n)]
next = [(i + 1) % n for i in range(n)]
# Массив, забрана ли деталь (1 - нет, 0 - да)
lab = [1] * n
for _ in range(k):
    # Индекс детали, которую РобоМакс собирается проверить
    a = int(input()) - 1
    # Проверяем, могут ли соседние детали быть одного цвета
    if pred[a] % 2 == next[a] % 2 and lab[a] == 1:
        lab[a] = 0 # Забираем деталь
        next[pred[a]] = next[a] # меняем соседние
        pred[next[a]] = pred[a]
# Подсчитываем количество оставшихся деталей
ans = sum(lab)
print(ans)

if __name__ == "__main__":
    main()
```

Давайте проанализируем временную и пространственную сложность решения. Создание массивов `pred`, `next` и `lab` требует $O(n)$ времени, поскольку мы проходимся по всем деталям. Цикл по запросам выполняется k раз. время выполнения всей программы будет суммироваться: $O(n)+O(k)=O(n+k)$.

Мы используем три массива: `pred`, `next` и `lab`, каждый из которых занимает $O(n)$ пространства. Следовательно, потребление памяти для этих массивов составляет $O(n)$.

Результат выполнения алгоритма для указанных исходных данных

стандартный ввод	стандартный вывод
12 11 1 4 3 12 10 9 8 1 4 7 9	7



ВАРИАНТ 3

Задача 1 (10 баллов)

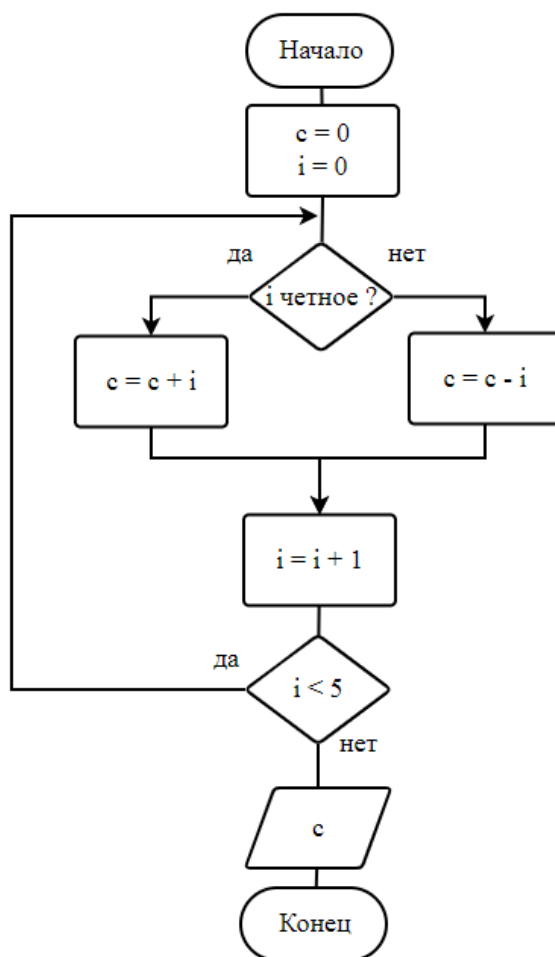
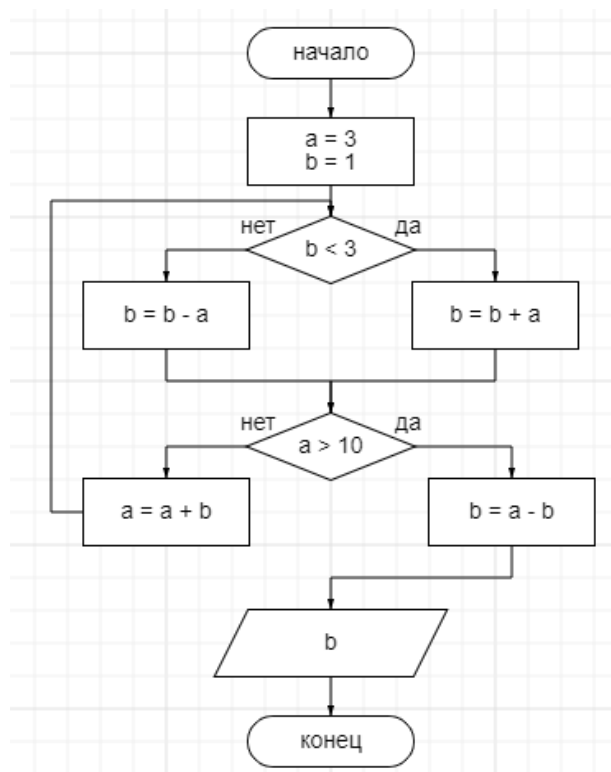
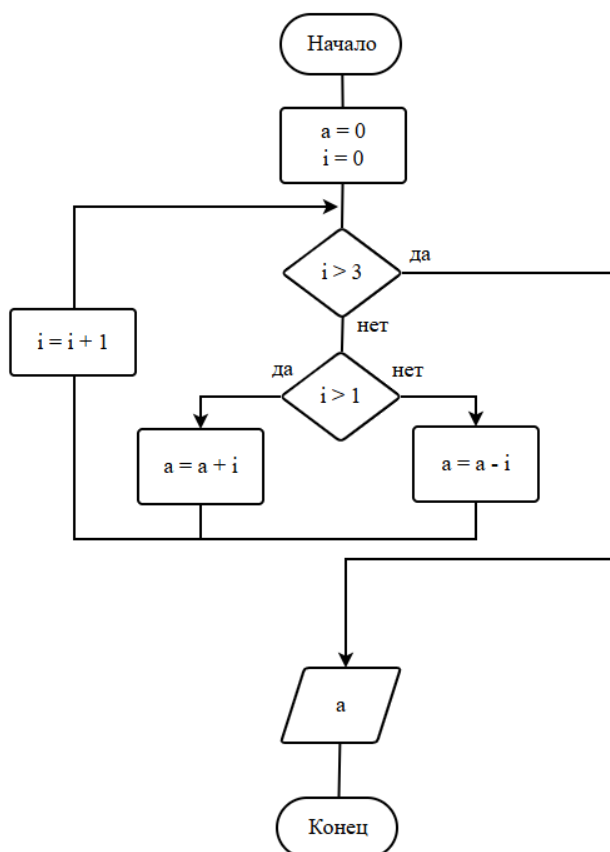
Разведчики

Нашими разведчиками на линии боевого соприкосновения были обнаружены средства противника, сведения о которых они передали в виде закодированной электронной таблицы на командный пункт. Определите количество этих технических средств, значение которых расположены в ячейках B2:F2. В ответе запишите значения ячеек в одну строку, через запятую, начиная с B2 и заканчивая F2.

	A	B	C	D	E	F
		Танки	БМП	Орудия	Установки РЭБ	Опорные пункты
1						
2						
3						
4	a =				1) =СЧЁТЕСЛИ(B2:F2;B4)	
5	b =				2) =СЧЁТЕСЛИ(E2;B4)+B5	
6	c =				3) =СУММЕСЛИ(D2:E2;B4)-B6	
7	d =				4) =СУММ(D2:F2)-B6	
8					5) =ЕСЛИ(B7>5;B2-C2;СУММ(B2;C2))	
9					6) =ЕСЛИ(B7<=5;B2-C2;СУММ(B2;C2))	
10						

	A	B	C	D	E	F
		Танки	БМП	Орудия	Установки РЭБ	Опорные пункты
1						
2						
3						
4	a =				1)	2
5	b =				2)	17
6	c =				3)	6
7	d =				4)	9
8					5)	15
9					6)	-3
10						

Значения a, b и c определите, выполнив соответствующие алгоритмы по блок-схемам, d (ячейка B7) – неизвестное натуральное число.





Решение

Определяем значения коэффициентов a , b , c выполнив соответствующие алгоритмы по представленным блок-схемам. В результате получаем следующие значения.

	A	B	C
3			
4	a =	4	
5	b =	16	
6	c =	2	
7	d =		
8			

1) Значение формулы в ячейке E4 «=СЧЁТЕСЛИ(B2:F2;B4)» равно 2, следовательно в ячейках B2:F2 находятся две 4.

2) Значение формулы в ячейке E5 «=СЧЁТЕСЛИ(E2;B4)+B5» равно 17, откуда значение «=СЧЁТЕСЛИ(E2;B4)» равно $17 - B5 = 17 - 16 = 1$. Следовательно в ячейке E2 находится 4.

3) Значение формулы в ячейке E6 «=СУММЕСЛИ(D2:E2;B4)-B6» равно 6, откуда значение «=СУММЕСЛИ(D2:E2;B4)» равно $6 + B6 = 6 + 2 = 8$. Следовательно в ячейке D2, также как и в E2, находится значение 4.

4) Значение формулы в ячейке E7 «=СУММ(D2:F2)-B6» равно 9, откуда значение «=СУММ(D2:F2)» равно $9 + B6 = 9 + 2 = 11$. Значения ячеек D2 и E2 известны, находим значение ячейки F2. $F2 = 11 - 4 - 4 = 3$.

5) Анализируя формулы в ячейке E8 «=ЕСЛИ(B7>5;B2-C2;СУММ(B2;C2))» и ячейке E9 «=ЕСЛИ(B7<=5;B2-C2;СУММ(B2;C2))» понимаем, что если значение в ячейке B7 будет больше 5, то в ячейке E8 будет $B2 - C2$, а в ячейке E9: $B2 + C2$, в противном случае в ячейке E8 будет $B2 + C2$, а в ячейке E9: $B2 - C2$. Так как, по условию, значения в ячейках B2:F2 натуральные числа (количество объектов), то разность всегда меньше суммы, получаем:

$$\begin{cases} B2 + C2 = 15 \\ B2 - C2 = -3 \end{cases} \Rightarrow \begin{cases} C2 - 3 + C2 = 15 \\ B2 = C2 - 3 \end{cases} \Rightarrow \begin{cases} C2 = 9 \\ B2 = 6 \end{cases}$$

Ответ: 6, 9, 4, 4, 3.



Задача 2 (10 баллов)

Чертёжник

Исполнитель Чертёжник перемещается на координатной плоскости, оставляя след в виде линии. Чертёжник может выполнять следующие команды:

1) Сместиться на (a, b) , где a, b – целые числа. Эта команда перемещает исполнителя из точки с координатами (x, y) в точку с координатами $(x + a; y + b)$. Например, если Чертёжник находится в точке с координатами $(1, 2)$, то команда сместиться на $(2, -1)$ переместит Чертёжника в точку $(3, 1)$.

2) Цикл:

ПОКА <условие>

последовательность команд

КОНЕЦ ПОКА

Этот цикл означает, что последовательность команд будет выполняться до тех пор, пока условие <условие> выполняется.

Исполнителю Чертёжник был дан для исполнения следующий алгоритм (буквами a, b обозначены неизвестные числа):

НАЧАЛО

сместиться на $(2, -2)$

ПОКА $x + y$ не равно 4

сместиться на (a, b)

сместиться на $(-6.5, 2.5)$

КОНЕЦ ПОКА

сместиться на $(-7, 7)$

КОНЕЦ

После выполнения программы смещение Чертёжника по оси y будет в три раза меньше смещения по оси x . Сколько раз выполнялась последовательность команд цикла ПОКА, если известно, что команды цикла ПОКА были выполнены более одного раза и сумма значений координат по осям x и y до выполнения программы была равна 0

Решение

Значение координат Чертёжника в результате выполнения цикла ПОКА:

$$x_n = x_0 + \Delta x = x_0 + 2 + N(a - 1.5)$$

$$y_n = y_0 + \Delta y = y_0 - 3 + N(b + 5.5)$$

В этом уравнении N – количество повторений команд цикла ПОКА, которое является искомым значением.

Из условия завершения цикла $x_n + y_n = 4$, получим:

$$x_0 + 2 + N(a - 6.5) + y_0 - 2 + N(b + 2.5) = 4$$

Из условия $y_0 + x_0 = 0$, получим:



$$N(a - b - 4) - 4 = 0$$

Общее изменение координат Чертёжника в результате выполнения этого алгоритма:

$$x - x_0 = 2 + N(a - 1,5) - 7$$

$$y - y_0 = -2 + N(b + 5,5) + 7$$

Так как изменение координат по оси x будет превышать изменение координат по оси y в три раза, получим:

$$(2 + N(a - 1,5) - 7) = 3(-2 + N(b + 5,5) + 7)$$

$$N(a - 6,5) - 5 = N(3b + 7,5) + 15$$

$$N(-a + 3b + 14) = 20$$

Получим систему уравнений:

$$\begin{cases} N(a + b - 4) = 4 \\ N(-a + 3b + 14) = 20 \end{cases}$$

Нужно найти натуральное N , при котором эта система уравнений разрешима в целых числах относительно a и b . Для этого число N должно быть одновременно делителем чисел 4 и 20. Общими делителями этих чисел являются числа 1, 2 и 4, при этом 1 не подходит по условию, что цикл повторялся более 1 раза, а число 2 не подходит из условия, что a, b – целые числа, при подстановке в систему уравнений выше.

Ответ: последовательность команд цикла ПОКА выполнилась 4 раза.

**Задача 3 (10 баллов)****Шкатулка**

Николай купил новую цифровую шкатулку с четырехзначным кодом. Он придумал код. Чтобы лучше его запомнить, он решил преобразовать каждую цифру этого 4-значного числа в двоичный код. Далее он составил логическое выражение. В результирующем столбце таблицы истинности к этому выражению отобразилась та самая двоичная последовательность для всего кода.

$$(a \wedge b) \rightarrow (\neg b \wedge c) \equiv (a \vee \neg c)$$

Строки в таблице истинности идут последовательно, 000, 001, 010 и т.д.

Какой шифр на шкатулке у Николая, если известно, что первые 2 цифры одинаковые?

Решение

Составим таблицу истинности

a	b	c	$a \wedge b$	$\neg b$	$\neg b \wedge c$	$a \wedge b \rightarrow \neg b \wedge c$	$\neg c$	$a \vee \neg c$	$a \wedge b \rightarrow \neg b \wedge c \equiv (a \vee \neg c)$
0	0	0	0	1	0	1	1	1	1
0	0	1	0	1	1	1	0	0	0
0	1	0	0	0	0	1	1	1	1
0	1	1	0	0	0	1	0	0	0
1	0	0	0	1	0	1	1	1	1
1	0	1	0	1	1	1	0	1	1
1	1	0	1	0	0	0	1	1	0
1	1	1	1	0	0	0	0	1	0

Двоичный код в последнем столбце можно разбить на двоичные числа следующим образом: $10_2 10_2 11_2 100_2$

Таким образом получаем следующий десятичный код: 2 2 1 4

Ответ: 2 2 1 4

**Задача 4 (20 баллов)****Добыча урана**

Шахта «Уран-2» использует систему шифрования для передачи сообщения о добыче урана в головную организацию. Для дешифровки данного сообщения каждый элемент закрытого ключа, хранящегося в головной организации, умножается на n^{-1} – мультипликативное обратное к значению n по модулю m (остаток от деления $n \cdot n^{-1}$ на m равен 1). Результат вычисляется как остаток от деления полученного числа на m . Полученный модифицированный закрытый ключ используется для дешифрования, а 8-битный ключ определяет, какие элементы модифицированного ключа суммируются для получения расшифрованного сообщения.

В понедельник добыча урана на шахте составила $4A_{16}$ килограмм. Ежедневно добываемая масса урана увеличивалась согласно арифметической прогрессии, состоящей из последовательности натуральных чисел (например, количество добытого урана во второй день равно $4A_{16} + 1_{16}$ кг и т.д.). С шахты в головную организацию в течение рабочей недели поступило пять сообщений с количеством добытого урана, которые преобразовывались в головной организации в 8-битный двоичный код и служили ключом для дешифрования. Закрытый ключ, хранящийся в организации, представляет собой следующую последовательность $\{2, 4, 7, 13, 22, 36, 60, 98\}$, значения n^{-1} и m равны 53 и 245 соответственно.

Расшифруйте полученные сообщения, зная, что они представляют номер склада, где хранится добытый уран в каждый из дней рабочей недели. В ответе укажите последовательность пяти чисел через запятую, начиная с номера склада, где хранился добытый уран в понедельник, заканчивая номером склада, где хранился добытый уран в пятницу.

Решение

На первом этапе необходимо определить сообщения, которые были переданы в административное управление со вторника по пятницу. В условии сказано, что ежедневно добываемая масса урана (шифротекст) увеличивалась согласно арифметической прогрессии, состоящей из последовательности натуральных чисел, т.е. во второй день количество добытого урана увеличилось на 1_{16} кг, в третий день – на 2_{16} кг, в четвертый день – на 3_{16} кг и в пятый день – на 4_{16} кг. Для нахождения переданных сообщений произведем сложение чисел в шестнадцатеричной системе.

Сложение чисел в шестнадцатеричной системе счисления выполняется поразрядно, начиная с младшего разряда. Если при сложении двух чисел одинакового разряда получается двузначное число, то значение его старшего разряда (единицу) добавляют в старший разряд. Следовательно:

$$\text{Второе сообщение: } 4A_{16} + 1_{16} \Rightarrow A + 1 = B \Rightarrow 4B_{16}$$



Третье сообщение: $4B_{16} + 2_{16} \Rightarrow B + 2 = D \Rightarrow 4D_{16}$

Четвертое сообщение: $4D_{16} + 3_{16} \Rightarrow D + 3 = 10 \Rightarrow 4 + 1 \Rightarrow 50_{16}$

Пятое сообщение: $50_{16} + 4_{16} \Rightarrow 0 + 4 = 4 \Rightarrow 54_{16}$

Таким образом, последовательность переданных сообщений в течение рабочей недели: $4A_{16}, 4B_{16}, 4D_{16}, 50_{16}, 54_{16}$

Далее необходимо найти ключ для дешифрования каждого сообщения. По условию задачи он представляет собой преобразованное значение переданного сообщения в 8-битном двоичном виде. Для перевода числа из шестнадцатеричной системы счисления в двоичную можно воспользоваться несколькими способами, мы воспользуемся переводом сначала в десятичную систему счисления, затем в двоичную.

Каждую цифру шестнадцатеричного числа умножим на 16 в степени, равной позиции этой цифры, и затем сложим все полученные произведения. Далее разделим полученное число в десятичной системе на 2 и запишем остаток до тех пор, пока неполное частное не будет равно нулю. Полученные остатки записываем в обратном порядке.

Число в 16-ой системе счисления	Перевод в 10-ую систему счисления	Число в 10-ой системе счисления	Перевод в 2-ую систему счисления	Число в 2-ой системе счисления
$4A_{16}$	$4 \times 16^1 + A \times 16^0$ $= 4 \times 16^1 + 10$ $\times 16^0 = 64 + 10$ $= 74$	74_{10}	$1) \frac{74}{2} = 37$ (остаток 0) $2) \frac{37}{2} = 18$ (остаток 1) $3) \frac{18}{2} = 9$ (остаток 0) $4) \frac{9}{2} = 4$ (остаток 1) $5) \frac{4}{2} = 2$ (остаток 0) $6) \frac{2}{2} = 1$ (остаток 0) $7) \frac{1}{2} = 0$ (остаток 1)	1001010_2
$4B_{16}$	$4 \times 16^1 + B \times 16^0$ $= 4 \times 16^1 + 11$ $\times 16^0 = 64 + 11$ $= 75$	75_{10}	$1) \frac{75}{2} = 37$ (остаток 1) $2) \frac{37}{2} = 18$ (остаток 1) $3) \frac{18}{2} = 9$ (остаток 0) $4) \frac{9}{2} = 4$ (остаток 1) $5) \frac{4}{2} = 2$ (остаток 0) $6) \frac{2}{2} = 1$ (остаток 0) $7) \frac{1}{2} = 0$ (остаток 1)	1001011_2



$4D_{16}$	$4 \times 16^1 + D \times 16^0$ $= 64 + 13 = 77$	77_{10}	$\begin{array}{l} 1) \frac{77}{2} = 38 \text{ (остаток 1)} \\ 2) \frac{38}{2} = 19 \text{ (остаток 0)} \\ 3) \frac{19}{2} = 9 \text{ (остаток 1)} \\ 4) \frac{9}{2} = 4 \text{ (остаток 1)} \\ 5) \frac{4}{2} = 2 \text{ (остаток 0)} \\ 6) \frac{2}{2} = 1 \text{ (остаток 0)} \\ 7) \frac{1}{2} = 0 \text{ (остаток 1)} \end{array}$	1001101_2
50_{16}	$5 \times 16^1 + 0 \times 16^0$ $= 80 + 0 = 80$	80_{10}	$\begin{array}{l} 1) \frac{80}{2} = 40 \text{ (остаток 0)} \\ 2) \frac{40}{2} = 20 \text{ (остаток 0)} \\ 3) \frac{20}{2} = 10 \text{ (остаток 0)} \\ 4) \frac{10}{2} = 5 \text{ (остаток 0)} \\ 5) \frac{5}{2} = 2 \text{ (остаток 1)} \\ 6) \frac{2}{2} = 1 \text{ (остаток 0)} \\ 7) \frac{1}{2} = 0 \text{ (остаток 1)} \end{array}$	1010000_2
54_{16}	$5 \times 16^1 + 4 \times 16^0$ $= 80 + 4 = 84$	84_{10}	$\begin{array}{l} 1) \frac{84}{2} = 42 \text{ (остаток 0)} \\ 2) \frac{42}{2} = 21 \text{ (остаток 0)} \\ 3) \frac{21}{2} = 10 \text{ (остаток 1)} \\ 4) \frac{10}{2} = 5 \text{ (остаток 0)} \\ 5) \frac{5}{2} = 2 \text{ (остаток 1)} \\ 6) \frac{2}{2} = 1 \text{ (остаток 0)} \\ 7) \frac{1}{2} = 0 \text{ (остаток 1)} \end{array}$	1010100_2

Мы нашли 8-битный двоичный код для каждого сообщения, который служит ключом для дешифрирования. Далее исходя из условия задачи для дешифровки сообщений требуется модифицированный закрытый ключ, который рассчитывается как каждый элемент закрытого ключа умноженный на n^{-1} и найденный остаток от деления полученного числа на m . В нашем случае закрытый ключ представляет последовательность $\{2, 4, 7, 13, 22, 36, 60, 98\}$, значения n^{-1} и m равны 53 и 245 соответственно.

Модифицированный закрытый ключ:

- 1) (2×53) делится на 245 с остатком 106
- 2) (4×53) делится на 245 с остатком 212



- 3) (7×53) делится на 245 с остатком 126
- 4) (13×53) делится на 245 с остатком 199
- 5) (22×53) делится на 245 с остатком 186
- 6) (36×53) делится на 245 с остатком 193
- 7) (60×53) делится на 245 с остатком 240
- 8) (98×53) делится на 245 с остатком 49

Модифицированный закрытый ключ {106, 212, 126, 199, 186, 193, 240, 49}.

Найденный ранее 8-битный ключ определяет, какие элементы модифицированного ключа суммируются для получения расшифрованного сообщения. При нехватке элементов 8-битного ключа при выборе элементов модифицированного ключа, недостающие элементы приравниваются к 0). Таким образом, произведем дешифровку принятых в административном управлении сообщений:

Первое сообщение:

$$\begin{array}{cccccccc} 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 106 & 212 & 126 & 199 & 186 & 193 & 240 & 49 \end{array} = 212 + 186 + 240 = 638$$

Второе сообщение:

$$\begin{array}{cccccccc} 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 106 & 212 & 126 & 199 & 186 & 193 & 240 & 49 \end{array} = 212 + 186 + 24 + 49 = 687$$

Третье сообщение:

$$\begin{array}{cccccccc} 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 106 & 212 & 126 & 199 & 186 & 193 & 240 & 49 \end{array} = 212 + 186 + 193 + 49 = 640$$

Четвертое сообщение:

$$\begin{array}{cccccccc} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 106 & 212 & 126 & 199 & 186 & 193 & 240 & 49 \end{array} = 212 + 199 = 411$$

Пятое сообщение:

$$\begin{array}{cccccccc} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 106 & 212 & 126 & 199 & 186 & 193 & 240 & 49 \end{array} = 212 + 199 + 193 = 604$$

В ответе необходимо указать последовательность пяти чисел через запятую, начиная с номера склада, где хранился добытый уран в понедельник, заканчивая номером склада, где хранился добытый уран в пятницу. Следовательно, ответ задачи выглядит следующим образом 638, 687, 640, 411, 604.

Ответ: 638, 687, 640, 411, 604



Задача 5 (25 баллов)

Задача об удивительных последовательностях

Профессор Тимофей Чудоген, известный биолог, проводит интересное исследование, в котором изучает особый искусственный генетический код. Этот код состоит из последовательности искусственных нуклеотидов, которые обозначаются строчными латинскими буквами. Однако в этой последовательности есть вопросы — знаки, которые нужно заменить на определенные буквы, чтобы завершить анализ.



Профессор считает, что этот код имеет удивительную особенность, которую он называет А-периодичность. Периодичность означает, что если взять два искусственных нуклеотида, расположенных на расстоянии, кратном заданному числу А, то они должны быть одинаковыми. Это значит, что все нуклеотиды, стоящие на тех же позициях, кратных А, должны совпадать.

Профессор Чудоген уверен, что понимание этих особенностей поможет в будущем в таких науках, как медицина, экология и биология. Его задача заключается в том, чтобы найти правильные буквы для замены знаков вопроса и определить можно ли соответствующую строку сделать А-периодичной путем замены всех знаков вопроса. Помогите профессору.

Формат входных данных

В первой строке содержится два числа через пробел: N — общее количество строк, которые вам нужно проверить на А-периодичность и число А ($1 \leq A \leq 5 \cdot 10^3$). Далее содержится N непустых строк, состоящих из малых букв латиницы и знаков вопроса $1 \leq N \leq 5 \cdot 10^4$, суммарная длина всех строк не превосходит $2 \cdot 10^5$.

Формат выходных данных

Вывести N строк, в каждой из которых вывести либо "YES", если соответствующую строку можно сделать А-периодичной путем замены всех знаков вопроса на некоторые буквы и "NO" в противном случае. Будем считать последовательность А-периодичной, если для любых двух символов, расстояние между которыми кратно А верно, что они совпадают.



Примеры

стандартный ввод	стандартный вывод
8 1	NO
abacabaca	NO
abracadabra	YES
aa?aaa?aaaaaaaaaaaa	YES
q	YES
???????	NO
q?er?w?	NO
q?erw??	NO
q?er?wer?werw?er	

Решите указанную задачу, используя один из языков программирования, и укажите, какой именно язык был выбран. Предложите эффективное решение, учитывая минимизацию времени выполнения программы и объема памяти, необходимого для её работы. Постарайтесь использовать алгоритмы и структуры данных, которые обеспечивают оптимальную производительность.

Реализуйте решение в виде программного кода, обязательно добавив комментарии, которые поясняют ключевые шаги алгоритма и переменные, если это важно для понимания проверяющему. Допускается использование только стандартных библиотек, встроенных в язык. Подробно опишите алгоритм решения, предоставив текстовое описание или блок-схему, объяснив логику работы программы.

Приведите результат выполнения программы для заданных исходных данных. Результат выполнения для указанных исходных данных должен включать вывод программы, а также, если соответствующую строку можно сделать A-периодичной заменой всех знаков вопроса, привести полученную строку.

Исходные данные для выполнения задания

стандартный ввод	стандартный вывод
2 6 xy?w?xyz?oxyz? mn?a?c?noabc?n	?

Ваше решение должно быть аккуратно оформлено, читабельно и соответствовать стандартам выбранного языка программирования.

Решение

Чтение входных данных

Программа сначала считывает количество строк и значение, которое определяет период, по которому будет происходить проверка.

Инициализация

Для каждой строки создается флаг ok, который по умолчанию устанавливается в true. Этот флаг отслеживает, соответствует ли текущая



строка необходимым критериям А-периодичности.

Проверка А-периодичности

Для каждого индекса от 0 до $p-1$ (внешний цикл) инициализируется символ s , устанавливаемый в '#'. Он будет использоваться для отслеживания первого ненулевого символа в строке.

Внутренний цикл проходит по строке с шагом p , начиная с текущего индекса. Если текущий символ не равен '?', проверяется:

Если s все еще '#', это значит, что найден первый ненулевой символ, и s устанавливается на этот символ.

Если s уже установлен, проверяется, совпадает ли текущий символ с s . Если нет — флаг ok устанавливается в `false`, и цикл прерывается.

Сохранение результата

После завершения проверки для каждой строки результат ("YES" или "NO") добавляется в список `results`.

Вывод результата

В конце программа выводит результаты всех тестов.

Реализация на C++

```
#include <bits/stdc++.h>
#define int long

using namespace std;

signed main() {
    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);

    int u, p;
    cin >> u >> p;
    vector<string> results; // Вектор для хранения результатов

    while (u--) {
        string s;
        cin >> s;
        bool ok = true; // Флаг, отслеживает правильность строк

        for (int i = 0; i < p; i++) {
            char c = '#'; // Устанавливаем символ по умолчанию
            for (int j = 0; i + j < (int)(s).size(); j += p) {
                if (s[i + j] != '?') {
                    if (c == '#') {
                        c = s[i + j]; // Устанавливаем 1-ый ненулевой
                    } else {
                        if (c != s[i + j]) {
                            ok = false; // Устанавливаем флаг в false
                        }
                    }
                }
            }
        }

        if (ok) {
            results.push_back("YES"); // Сохраняем результат
        } else {
            results.push_back("NO"); // Сохраняем результат
        }
    }
}
```



```
    }  
}  
  
// Выводим все результаты после завершения всех проверок  
for (int i = 0; i < (int) (results).size(); i++) {  
    cout << results[i] << "\n";  
}  
}
```

Реализация на Python

```
# Считываем с клавиатуры количество строк и значение p  
u, p = map(int, input().split())  
  
# Создаём список для хранения результатов  
results = []  
  
# Обрабатываем u строк  
for _ in range(u):  
    s = input().strip() # Считываем текущую строку  
    ok = True # Флаг, указывающий на A-периодичность  
  
    # Проверяем A-периодичность строки  
    for i in range(p):  
        c = '#' # Инициализируем переменную для отслеживания символа  
        for j in range(i, len(s), p):  
            if s[j] != '?':  
                if c == '#':  
                    c = s[j] # Устанавливаем 1-ый ненулевой  
                elif c != s[j]:  
                    ok = False # Устанавливаем флаг в False  
                    break  
        if not ok:  
            break  
  
    # Сохраняем результат в список  
    if ok:  
        results.append("YES")  
    else:  
        results.append("NO")  
  
# Выводим все результаты после завершения обработки строк  
for result in results:  
    print(result)
```

Временную сложность алгоритма можно определить следующим образом:

Пусть длина строки n . Внутренний цикл выполняется для каждого i от 0 до $p-1$. В каждой итерации внутреннего цикла мы проходим элементы строки с шагом p . Таким образом, количество итераций внутреннего цикла составляет примерно n/p для каждого i . В итоге, временная сложность будет: $O(u \times (p + n/p))$, где u — количество строк для проверки. Так как n значительно больше p , то можно упростить до: $O(u \times n)$.

Основные потребности в памяти включают вектор для хранения результатов и саму строку.



Результат выполнения алгоритма для указанных исходных данных

стандартный ввод	стандартный вывод
2 6 xy?w?xyz?oxyz? mn?a?c?noabc?n	NO YES

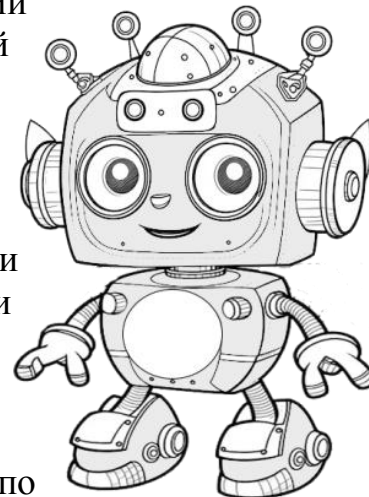
Для первого слова можем получить A-периодичную последовательность **mnoabctmnoabctmn**.



Задача 6 (25 баллов)

РобоМакс и тайна цветных деталей

На производственном предприятии «Механический Мастер» функционирует длинный конвейер, на котором детали пронумерованы от 1 до N (где $N \leq 10^5$). Эти детали движутся по кругу, таким образом, перед первой деталью оказывается деталь с номером N , а перед любой другой — деталь с номером, на единицу меньшим. Детали с четными номерами предназначены для зеленых машин и окрашены в яркий зеленый цвет, в то время как нечетные детали окрашены в насыщенный синий цвет и предназначены для синих машин.



На этом заводе трудится наш герой — робот по имени РобоМакс. РобоМакс — маленький, но невероятно умный робот с большими глазами и яркой светодиодной панелью. Его задача состоит в том, чтобы забирать детали для дальнейшего производства. Однако у него есть особое правило: он может забрать только те детали, у которых обе соседние детали (та, что перед ней, и та, что за ней) имеют одинаковый цвет.

РобоМакс внимательно осматривает конвейер. Если обе соседние детали имеют одинаковый цвет, он забирает центральную деталь, в то время как соседние продолжают двигаться по кругу, не меняя своих положений. Если же одна деталь зеленая, а другая синяя, РобоМакс не может забрать нужную деталь, и она остается на месте, вызывая у него легкую грусть.

Представьте себе, как быстро и эффективно работает РобоМакс, вызывая восхищение рабочих. Однако постепенно становится заметно, что некоторые детали остаются без внимания. Теперь возникает вопрос: сколько деталей останется на конвейере после нескольких успешных операций РобоМакса? Ваша задача заключается в том, чтобы помочь ему выяснить, сколько деталей в итоге останется на конвейере, учитывая правила его работы.

Формат входных данных

В первой строке два натуральных числа N и K ($3 \leq N \leq 50000$, $1 \leq K \leq N-2$). Во второй строке через пробел K чисел, каждое из интервала от 1 до N — номера деталей в том порядке, в котором их пытается взять РобоМакс. Если детали с указанным номером к моменту, когда за ней приходит РобоМакс, уже нет, ничего не происходит.

Формат выходных данных



Одно число — ответ на вопрос.

Примеры

стандартный ввод	стандартный вывод
13 11 1 2 1 13 2 6 12 3 13 5 3	8

Решите указанную задачу, используя один из языков программирования, и укажите, какой именно язык был выбран. Предложите эффективное решение, учитывая минимизацию времени выполнения программы и объема памяти, необходимого для её работы. Постарайтесь использовать алгоритмы и структуры данных, которые обеспечивают оптимальную производительность.

Реализуйте решение в виде программного кода, обязательно добавив комментарии, которые поясняют ключевые шаги алгоритма и переменные, если это важно для понимания проверяющему. Допускается использование только стандартных библиотек, встроенных в язык. Подробно опишите алгоритм решения, предоставив текстовое описание или блок-схему, объяснив логику работы программы.

Приведите результат выполнения программы для заданных исходных данных.

Исходные данные для выполнения задания

стандартный ввод	стандартный вывод
15 12 15 1 2 4 5 3 11 12 14 8 7 9	?

Ваше решение должно быть аккуратно оформлено, читабельно и соответствовать стандартам выбранного языка программирования.

Решение

Чтение входных данных

Сначала считываются значения n (количество деталей) и k (количество операций).

Инициализация массивов

Создаются два массива `pred` и `next`, которые представляют индексы предыдущей и следующей деталей соответственно. Массив `lab` инициализируется единицами, показывая, что изначально все детали находятся на конвейере.

Обработка операций

Считывается номер детали и проверяется, можно ли забрать деталь по



заданным правилам. Если обе соседние детали одного цвета и текущая деталь не была забрана, она забирается, и связи между соседними деталями обновляются.

Подсчет оставшихся деталей

В конце программы производится подсчет оставшихся на конвейере деталей, после чего результат выводится на экран.

Реализация на C++

```
#include<bits/stdc++.h>
#define int long
using namespace std;
int n, k, a; // Объявляем глобальные переменные: n (количество
деталей), k (количество запросов), a (номер детали запроса)

signed main(){
    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
    // Чтение входных данных: количество деталей и количество
запросов
    cin >> n >> k;
    // Массивы для предшествующих, последующих и меток деталей
    vector<int> pred(n), next(n), lab(n, 1);

    // Заполняем массив предшествующих, указывая,
    // что предыдущая деталь у первой – последняя
    for(int i = 0; i < n; i++) pred[i] = (i-1 + n) % n;
    // Заполняем массив последующих, указывая,
    // что следующая деталь у последней – первая
    for(int i = 0; i < n; i++) next[i] = (i+1) % n;
    // Обработка запросов РобоМакса
    for(int i = 0; i < k; i++){
        cin >> a; // Читаем номер детали
        a--;
        // Переход к нулевой индексации (т.к. вводятся номера от 1 до N)
        if(pred[a]%2 == next[a]%2 && lab[a] == 1){
            lab[a] = 0; // Помечаем деталь как убранную
            next[pred[a]] = next[a];
            // Обновляем следующую деталь для предшествующей
            pred[next[a]] = pred[a];
            // Обновляем предшествующую для следующей детали
        }
    }
    // Подсчет оставшихся деталей на конвейере
    int ans = 0;
    for(auto q : lab) ans += q;
    cout << ans; // Выводим результат
}
```

Реализация на Python

```
def main():
    n, k = map(int, input().split())
```




```
# Создаем массивы предшествующих и последующих деталей
pred = [(i - 1 + n) % n for i in range(n)]
next = [(i + 1) % n for i in range(n)]
# Массив, забрана ли деталь (1 - нет, 0 - да)
lab = [1] * n
for _ in range(k):
    # Индекс детали, которую РобоМакс собирается проверить
    a = int(input()) - 1
    # Проверяем, могут ли соседние детали быть одного цвета
    if pred[a] % 2 == next[a] % 2 and lab[a] == 1:
        lab[a] = 0 # Забираем деталь
        next[pred[a]] = next[a] # меняем соседние
        pred[next[a]] = pred[a]
# Подсчитываем количество оставшихся деталей
ans = sum(lab)
print(ans)

if __name__ == "__main__":
    main()
```

Давайте проанализируем временную и пространственную сложность решения. Создание массивов `pred`, `next` и `lab` требует $O(n)$ времени, поскольку мы проходимся по всем деталям. Цикл по запросам выполняется k раз. время выполнения всей программы будет суммироваться: $O(n)+O(k)=O(n+k)$.

Мы используем три массива: `pred`, `next` и `lab`, каждый из которых занимает $O(n)$ пространства. Следовательно, потребление памяти для этих массивов составляет $O(n)$.

Результат выполнения алгоритма для указанных исходных данных

стандартный ввод	стандартный вывод
15 12 15 1 2 4 5 3 11 12 14 8 7 9	9



ВАРИАНТ 4

Задача 1 (10 баллов)

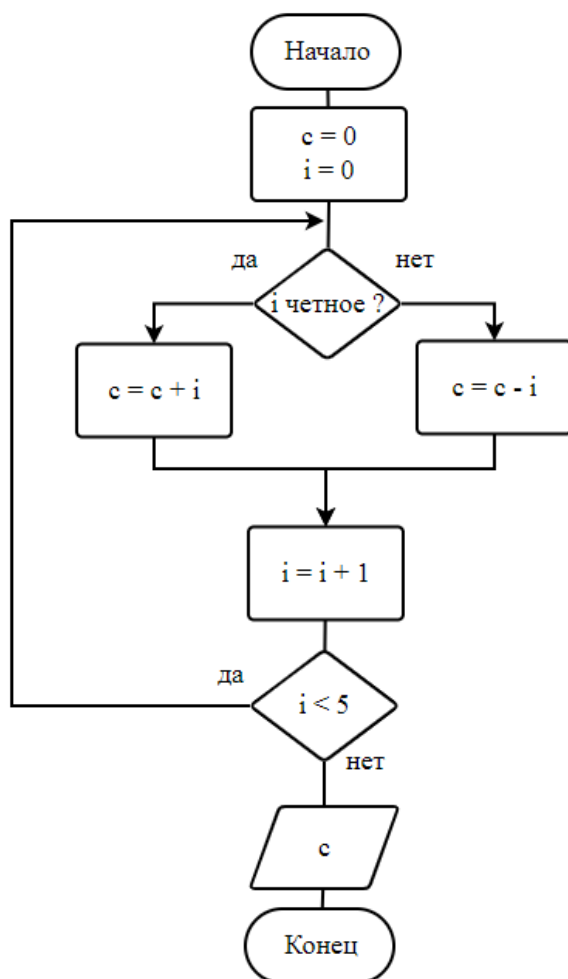
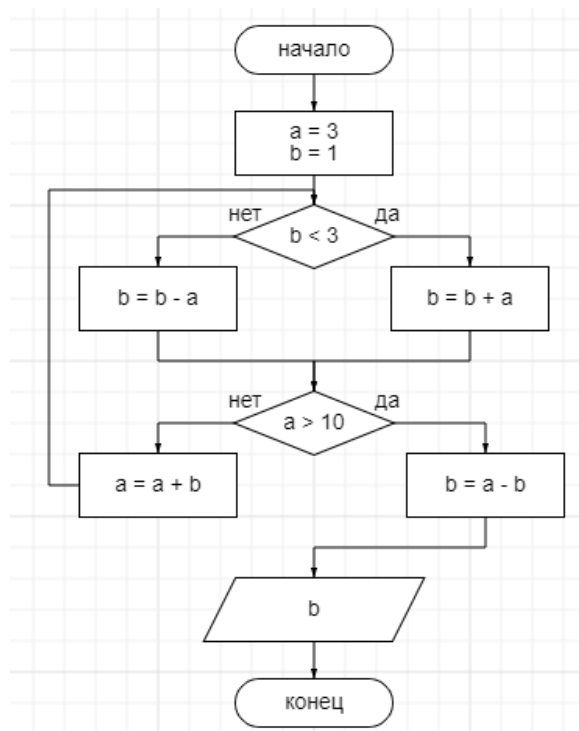
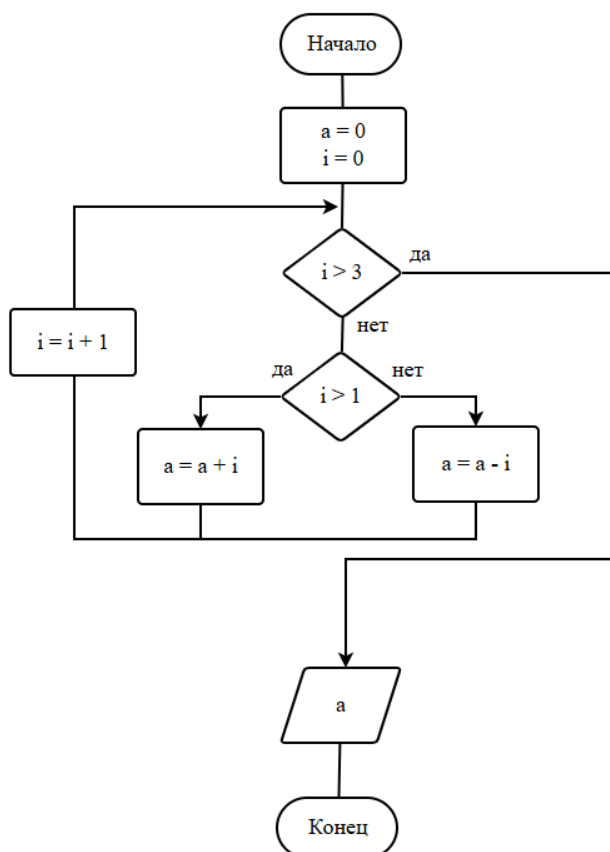
Разведчики

Нашими разведчиками на линии боевого соприкосновения были обнаружены средства противника, сведения о которых они передали в виде закодированной электронной таблицы на командный пункт. Определите количество этих технических средств, значение которых расположены в ячейках B2:F2. В ответе запишите значения ячеек в одну строчку, через запятую, начиная с B2 и заканчивая F2.

	A	B	C	D	E	F
		Танки	БМП	Орудия	Установки РЭБ	Опорные пункты
1						
2						
3						
4	a =			1) =СЧЁТЕСЛИ(B2:F2;B4)		
5	b =			2) =СЧЁТЕСЛИ(E2;B4)+B5		
6	c =			3) =СУММЕСЛИ(D2:E2;B4)-B6		
7	d =			4) =СУММ(D2:F2)-B6		
8				5) =ЕСЛИ(B7>5;B2-C2;СУММ(B2;C2))		
9				6) =ЕСЛИ(B7<=5;B2-C2;СУММ(B2;C2))		
10						

	A	B	C	D	E	F
		Танки	БМП	Орудия	Установки РЭБ	Опорные пункты
1						
2						
3						
4	a =			1)	2	
5	b =			2)	17	
6	c =			3)	6	
7	d =			4)	7	
8				5)	7	
9				6)	-7	
10						

Значения a, b и c определите, выполнив соответствующие алгоритмы по блок-схемам, d (ячейка B7) – неизвестное натуральное число.





Решение

Определяем значения коэффициентов a , b , c выполнив соответствующие алгоритмы по представленным блок-схемам. В результате получаем следующие значения.

	A	B	C
3			
4	a =	4	
5	b =	16	
6	c =	2	
7	d =		
8			

1) Значение формулы в ячейке E4 «=СЧЁТЕСЛИ(B2:F2;B4)» равно 2, следовательно в ячейках B2:F2 находятся две 4.

2) Значение формулы в ячейке E5 «=СЧЁТЕСЛИ(E2;B4)+B5» равно 17, откуда значение «=СЧЁТЕСЛИ(E2;B4)» равно $17 - B5 = 17 - 16 = 1$. Следовательно в ячейке E2 находится 4.

3) Значение формулы в ячейке E6 «=СУММЕСЛИ(D2:E2;B4)-B6» равно 6, откуда значение «=СУММЕСЛИ(D2:E2;B4)» равно $6 + B6 = 6 + 2 = 8$. Следовательно в ячейке D2, также как и в E2, находится значение 4.

4) Значение формулы в ячейке E7 «=СУММ(D2:F2)-B6» равно 7, откуда значение «=СУММ(D2:F2)» равно $7 + B6 = 7 + 2 = 9$. Значения ячеек D2 и E2 известны, находим значение ячейки F2. $F2 = 9 - 4 - 4 = 1$.

5) Анализируя формулы в ячейке E8 «=ЕСЛИ(B7>5;B2-C2;СУММ(B2;C2))» и ячейке E9 «=ЕСЛИ(B7<=5;B2-C2;СУММ(B2;C2))» понимаем, что если значение в ячейке B7 будет больше 5, то в ячейке E8 будет $B2 - C2$, а в ячейке E9: $B2 + C2$, в противном случае в ячейке E8 будет $B2 + C2$, а в ячейке E9: $B2 - C2$. Так как, по условию, значения в ячейках B2:F2 натуральные числа (количество объектов), то разность всегда меньше суммы, получаем:

$$\begin{cases} B2 + C2 = 7 \\ B2 - C2 = -7 \end{cases} \Rightarrow \begin{cases} C2 - 7 + C2 = 7 \\ B2 = C2 - 7 \end{cases} \Rightarrow \begin{cases} C2 = 7 \\ B2 = 0 \end{cases}$$

Ответ: 0, 7, 4, 4, 1.



Задача 2 (10 баллов)

Чертёжник

Исполнитель Чертёжник перемещается на координатной плоскости, оставляя след в виде линии. Чертёжник может выполнять следующие команды:

1) Сместиться на (a, b) , где a, b – целые числа. Эта команда перемещает исполнитель из точки с координатами (x, y) в точку с координатами $(x + a; y + b)$. Например, если Чертёжник находится в точке с координатами $(1, 2)$, то команда сместиться на $(2, -1)$ переместит Чертёжника в точку $(3, 1)$.

2) Цикл:

ПОКА <условие>

последовательность команд

КОНЕЦ ПОКА

Этот цикл означает, что последовательность команд будет выполняться до тех пор, пока условие <условие> выполняется.

Исполнителю Чертёжник был дан для исполнения следующий алгоритм (буквами a, b обозначены неизвестные числа):

НАЧАЛО

сместиться на $(2, 3)$

ПОКА $x + y$ не равно 10

сместиться на (a, b)

сместиться на $(6, 3)$

КОНЕЦ ПОКА

сместиться на $(5, -4)$

КОНЕЦ

После выполнения программы смещение Чертёжника по оси y будет в два раза больше смещения по оси x . Сколько раз выполнялась последовательность команд цикла ПОКА, если известно, что команды цикла ПОКА были выполнены более одного раза и сумма значений координат по осям x и y до выполнения программы была равна 1

Решение

Значение координат Чертёжника в результате выполнения цикла ПОКА:

$$x_n = x_0 + \Delta x = x_0 + 2 + N(a + 6)$$

$$y_n = y_0 + \Delta y = y_0 + 3 + N(b + 3)$$

В этом уравнении N – количество повторений команд цикла ПОКА, которое является искомым значением.

Из условия завершения цикла $x_n + y_n = 10$, получим:

$$x_0 + 2 + N(a + 6) + y_0 + 3 + N(b + 3) = 10$$

Из условия $x_0 + y_0 = 1$, получим:



$$N(a + b + 9) = 4$$

Общее изменение координат Чертёжника в результате выполнения этого алгоритма:

$$x - x_0 = 2 + N(a + 6) + 5$$

$$y - y_0 = 3 + N(b + 3) - 4$$

Так как изменение координат по оси y будет превышать изменение координат по оси x в три раза, получим:

$$3(2 + N(a + 6) + 5) = 3 + N(b + 3) - 4$$

$$N(3a + 18) - 21 = N(b + 3) - 1$$

$$N(3a - b + 15) = 20$$

Получим систему уравнений:

$$\begin{cases} N(a + b + 9) = 4 \\ N(3a - b + 15) = 20 \end{cases}$$

Нужно найти натуральное N , при котором эта система уравнений разрешима в целых числах относительно a и b . Для этого число N должно быть одновременно делителем чисел 4 и 20. Общими делителями этих чисел являются числа 1, 2 и 4, при этом 1 не подходит по условию, что цикл повторялся более 1 раза, а число 4 не подходит из условия, что a, b – целые числа, при подстановке в систему уравнений выше.

Ответ: последовательность команд цикла ПОКА выполнилась 2 раза.

**Задача 3 (10 баллов)****Шкатулка**

Николай купил новую цифровую шкатулку с четырехзначным кодом. Он придумал код. Чтобы лучше его запомнить, он решил преобразовать каждую цифру этого 4-значного числа в двоичный код. Далее он составил логическое выражение. В результирующем столбце таблицы истинности к этому выражению отобразилась та самая двоичная последовательность для всего кода.

$$(a \rightarrow b \equiv \neg c) \vee (c \wedge b \wedge \neg a)$$

Строки в таблице истинности идут последовательно, 000, 001, 010 и т.д.

Какой шифр на шкатулке у Николая, если известно, что первая цифра кода - 5?

Решение

Составим таблицу истинности

a	b	c	$a \rightarrow b$	$\neg c$	$a \rightarrow b \equiv \neg c$	$c \wedge b$	$\neg a$	$c \wedge b \wedge \neg a$	$a \rightarrow b \equiv \neg c \vee c \wedge b \wedge \neg a$
0	0	0	1	1	1	0	1	0	1
0	0	1	1	0	0	0	1	0	0
0	1	0	1	1	1	0	1	0	1
0	1	1	1	0	0	1	1	1	1
1	0	0	0	1	0	0	0	0	0
1	0	1	0	0	1	0	0	0	1
1	1	0	1	1	1	0	0	0	1
1	1	1	1	0	0	1	0	0	0

Двоичный код в последнем столбце можно разбить на двоичные числа следующим образом: $101_2 10_2 1_2 10_2$

Таким образом получаем следующий десятичный код: 5 2 1 2

Ответ: 5 2 1 2

**Задача 4 (20 баллов)****Добыча урана**

Шахта «Уран-2» использует систему шифрования для передачи сообщения о добыче урана в головную организацию. Для дешифровки данного сообщения каждый элемент закрытого ключа, хранящегося в головной организации, умножается на n^{-1} – мультипликативное обратное к значению n по модулю m (остаток от деления $n \cdot n^{-1}$ на m равен 1). Результат вычисляется как остаток от деления полученного числа на m . Полученный модифицированный закрытый ключ используется для дешифрования, а 8-битный ключ определяет, какие элементы модифицированного ключа суммируются для получения расшифрованного сообщения.

В понедельник добыча урана на шахте составила $4A_{16}$ килограмм. Ежедневно добываемая масса урана увеличивалась согласно арифметической прогрессии, состоящей из последовательности нечетных чисел, начиная со второго члена (например, количество добытого урана во второй день равно $4A_{16} + 3_{16}$ кг и т.д.). С шахты в головную организацию в течение рабочей недели поступило пять сообщений с количеством добытого урана, которые преобразовывались в головной организации в 8-битный двоичный код и служили ключом для дешифрования. Закрытый ключ, хранящийся в организации, представляет собой следующую последовательность $\{2, 3, 6, 12, 19, 34, 64, 110\}$, значения n^{-1} и m равны 97 и 252 соответственно.

Расшифруйте полученные сообщения, зная, что они представляют номер склада, где хранится добытый уран в каждый из дней рабочей недели. В ответе укажите последовательность пяти чисел через запятую, начиная с номера склада, где хранился добытый уран в понедельник, заканчивая номером склада, где хранился добытый уран в пятницу.

Решение

На первом этапе необходимо определить сообщения, которые были переданы в административное управление со вторника по пятницу. В условии сказано, что ежедневно добываемая масса урана (шифротекст) увеличивалась согласно арифметической прогрессии, состоящей из последовательности нечетных чисел, начиная со второго члена, т.е. во второй день количество добытого урана увеличилось на 3_{16} кг, в третий день – на 5_{16} кг, в четвертый день – на 7_{16} кг и в пятый день – на 9_{16} кг. Для нахождения переданных сообщений произведем сложение чисел в шестнадцатеричной системе.

Сложение чисел в шестнадцатеричной системе счисления выполняется поразрядно, начиная с младшего разряда. Если при сложении двух чисел одинакового разряда получается двузначное число, то значение его старшего разряда (единицу) добавляют в старший разряд. Следовательно:



Второе сообщение: $3E_{16} + 3_{16} \Rightarrow E + 3 = 11 \Rightarrow 3 + 1 = 4 \Rightarrow 41_{16}$

Третье сообщение: $41_{16} + 5_{16} \Rightarrow 1 + 5 = 6 \Rightarrow 46_{16}$

Четвертое сообщение: $46_{16} + 7_{16} \Rightarrow 6 + 7 = D \Rightarrow 4D_{16}$

Пятое сообщение: $4D_{16} + 9_{16} \Rightarrow D + 9 = 16 \Rightarrow 5 + 1 = 6 \Rightarrow 56_{16}$

Таким образом, последовательность переданных сообщений в течение рабочей недели: $3E_{16}, 41_{16}, 46_{16}, 4D_{16}, 56_{16}$

Далее необходимо найти ключ для дешифрования каждого сообщения. По условию задачи он представляет собой преобразованное значение переданного сообщения в 8-битном двоичном виде. Для перевода числа из шестнадцатеричной системы счисления в двоичную можно воспользоваться несколькими способами, мы воспользуемся переводом сначала в десятичную систему счисления, затем в двоичную.

Каждую цифру шестнадцатеричного числа умножим на 16 в степени, равной позиции этой цифры, и затем сложим все полученные произведения. Далее разделим полученное число в десятичной системе на 2 и запишем остаток до тех пор, пока неполное частное не будет равно нулю. Полученные остатки записываем в обратном порядке.

Число в 16-ой системе счисления	Перевод в 10-ую систему счисления	Число в 10-ой системе счисления	Перевод в 2-ую систему счисления	Число в 2-ой системе счисления
$3E_{16}$	$3 \times 16^1 + E \times 16^0$ $= 3 \times 16^1 + 14$ $\times 16^0 = 48 + 14$ $= 62$	62_{10}	1) $\frac{62}{2} = 31$ (остаток 0) 2) $\frac{31}{2} = 15$ (остаток 1) 3) $\frac{15}{2} = 7$ (остаток 1) 4) $\frac{7}{2} = 3$ (остаток 1) 5) $\frac{3}{2} = 1$ (остаток 1) 6) $\frac{1}{2} = 0$ (остаток 1)	111110_2
41_{16}	$4 \times 16^1 + 1 \times 16^0$ $= 4 \times 16^1 + 1$ $\times 16^0 = 64 + 1$ $= 65$	65_{10}	1) $\frac{65}{2} = 32$ (остаток 1) 2) $\frac{32}{2} = 16$ (остаток 0) 3) $\frac{16}{2} = 8$ (остаток 0) 4) $\frac{8}{2} = 4$ (остаток 0) 5) $\frac{4}{2} = 2$ (остаток 0) 6) $\frac{2}{2} = 1$ (остаток 0) 7) $\frac{1}{2} = 0$ (остаток 1)	1000001_2



46_{16}	$4 \times 16^1 + 6 \times 16^0$ $= 64 + 6 = 70$	70_{10}	$\frac{70}{2} = 35$ (остаток 0) $\frac{35}{2} = 17$ (остаток 1) $\frac{17}{2} = 8$ (остаток 1) $\frac{8}{2} = 4$ (остаток 0) $\frac{4}{2} = 2$ (остаток 0) $\frac{2}{2} = 1$ (остаток 0) $\frac{1}{2} = 0$ (остаток 1)	1000110_2
$4D_{16}$	$4 \times 16^1 + D \times 16^0$ $= 64 + 13 = 77$	77_{10}	$\frac{77}{2} = 38$ (остаток 1) $\frac{38}{2} = 19$ (остаток 0) $\frac{19}{2} = 9$ (остаток 1) $\frac{9}{2} = 4$ (остаток 1) $\frac{4}{2} = 2$ (остаток 0) $\frac{2}{2} = 1$ (остаток 0) $\frac{1}{2} = 0$ (остаток 1)	1001101_2
56_{16}	$5 \times 16^1 + 6 \times 16^0$ $= 80 + 6 = 86$	86_{10}	$\frac{86}{2} = 43$ (остаток 0) $\frac{43}{2} = 21$ (остаток 1) $\frac{21}{2} = 10$ (остаток 1) $\frac{10}{2} = 5$ (остаток 0) $\frac{5}{2} = 2$ (остаток 1) $\frac{2}{2} = 1$ (остаток 0) $\frac{1}{2} = 0$ (остаток 1)	1010110_2

Мы нашли 8-битный двоичный код для каждого сообщения, который служит ключом для дешифрирования. Далее исходя из условия задачи для дешифровки сообщений требуется модифицированный закрытый ключ, который рассчитывается как каждый элемент закрытого ключа умноженный на n^{-1} и найденный остаток от деления полученного числа на m . В нашем случае закрытый ключ представляет последовательность $\{2, 3, 6, 12, 19, 34, 64, 110\}$, значения n^{-1} и m равны 97 и 252 соответственно.

Модифицированный закрытый ключ:

1) (2×97) делится на 252 с остатком 194



- 2) (2×97) делится на 252 с остатком 39
- 3) (6×97) делится на 252 с остатком 78
- 4) (12×97) делится на 252 с остатком 156
- 5) (19×97) делится на 252 с остатком 79
- 6) (34×97) делится на 252 с остатком 22
- 7) (64×97) делится на 252 с остатком 160
- 8) (110×97) делится на 252 с остатком 86

Модифицированный закрытый ключ {194, 39, 78, 156, 79, 22, 160, 86}.

Найденный ранее 8-битный ключ определяет, какие элементы модифицированного ключа суммируются для получения расшифрованного сообщения. При нехватке элементов 8-битного ключа при выборе элементов модифицированного ключа, недостающие элементы приравниваются к 0). Таким образом, произведем дешифровку принятых в административном управлении сообщений:

Первое сообщение:

$$\begin{array}{cccccccc} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 194 & 39 & 78 & 156 & 79 & 22 & 160 & 86 \end{array} = 78 + 156 + 79 + 22 + 160 = 495$$

Второе сообщение:

$$\begin{array}{cccccccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 194 & 39 & 78 & 156 & 79 & 22 & 160 & 86 \end{array} = 39 + 86 = 125$$

Третье сообщение:

$$\begin{array}{cccccccc} 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 194 & 39 & 78 & 156 & 79 & 22 & 160 & 86 \end{array} = 39 + 22 + 160 = 221$$

Четвертое сообщение:

$$\begin{array}{cccccccc} 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 194 & 39 & 78 & 156 & 79 & 22 & 160 & 86 \end{array} = 39 + 79 + 22 + 86 = 226$$

Пятое сообщение:

$$\begin{array}{cccccccc} 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 194 & 39 & 78 & 156 & 79 & 22 & 160 & 86 \end{array} = 39 + 156 + 22 + 160 = 377$$

В ответе необходимо указать последовательность пяти чисел через запятую, начиная с номера склада, где хранился добытый уран в понедельник, заканчивая номером склада, где хранился добытый уран в пятницу. Следовательно, ответ задачи выглядит следующим образом 495, 125, 221, 226, 377.

Ответ: 495, 125, 221, 226, 377



Задача 5 (25 баллов)

Задача об удивительных последовательностях

Профессор Тимофей Чудоген, известный биолог, проводит интересное исследование, в котором изучает особый искусственный генетический код. Этот код состоит из последовательности искусственных нуклеотидов, которые обозначаются строчными латинскими буквами. Однако в этой последовательности есть вопросы — знаки, которые нужно заменить на определенные буквы, чтобы завершить анализ.



Профессор считает, что этот код имеет удивительную особенность, которую он называет А-периодичность. Периодичность означает, что если взять два искусственных нуклеотида, расположенных на расстоянии, кратном заданному числу А, то они должны быть одинаковыми. Это значит, что все нуклеотиды, стоящие на тех же позициях, кратных А, должны совпадать.

Профессор Чудоген уверен, что понимание этих особенностей поможет в будущем в таких науках, как медицина, экология и биология. Его задача заключается в том, чтобы найти правильные буквы для замены знаков вопроса и определить можно ли соответствующую строку сделать А-периодичной путем замены всех знаков вопроса. Помогите профессору.

Формат входных данных

В первой строке содержится два числа через пробел: N — общее количество строк, которые вам нужно проверить на А-периодичность и число А ($1 \leq A \leq 5 \cdot 10^3$). Далее содержится N непустых строк, состоящих из малых букв латиницы и знаков вопроса $1 \leq N \leq 5 \cdot 10^4$, суммарная длина всех строк не превосходит $2 \cdot 10^5$.

Формат выходных данных

Вывести N строк, в каждой из которых вывести либо "YES", если соответствующую строку можно сделать А-периодичной путем замены всех знаков вопроса на некоторые буквы и "NO" в противном случае. Будем считать последовательность А-периодичной, если для любых двух символов, расстояние между которыми кратно А верно, что они совпадают.



Примеры

стандартный ввод	стандартный вывод
8 1	NO
abacabaca	NO
abracadabra	YES
aa?aaa?aaaaaaaaaaa	YES
q	YES
???????	NO
q?er?w?	NO
q?erw??	NO
q?er?wer?werw?er	

Решите указанную задачу, используя один из языков программирования, и укажите, какой именно язык был выбран. Предложите эффективное решение, учитывая минимизацию времени выполнения программы и объема памяти, необходимого для её работы. Постарайтесь использовать алгоритмы и структуры данных, которые обеспечивают оптимальную производительность.

Реализуйте решение в виде программного кода, обязательно добавив комментарии, которые поясняют ключевые шаги алгоритма и переменные, если это важно для понимания проверяющему. Допускается использование только стандартных библиотек, встроенных в язык. Подробно опишите алгоритм решения, предоставив текстовое описание или блок-схему, объяснив логику работы программы.

Приведите результат выполнения программы для заданных исходных данных. Результат выполнения для указанных исходных данных должен включать вывод программы, а также, если соответствующую строку можно сделать A-периодичной заменой всех знаков вопроса, привести полученную строку.

Исходные данные для выполнения задания

стандартный ввод	стандартный вывод
2 5 q?er?wer?wer?w xy?w?xyz?oxyz?	?

Ваше решение должно быть аккуратно оформлено, читабельно и соответствовать стандартам выбранного языка программирования.

Решение

Чтение входных данных

Программа сначала считывает количество строк и значение, которое определяет период, по которому будет происходить проверка.

Инициализация

Для каждой строки создается флаг ok, который по умолчанию устанавливается в true. Этот флаг отслеживает, соответствует ли текущая



строка необходимым критериям А-периодичности.

Проверка А-периодичности

Для каждого индекса от 0 до $p-1$ (внешний цикл) инициализируется символ s , устанавливаемый в '#'. Он будет использоваться для отслеживания первого ненулевого символа в строке.

Внутренний цикл проходит по строке с шагом p , начиная с текущего индекса. Если текущий символ не равен '?', проверяется:

Если s все еще '#', это значит, что найден первый ненулевой символ, и s устанавливается на этот символ.

Если s уже установлен, проверяется, совпадает ли текущий символ с s . Если нет — флаг ok устанавливается в `false`, и цикл прерывается.

Сохранение результата

После завершения проверки для каждой строки результат ("YES" или "NO") добавляется в список `results`.

Вывод результата

В конце программа выводит результаты всех тестов.

Реализация на C++

```
#include <bits/stdc++.h>
#define int long

using namespace std;

signed main() {
    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);

    int u, p;
    cin >> u >> p;
    vector<string> results; // Вектор для хранения результатов

    while (u--) {
        string s;
        cin >> s;
        bool ok = true; // Флаг, отслеживает правильность строк

        for (int i = 0; i < p; i++) {
            char c = '#'; // Устанавливаем символ по умолчанию
            for (int j = 0; i + j < (int)(s).size(); j += p) {
                if (s[i + j] != '?') {
                    if (c == '#') {
                        c = s[i + j]; // Устанавливаем 1-ый ненулевой
                    } else {
                        if (c != s[i + j]) {
                            ok = false; // Устанавливаем флаг в false
                        }
                    }
                }
            }
        }

        if (ok) {
            results.push_back("YES"); // Сохраняем результат
        } else {
            results.push_back("NO"); // Сохраняем результат
        }
    }
}
```



```
    }  
}  
  
// Выводим все результаты после завершения всех проверок  
for (int i = 0; i < (int)(results).size(); i++) {  
    cout << results[i] << "\n";  
}  
}
```

Реализация на Python

```
# Считываем с клавиатуры количество строк и значение p  
u, p = map(int, input().split())  
  
# Создаём список для хранения результатов  
results = []  
  
# Обрабатываем u строк  
for _ in range(u):  
    s = input().strip() # Считываем текущую строку  
    ok = True # Флаг, указывающий на A-периодичность  
  
    # Проверяем A-периодичность строки  
    for i in range(p):  
        c = '#' # Инициализируем переменную для отслеживания символа  
        for j in range(i, len(s), p):  
            if s[j] != '?':  
                if c == '#':  
                    c = s[j] # Устанавливаем 1-ый ненулевой  
                elif c != s[j]:  
                    ok = False # Устанавливаем флаг в False  
                    break  
        if not ok:  
            break  
  
    # Сохраняем результат в список  
    if ok:  
        results.append("YES")  
    else:  
        results.append("NO")  
  
# Выводим все результаты после завершения обработки строк  
for result in results:  
    print(result)
```

Временную сложность алгоритма можно определить следующим образом:

Пусть длина строки n . Внутренний цикл выполняется для каждого i от 0 до $p-1$. В каждой итерации внутреннего цикла мы проходим элементы строки с шагом p . Таким образом, количество итераций внутреннего цикла составляет примерно n/p для каждого i . В итоге, временная сложность будет: $O(u \times (p + n/p))$, где u — количество строк для проверки. Так как n значительно больше p , то можно упростить до: $O(u \times n)$.

Основные потребности в памяти включают вектор для хранения результатов и саму строку.



Результат выполнения алгоритма для указанных исходных данных

стандартный ввод	стандартный вывод
2 5 q?er?wer?wer?w xy?w?xyz?oxyz?	NO YES

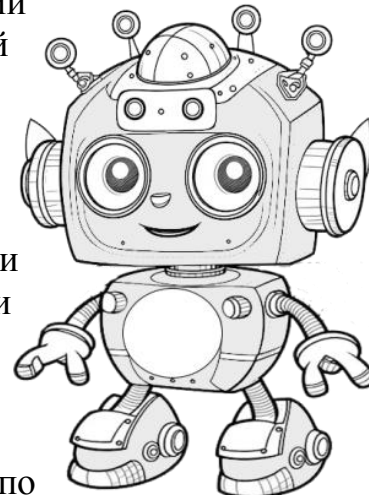
Для первого слова можем получить A-периодичную последовательность **xuzwoxuzwoxuzw**.



Задача 6 (25 баллов)

РобоМакс и тайна цветных деталей

На производственном предприятии «Механический Мастер» функционирует длинный конвейер, на котором детали пронумерованы от 1 до N (где $N \leq 10^5$). Эти детали движутся по кругу, таким образом, перед первой деталью оказывается деталь с номером N , а перед любой другой — деталь с номером, на единицу меньшим. Детали с четными номерами предназначены для зеленых машин и окрашены в яркий зеленый цвет, в то время как нечетные детали окрашены в насыщенный синий цвет и предназначены для синих машин.



На этом заводе трудится наш герой — робот по имени РобоМакс. РобоМакс — маленький, но невероятно умный робот с большими глазами и яркой светодиодной панелью. Его задача состоит в том, чтобы забирать детали для дальнейшего производства. Однако у него есть особое правило: он может забрать только те детали, у которых обе соседние детали (та, что перед ней, и та, что за ней) имеют одинаковый цвет.

РобоМакс внимательно осматривает конвейер. Если обе соседние детали имеют одинаковый цвет, он забирает центральную деталь, в то время как соседние продолжают двигаться по кругу, не меняя своих положений. Если же одна деталь зеленая, а другая синяя, РобоМакс не может забрать нужную деталь, и она остается на месте, вызывая у него легкую грусть.

Представьте себе, как быстро и эффективно работает РобоМакс, вызывая восхищение рабочих. Однако постепенно становится заметно, что некоторые детали остаются без внимания. Теперь возникает вопрос: сколько деталей останется на конвейере после нескольких успешных операций РобоМакса? Ваша задача заключается в том, чтобы помочь ему выяснить, сколько деталей в итоге останется на конвейере, учитывая правила его работы.

Формат входных данных

В первой строке два натуральных числа N и K ($3 \leq N \leq 50000$, $1 \leq K \leq N-2$). Во второй строке через пробел K чисел, каждое из интервала от 1 до N — номера деталей в том порядке, в котором их пытается взять РобоМакс. Если детали с указанным номером к моменту, когда за ней приходит РобоМакс, уже нет, ничего не происходит.

Формат выходных данных



Одно число — ответ на вопрос.

Примеры

стандартный ввод	стандартный вывод
13 11 1 2 1 13 2 6 12 3 13 5 3	8

Решите указанную задачу, используя один из языков программирования, и укажите, какой именно язык был выбран. Предложите эффективное решение, учитывая минимизацию времени выполнения программы и объема памяти, необходимого для её работы. Постарайтесь использовать алгоритмы и структуры данных, которые обеспечивают оптимальную производительность.

Реализуйте решение в виде программного кода, обязательно добавив комментарии, которые поясняют ключевые шаги алгоритма и переменные, если это важно для понимания проверяющему. Допускается использование только стандартных библиотек, встроенных в язык. Подробно опишите алгоритм решения, предоставив текстовое описание или блок-схему, объяснив логику работы программы.

Приведите результат выполнения программы для заданных исходных данных.

Исходные данные для выполнения задания

стандартный ввод	стандартный вывод
12 12 1 2 7 11 9 12 4 7 9 10 4 5	?

Ваше решение должно быть аккуратно оформлено, читабельно и соответствовать стандартам выбранного языка программирования.

Решение

Чтение входных данных

Сначала считываются значения n (количество деталей) и k (количество операций).

Инициализация массивов

Создаются два массива `pred` и `next`, которые представляют индексы предыдущей и следующей деталей соответственно. Массив `lab` инициализируется единицами, показывая, что изначально все детали находятся на конвейере.

Обработка операций

Считывается номер детали и проверяется, можно ли забрать деталь по



заданным правилам. Если обе соседние детали одного цвета и текущая деталь не была забрана, она забирается, и связи между соседними деталями обновляются.

Подсчет оставшихся деталей

В конце программы производится подсчет оставшихся на конвейере деталей, после чего результат выводится на экран.

Реализация на C++

```
#include<bits/stdc++.h>
#define int long
using namespace std;
int n, k, a; // Объявляем глобальные переменные: n (количество
деталей), k (количество запросов), a (номер детали запроса)

signed main(){
    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
    // Чтение входных данных: количество деталей и количество
запросов
    cin >> n >> k;
    // Массивы для предшествующих, последующих и меток деталей
    vector<int> pred(n), next(n), lab(n, 1);

    // Заполняем массив предшествующих, указывая,
    // что предыдущая деталь у первой – последняя
    for(int i = 0; i < n; i++) pred[i] = (i-1 + n) % n;
    // Заполняем массив последующих, указывая,
    // что следующая деталь у последней – первая
    for(int i = 0; i < n; i++) next[i] = (i+1) % n;
    // Обработка запросов РобоМакса
    for(int i = 0; i < k; i++){
        cin >> a; // Читаем номер детали
        a--;
        // Переход к нулевой индексации (т.к. вводятся номера от 1 до N)
        if(pred[a]%2 == next[a]%2 && lab[a] == 1){
            lab[a] = 0; // Помечаем деталь как убранную
            next[pred[a]] = next[a];
            // Обновляем следующую деталь для предшествующей
            pred[next[a]] = pred[a];
            // Обновляем предшествующую для следующей детали
        }
    }
    // Подсчет оставшихся деталей на конвейере
    int ans = 0;
    for(auto q : lab) ans += q;
    cout << ans; // Выводим результат
}
```

Реализация на Python

```
def main():
    n, k = map(int, input().split())
```



```
# Создаем массивы предшествующих и последующих деталей
pred = [(i - 1 + n) % n for i in range(n)]
next = [(i + 1) % n for i in range(n)]
# Массив, забрана ли деталь (1 - нет, 0 - да)
lab = [1] * n
for _ in range(k):
    # Индекс детали, которую РобоМакс собирается проверить
    a = int(input()) - 1
    # Проверяем, могут ли соседние детали быть одного цвета
    if pred[a] % 2 == next[a] % 2 and lab[a] == 1:
        lab[a] = 0 # Забираем деталь
        next[pred[a]] = next[a] # меняем соседние
        pred[next[a]] = pred[a]
# Подсчитываем количество оставшихся деталей
ans = sum(lab)
print(ans)

if __name__ == "__main__":
    main()
```

Давайте проанализируем временную и пространственную сложность решения. Создание массивов `pred`, `next` и `lab` требует $O(n)$ времени, поскольку мы проходимся по всем деталям. Цикл по запросам выполняется k раз. время выполнения всей программы будет суммироваться: $O(n)+O(k)=O(n+k)$.

Мы используем три массива: `pred`, `next` и `lab`, каждый из которых занимает $O(n)$ пространства. Следовательно, потребление памяти для этих массивов составляет $O(n)$.

Результат выполнения алгоритма для указанных исходных данных

стандартный ввод	стандартный вывод
12 12 1 2 7 11 9 12 4 7 9 10 4 5	5