Информатика 9 - 11 класс
Всероссийская олимпиада школьников 2024/25
Муниципальный этап
Чукотский автономный округ

Максимальная продолжительность – 210 мин.

Максимально возможное количество баллов – 500

К данному файлу также прилагается папка с тестами для жюри.

Задача 1 (100 баллов)

Руководитель крупного коммерческого предприятия Петров С.С. хочет позвонить партнеру по бизнесу, чтобы решить важный вопрос. Петров находится в городе, который расположен в часовом поясе UTC+A, а бизнес-партнер - в другом городе, в часовом поясе UTC+B. Часы Петрова С.С. показывают ровно N часов, помогите бизнесмену определить время в часовом поясе партнера в этот момент.

Входные данные: три целых числа – N , A и B , $0 \leq N \leq 23$, $-11 \leq A \leq 12$, $-11 \leq B \leq 12$.

В часовом поясе UTC+A местное время больше, чем время в часовом поясе UTC+0 на A часов (если же $A < 0$, то меньше на $|A|$ часов). Например, если в часовом поясе UTC+0 сейчас 12 часов, то в часовом поясе UTC+1 – 13 часов, а в часовом поясе UTC-1 – 11 часов.

Выходные данные: одно число – время (количество часов, которое может принимать значения от 0 до 23) в часовом поясе бизнес партнера.

При решении задачи обратите внимание, что в часовом поясе бизнес-партнера может быть уже следующая дата или предыдущая дата, программа должна вывести количество часов на часах бизнес-партнера в этот момент, то есть число от 0 до 23.

Пример работы программы

Ввод	Вывод	Примечание
13 1 -3	9	У Петрова С.С. – 13 часов, он живёт в часовом поясе UTC+1. В часовом поясе UTC+0 сейчас 12 часов. Бизнес-партнер живёт в часовом поясе UTC-3, и у него сейчас 9 часов.

Решение

Если в часовом поясе UTC+A время N часов, то в поясе UTC+0 время $N - A$, а в часовом поясе UTC+B время будет равно $N - A + B$. Но при этом не учитываются возможные переходы между сутками — время может оказаться в следующих сутках или в предыдущих сутках. Это можно учесть при помощи двух условий: если получили время, больше или равное 24 часа, то нужно вычесть 24 часа, а если получили отрицательное время, то нужно прибавить 24 часа. Получается следующее решение:

```
N = int(input())
A = int(input())
B = int(input())
answer = N - A + B
if answer >= 24:
    answer -= 24
if answer < 0:
    answer += 24
print(answer)
```

Но вместо условий проще взять остаток от деления на 24:

```

N = int(input())
A = int(input())
B = int(input())
answer = (N - A + B) % 24
print(answer)

```

Заметим, что в большинстве языков программирования (Pascal, C++, Java, C#) при взятии остатка от деления отрицательного числа на положительное получится отрицательный результат, то есть решение задачи будет неверным. В этом случае нужно действовать так (пример для Pascal).

```
answer := (N - A + B + 24) mod 24__
```

За программу без ошибок начисляется 100 баллов, программа содержит несущественные ошибки, не влияющие на правильность решения – 50 баллов, в ином случае – 0 баллов.

Задача 2 (Время – 1 сек., память – 16 Мб, 100 баллов)

В магазин «Детский мир» привезли партию игрушек, в ней оказались плюшевые собачки и кошечки разной высоты. Продавец расположил на витрине новые игрушки в ряд. Сначала поставил собачек по невозрастанию высоты, потом кошечек, тоже по невозрастанию высоты. Так, рядом с самой маленькой собачкой на витрине стоит самая высокая кошечка. Продавец хочет знать какая максимальная разница в высоте двух стоящих рядом игрушек. Напишите программу, которая поможет ответить на этот вопрос.

Входные данные: первая строка содержит целое число M – количество игрушек в партии ($2 \leq M \leq 50$). Следующие M строк содержат по два целых числа каждая: A_i и B_i – вид игрушки и высота в сантиметрах i -ой игрушки ($0 \leq A_i \leq 1$, $10 \leq B_i \leq 120$). Значение $A_i = 0$ означает, что i -ая игрушка – собачка, а значение $A_i = 1$ означает, что i -ая игрушка – кошечка.

Выходные данные: одно число – максимальное различие в высоте стоящих рядом на витрине игрушек.

Пример работы программы

Ввод	Вывод	Примечание
6 0 50 1 10 1 60 1 20 0 120 0 100	50	Максимальное различие в высоте у стоящих рядом собачек 100 см и 50 см.

Решение

Приведем наилучшее решение для Python:

```

import random

m = 10
list_toys = [[random.randint(0,1), random.randint(10,120)] for _ in range(m)]
list_toys = sorted(list_toys, key=lambda val: (val[0], -val[1]))
print(list_toys)
print(max([abs(list_toys[i][1] - list_toys[i+1][1]) for i in range(len(list_toys) - 1)]))

```

Здесь решение реализовано с помощью вложенных списков, для сортировки списков используется лямбда – функции, затем сравниваются соседние элементы списка.

Возможно формирование элементов списка вручную. Любое решение, удовлетворяющее целевым показателям времени и памяти, дающее верный результат, считается правильным.

За программу без ошибок начисляется 100 баллов, программа содержит несущественные ошибки, не влияющие на правильность решения – 50 баллов, в ином случае – 0 баллов.

Задача 3 (Время – 0,4 сек., память – 512 Мб, 100 баллов)

В город приехала популярная рок-группа. За билетами в концертный зал выстроилась большая очередь. Витя Осипов стоит в очереди под номером N и хочет знать заранее, какой номер ряда и места ему достанется. Помогите Вите ответить на этот вопрос, если известно, что касса продает билеты подряд: сначала в 1-й ряд на места с 1-го по K -е, потом — во 2-й ряд на места с 1-го по $K + 1$ -е, затем в 3-й ряд с 1-го места и т.д. Кроме этого, в этом концертном зале K мест в 1-м, 3-м, 5-м и всех нечётных рядах и $K + 1$ место во 2-м, 4-м и всех чётных рядах. Места в рядах нумеруются от 1 до K в нечётных рядах и от 1 до $K + 1$ в чётных рядах.

Входные данные: два целых числа. В первой строке записано число K ($1 \leq K \leq 10^9$) — количество мест в 1-м ряду кинотеатра. Во второй строке записано число N — порядковый номер в очереди ($1 \leq N \leq 2 \times 10^9$).

Выходные данные: два числа - номер ряда и номер места N -го проданного билета.

Решения, правильно работающие, когда входные числа не превосходят 1000, будут оцениваться в 60 баллов.

Пример работы программы

Ввод	Вывод	Примечание
5 58	11 3	Билеты с 1 по 5 будут проданы в первый ряд. Билеты с 6 по 11 будут проданы во второй ряд и т.д. 58-й билет окажется на 3-м месте 11-го ряда.

Решение

На 60 баллов можно было написать решение, суммирующее количество мест в каждом ряду: $K, K + 1, K, K + 1, \dots$ пока сумма станет не меньше N .

Для решения на полный балл заметим, что в двух соседних рядах сумма мест равна $2K + 1$, и если разбить ряды на пары, то можно определить, в какую пару рядов попадает N -е место, поделив и взяв остаток от деления на $2K + 1$. Также нужно потом рассмотреть один из двух случаев: в какой ряд (нечётный или чётный) попало наше место, сравнив остаток от деления на $2K + 1$ с числом мест в первом ряду K . Заметим, что в таких задачах удобней нумеровать объекты с 0. В приведённом ниже решении осуществляется переход в 0-нумерацию, вычитанием числа 1, а при выводе ответа прибавляется 1. Пример решения:

```
k = int(input())
n = int(input())
n -= 1
rows2 = n // (2 * k + 1)
n %= 2 * k + 1
if n < k:
    print(2 * rows2 + 1, n + 1)
else:
    print(2 * rows2 + 2, n + 1 - k)
```

Задача 4 (Время – 1 сек., 100 баллов)

У суперзвезды Ники очень большой гардероб, но обувь и сумочки она носит только чёрного или белого цвета. Однажды, собираясь на вечеринку, Ника решила надеть обувь и сумочку одного

цвета. И тут, как назло, погас свет в доме. Ника наугад достала X пар обуви и Y сумок, потом задумалась, какое минимально суммарное количество $(X+Y)$ пар обуви и сумок она должна взять, чтобы среди них гарантировано была бы пара обуви и сумочка одного цвета. Причем цвет комплекта неважен.

Помоги суперзвезде ответить на этот вопрос, если известно, в ее обширном гардеробе N белых и M черных пар обуви, K белых сумочек и T чёрных.

Входные данные: N, M, K, T – целые неотрицательные числа, не превосходящие 10^9 . N - количество белых пар обуви, M - количество черных пар обуви, K - количество белых сумочек и T - количество чёрных сумочек. Гарантируется, что в гардеробе есть одноцветный комплект пара обуви – сумочка.

Выходные данные: X и Y – целые неотрицательные числа, X – количество пар обуви, Y – количество сумочек, которое должна взять суперзвезда Ника. Сумма $(X+Y)$ должна быть минимальной.

Решение, правильно работающее только для случаев, когда все входные числа не превосходят 100, будет оцениваться в 35 баллов. Решение, правильно работающее только для случаев, когда все входные числа не превосходят 1000, будет оцениваться в 50 баллов. Решение, правильно работающее только для случаев, когда все входные числа не превосходят 10^5 , будет оцениваться в 70 баллов.

Пример работы программы

Ввод	Вывод	Примечание
10 10 10 10	11 1	В гардеробе имеется $N = 10$ белых пар обуви, $M = 10$ черных пар обуви, $K = 10$ белых сумочек и $T = 10$ чёрных сумочек. Если Ника возьмет 11 пар обуви, то среди них обязательно будет обувь и чёрного, и белого цвета. Поэтому достаточно взять всего одну сумочку, чтобы получился комплект одного цвета.

Решение:

Суперзвезда Ника для составления одноцветного комплекта может поступить несколькими способами:

1. Выбрать $N + 1$ пару обуви и $K + 1$ сумочку. Тогда поскольку количество белых пар обуви равно N , то ей обязательно попадет и чёрная пара. Также Нике обязательно попадет чёрная сумочка, если она возьмет $K + 1$ сумочку. Так, взяв $N + 1$ пару обуви и $K + 1$ сумочку, Ника обязательно получит чёрный комплект вещей.

2. Аналогично, можно получить комплект из белой пары обуви и белой сумки, взяв $M + 1$ пару обуви и $T + 1$ сумочку.

Решения, которые рассматривают только два этих случая, можно оценить в 50 баллов. Самый простой пример, на котором такое решение работает неправильно, это случай $N = M = K = T = 1$. Такое решение выдаст ответ (2, 2), хотя можно вытащить всего три предмета: например, (2, 1) или (1, 2). Это следующие способы выбора.

3. Взять $\max(N, M) + 1$ пару обуви. Тогда Ника гарантированно вытащит и чёрную, и белую обувь и тогда ей достаточно вытащить 1 сумочку, которая может быть любого цвета.

4. Взять 1 пару обуви и $\max(K, T) + 1$ сумочку. Решение, которое аккуратно разбирает все эти случаи (нужно не забыть, что ещё некоторые из данных чисел могут быть равны 0, поэтому не все указанные случаи возможны), набирает 100 баллов. Пример такого решения.

Пример решения на Python:

```
n = int (input ( ))
m = int (input ( ))
k = int (input ( ))
t = int (input ( ))
ans = [ ]
if n > 0 and k > 0 :
    ans . append ( [ m + 1 , t + 1 ] )
if m > 0 and t > 0 :
    ans . append ( [ n + 1 , k + 1 ] )
if n > 0 and m > 0 :
    ans . append ( [max( n , m ) + 1 , 1 ] )
if k > 0 and t > 0 :
    ans . append ( [ 1 , max( k , t ) + 1 ] )
s = min( ans , key=sum)
print (*s)
```

Здесь заводится список `ans`, в который будут складываться возможные варианты выбора Ники, например, $(m + 1, t + 1)$ и т.д. Каждый вариант — это пара чисел (тип данных кортеж). Кортежи мы будем добавлять в список, только проверив, что нет проблем с нулями, например, первый способ выбора применим только в случае $n > 0$ и $k > 0$.

Затем мы в списке `ans` находим наименьший элемент используя функцию `min` с параметром `key=sum`. Это означает, что кортежи сравниваются по результату вызова функции `sum` от каждого кортежа, т.е. мы находим кортеж с минимальной суммой элементов.

Также частичные баллы можно набрать при помощи «переборного» решения, рассматривающего все возможные ответы, выбирающее среди них подходящий и наилучший среди подходящих. Например, решение можно получить, перебирая количество выбранных пар обуви x от 1 до $n + m$ и количество выбранных сумочек y от 1 до $k + t$. Далее проверим, может ли пара (x, y) являться ответом. Для этого проще проверить, что пара (x, y) не может быть ответом. Это происходит, если все выбранные туфли и все выбранные сумки могут оказаться разноцветными, то есть если $x \leq n, y \leq t$ или $x \leq m, y \leq k$.

В приведённом ниже решении функция `bad(x, y)` проверяет это условие, то есть проверяет, что пара (x, y) не подходит. Поэтому в основной программе перебираются числа x и y и проверяется условие `not bad(x, y)`.

Пример переборного решения, набирающего 50 балла.

```
n = int (input ( ))
m = int (input ( ))
k = int (input ( ))
t = int (input ( ))
ans1 = n + m
ans2 = k + t

def bad ( x , y ) :
    return x <= n and y <= t or x <= m and y <= k

for x in range ( 1 , n + m + 1 ) :
    for y in range ( 1 , k + t + 1 ) :
```

```

if not bad ( x , y ) and x + y < ans1 + ans2 :
    ans1 = x
    ans2 = y
print ( ans1 , ans2 )

```

Переборное решение на 70 баллов можно получить, если перебирать только одно из чисел в ответе, а второе находить быстро.

Задача 5 (Время – 1 сек., 100 баллов)

Разведчик Светлов каждую неделю получает инструкции от куратора. Инструкции находятся в ячейке камеры хранения на Центральном вокзале или их приносит связной в условленное место. Куратор передает по каналу связи число K . Номер ячейки в камере хранения – минимальное число, произведение цифр которого равно K . Если ни одного подходящего решения не существует, то следует ожидать курьера. Помогите разведчику правильно определить номер ячейки с инструкциями.

Входные данные: натуральное число K , не превосходящее 10^{18} , — произведение цифр номера ячейки.

Обратите внимание, значение K может превышать возможное значение 32-битной целочисленной переменной, поэтому необходимо использовать 64-битные целочисленные типы данных (тип `long long` в языке C++, `int64` в Pascal, `long` в Java и C#).

Выходные данные: целое число — минимальное значение номера ячейки в камере хранения. Если ни одного подходящего номера не существует, программа должна вывести строку «XXX».

Решения, верно работающие в тех случаях, когда ответ не превосходит 10^5 , будут оцениваться не менее чем в 30 баллов.

Пример работы программы

Ввод	Вывод	Примечание
1024	2888	$2*8*8*8=1024$
51	XXX	Разложение числа 51 на простые множители $51=3*17$ не позволяет получить цифры от 2 до 9

Решение:

Нам нужно получить минимальное число, произведение цифр которого равно K . В частности, ответ должен содержать как можно меньше цифр. Также отметим, что в ответе не может быть цифры 1 (кроме случая $n = 1$).

В разложение числа K на простые множители могут входить только простые меньшие 10: 2, 3, 5, 7. С делителями 5 и 7 просто, они войдут в ответ, как отдельные цифры. А вот двойки и тройки можно сгруппировать так, чтобы цифр стало меньше. Три двойки можно заменить на 8: $2^3 = 8$. Две тройки можно заменить на 9: $3^2 = 9$. Осталось не более двух двоек и не более одной тройки. Если осталась одна двойка и одна тройка, их можно заменить на 6. Если после этого осталась две двойки, их можно заменить на 4. Если после этого остались двойка или тройка, то они войдут в ответ, как самостоятельные цифры. Полученные цифры нужно вывести в порядке убывания (от меньшей к большей).

Но есть и более простое решение. Оказывается, «выгодно» выделять как можно большие цифры в числе. Например, если $k = 18$ то из двух вариантов ответа 29 или 36 меньшим будет ответ 29. При этом в ответе 29 есть цифра 9, и если попробовать выделить сначала большую цифру 9, то

после деления $18/9$ получится меньший результат, который мы сможем использовать для старших разрядов записи ответа.

Получаем «жадное» решение — давайте делить наше число на все возможные цифры от 9 до 2 в порядке уменьшения, то есть попробуем в произведении выделить как можно больше девяток, затем как можно больше восьмёрок и т.д. Если в результате деления от числа k осталась 1, то нужно вывести найденные цифры в обратном порядке. Если же осталось число, большее 1, то значит у исходного k был делитель, больший 10, поэтому нужно вывести «XXX». Также нужно отдельно обработать случай, когда $k = 1$, тогда ответ также равен 1. В приведённом ниже решении это происходит, если строка `ans` пустая, то есть если ни одного однозначного делителя у числа k не было найдено.

Пример решения на языке Python.

```
k = int (input ( ))
ans = ""
for d in range ( 9 , 1 , -1):
    while k % d == 0 :
        k //= d
        ans = str ( d ) + ans
if k == 1 :
    print ( ans or "1" )
else :
    print ("XXX")
```

Решение на 30 баллов можно получить, если перебирать все числа, начиная с 1, пока не найдётся число, произведение цифр которого равно k . Пример такого решения.

```
k = int (input ( ))
ans = 1

def prod ( k ) :
    r e s = 1
    while n > 0 :
        r e s *= k % 10
        k //= 10
    return r e s

ans = 1
while prod ( ans ) != k :
    ans += 1
print ( ans )
```