

Задача А. Пароль

Данная задача является комбинаторной. Учащиеся могут решать ее по-разному (используя знания комбинаторики, рассуждениями с частичным перебором, написанием программы).

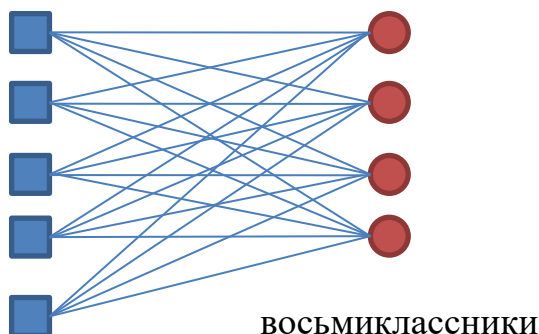
Учитывая, что переставляются цифры 8-значного пароля, среди которых есть повторяющиеся (1 повторяется три раза, 5 – два раза, 0 – два раза), то вычислить количество вариантов можно по формуле:

$$\frac{8!}{3! \cdot 2! \cdot 2!} = \frac{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8}{6 \cdot 2 \cdot 2} = 1680$$

Ответ: 1680

Задача В. Пейнтбол

Схематично взаимные попадания можно изобразить так:



семиклассники

Если предположим, что количество семиклассников – x , а восьмиклассников – y , то количество попаданий, соответственно будет $4y$ и $5x$. Учитывая, что количество попаданий одинаковое, то $4y = 5x$. Отсюда следует, что y должно быть кратно 5, а $x =$ четырем. Перебирая пары подходящих чисел и учитывая, что $x + y \leq 100$, находим, значения $x = 44$, $y = 55$.

Значит, общее количество равно 99.

Ответ: 99

Задача С. Посадка деревьев

Проанализировав количество деревьев на картинках, можно выявить связь между количеством рядов посадок и количеством плодовых деревьев и кустов:

Ряды N	Плодовые деревья P	Кусты K
1	1	8
2	4	16
3	9	24

$P = N^2$ – количество плодовых деревьев

$K = 8N$ – количество кустов

Отсюда, ответ на подзадачу С1: $8 \cdot 5 = 40$

Отсюда, ответ на подзадачу С2: 8 (случай, когда $8N = N^2$)

Ответ: 40

8

Задача D. Упаковка подарка

Длина ленты состоит из следующих частей:

- две ширины коробки (по верхней и нижней крышке)
- две длины коробки (по верхней и нижней крышке)
- четыре высоты коробки (по боковым сторонам).

Эту длину следует удвоить, т.к. такое же количество ленты требуется для банта.

Пример программы на языке Pascal ABC

```
program gift;  
var a, b, c, len: integer;  
begin  
  read(a);  
  read(b);  
  read(c);  
  len := 2 * (2 * a + 2 * b + 4 * c);  
  writeln(len);  
end.
```

Задача E. Странный указатель

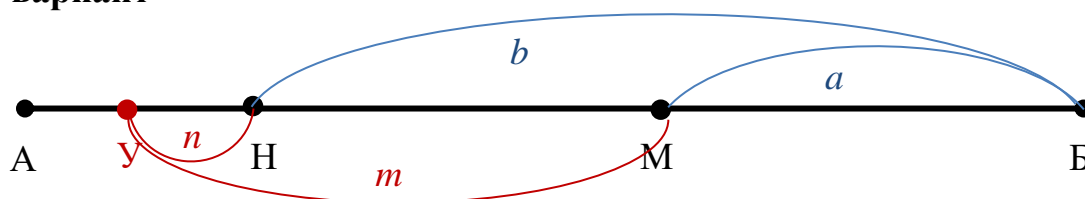
Рассмотрим возможные схемы расположения населенных пунктов на дороге от Айска (А) и Бейска (Б).

На схеме М – деревня Мская, Н – село Нское, У – указатель.

Учитывая, что $1 \leq A < B \leq 1000$, т.е. расстояние от Бейска от деревни Мской, B – расстояние от Бейска до села Нского, то деревня Мская гарантированно расположена ближе к Бейску, чем село Нское.

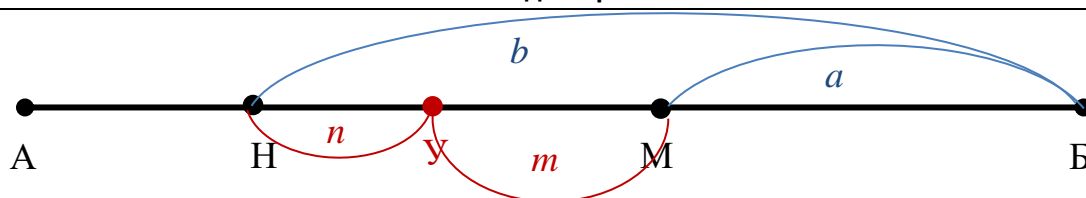
Тогда для расположения указателя возможны три варианта:

1 вариант



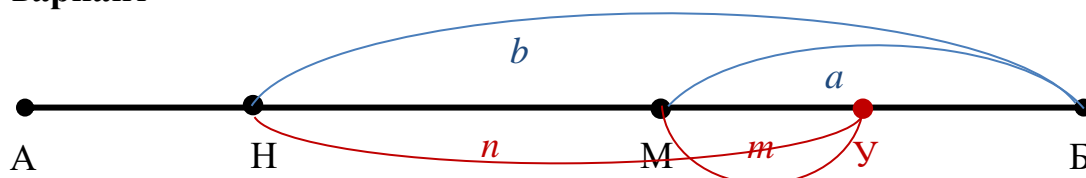
При таком расположении длину отрезка НМ можно найти как $b - a$ или как $m - n$. Если это равенство выполняется, то от указателя до Бейска расстояние $a + m$ (или $b + n$)

2 вариант



При таком расположении должно выполняться соотношение $b = n + m + a$. Если это равенство выполняется, то от указателя до Бейска расстояние $a + m$ (или $b - n$)

3 вариант



При таком расположении длину отрезка HM можно найти как $b - a$ или как $n - m$. Если это равенство выполняется, то от указателя до Бейска расстояние $a - m$ (или $b - n$)

Пример программы на языке Pascal ABC

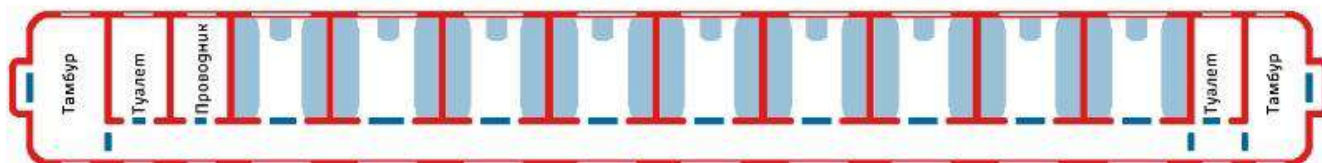
```
var a, b, m, n:integer;
begin
  read(a);
  read(b);
  read(m);
  read(n);
  if (b - a = m + n) or (b - a = m - n) then
    writeln(a + m)
  else
    writeln(a - m);
end.
```

Задача F. Поезд

По рисунку видно, что в вагоне 9 купе. Если их пронумеровать, начиная с 0, то в 0 купе будут места 0 – 3, в 1 купе – 5-7, в i купе – $4i - 4i + 3$.

Места:

2, 4	6, 8	10, 12	14, 16	18, 20	22, 24	26, 28	30, 32	34, 36
1, 3	5, 7	9, 11	13, 15	17, 19	21, 23	25, 27	29, 31	33, 35



Заведем массив из 9 элементов; каждый элемент соответствует номеру купе и хранит информацию о количестве занятых в купе мест. По номеру места легко определить, в каком купе оно находится, если выполнить целочисленное деление на 4.

Пример программы на языке Pascal ABC

```
var N, seat, number, i: integer;
    coupe: array [0..8] of integer;

begin
  read(N);
  for i := 0 to 8 do
    coupe[i] := 0;
  for i:= 1 to N do
  begin
    read(seat);
    coupe[(seat - 1) div 4] := coupe[(seat - 1) div 4] + 1;
  end;
  number := -1;
  for i:= 0 to 8 do
    if coupe[i] = 0 then
      begin
        number := i;
        break;
      end;
  if number = -1 then
    writeln(0)
  else
    writeln(number * 4 + 1, ' ', number * 4 + 4);
end.
```