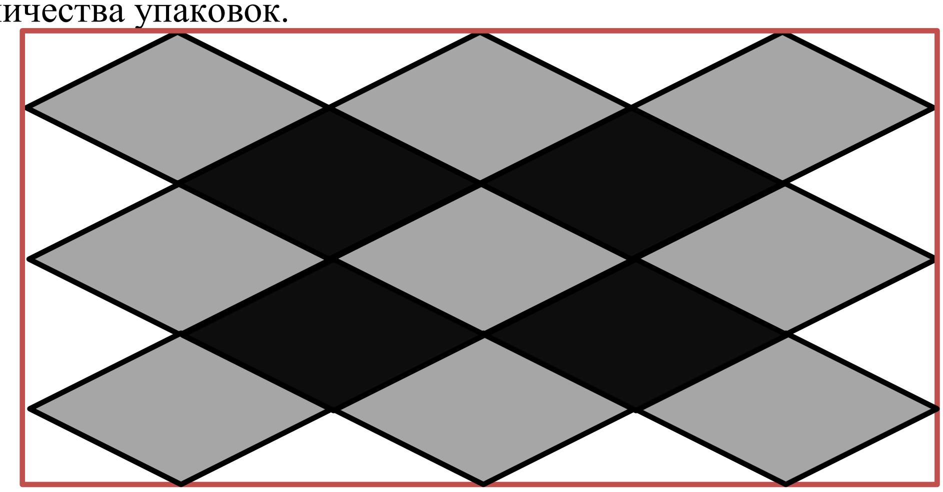
Разбор задач

Задача А. Укладка плитки

Выведем формулу для подсчета количества плитки, а затем для подсчета количества упаковок.



Посчитаем количество плитки в рядах, выделенных на рисунке серым цветом. Количество плитки в каждом ряду $\frac{N}{a}$, таких рядов – $\frac{M}{b}$, поэтому плиток в серых рядах будет $\frac{N}{a} \cdot \frac{M}{b}$

Можно заметить, что в рядах, выделенных черным, на одну плитку меньше, чем в серых. И самих рядов также на один меньше. Поэтому количество плиток в черных рядах $\left(\frac{N}{a} - 1\right) \cdot \left(\frac{M}{b} - 1\right)$

Тогда общее количество плитки:

$$\frac{N}{a} \cdot \frac{M}{b} + \left(\frac{N}{a} - 1\right) \cdot \left(\frac{M}{b} - 1\right)$$

А количество коробок плитки можно посчитать, разделив эту величину на K (округлив значение вверх до ближайшего целого)

Пример решения на языке Pascal ABC.NET

```
var n, m, a, b, k, tile, container: longint;
begin
  read(n, m, a, b, k);
  tile := (n div a)*(m div b)+(n div a -1)*(m div b - 1);
  container := (tile + k - 1) div k;
  write(container);
end.
```

Задача B. Непарная обувь

Упорядочим исходные числа по возрастанию, для удобства будем считать, что $a \leq b \leq c$. Рассмотрим два случая.

1 случай. Пусть $a + b \leq c$. Тогда Тролль сможет сделать непарную обувь в $(a + b)$ коробках. Тогда в каждую коробку можно положить туфлю и кроссовок (таких коробок будет a штук) или ботинок и кроссовок (таких коробок будет b штук). И часть кроссовок останется неиспользованными.
Поскольку первые два множества содержат всего $a + b$ элементов, сделать больше чем $a + b$ коробок не удастся.
2 случай. Пусть $a + b > c$. Тогда Тролль сможет испортить $\frac{a+b+c}{2}$ коробок (подразумевается целочисленное деление).
Доказать это можно, например, так. Тролль уравнивает количества обуви разных видов, приготовив вначале $(c - b)$ коробок из туфель и кроссовок, после чего остаётся $x = a - (c - b)$ туфель и $c - (c - b) = b$ кроссовок. Затем из ботинок и кроссовок он делает еще $(b - x)$ коробок непарной обуви, и значит, останется $b - (b - x) = x$ ботинок и столько же кроссовок. Теперь у Тролля x туфель, x ботинок и x кроссовок, из которых получаются $\frac{3x}{2}$ коробок.
Всего испорчено $(c - b) + (b - x) + \frac{3x}{2} = \frac{a+b+c}{2}$ коробок.
Поскольку все три множества содержат $a + b + c$ элементов, очевидно, что сделать больше чем $\frac{a+b+c}{2}$ коробок непарной обуви не удастся.
Следует обратить внимание на выбор подходящего типа данных, вмещающего в себя исходные значения и их суммы.

Пример решения на языке Pascal ABC.NET

```
var a, b, c, x, y, z, box: uint64;
begin
  read(x, y, z);
  a := min(min(x, y), z);
  c := max(max(x, y), z);
  b := x + y + z - a - c;
  if (a + b) <= c then
    write(a + b)
  else
    write((a + b + c) div 2);
end.
```

Задача C. Суперматрешка

Все числа каждого набора отсортируем по возрастанию. Теперь осталось проверить, что каждое число набора не меньше суммы всех предыдущих. Если это условие выполняется для всех k чисел набора (кроме первого), выводим **yes**; иначе — выводим **no**.

Пример решения на языке Pascal ABC.NET

```
var i, k, n, j:integer;
   s: uint64;
   a: array of uint64;
   f: boolean;
begin
  read(n, k);
  Setlength(a, k);
  for i := 1 to n do
  begin
    for j := 0 to k-1 do
      read(a[j]);
    sort(a);
    s := a[0];
    f := True;
    for j := 1 to k - 1 do
    begin
      if  s > a[j] then begin f := False; break; end;
      s := s + a[j];
    end;
    if f then
      writeln('yes')
    else
      writeln('no');
  end;
end.
```

Задача D. Инвестиции

Во-первых, обратим внимание, что любую сумму, большую или равную 8, можно распределить по 3 или по 5, не получив остатка. Если встретятся суммы, которые можно распределить разными способами (например, 15 можно разделить на 5 вкладов по 3, а можно – на 3 вклада по 5), то выберем вариант, где больше будет вкладов типа B, так как он прибыльнее.

Стратегия распределения будет следующей:

- если число кратно 5, то распределяем все средства на вклады B;
- если число при делении на 5 дает остаток 1, то тогда «заберем» один вклад B (в сумме с остатком 1 он даст 6 у.е.) и добавим два вклада A;
- если число при делении на 5 дает остаток 2, то тогда «заберем» два вклада B (в сумме с остатком 2 они дадут 12 у.е.) и добавим три вклада A;
- если число при делении на 5 дает остаток 3, то добавим вклад A;

– если число при делении на 5 дает остаток 4, то тогда «заберем» один вклад B (в сумме с остатком 4 он даст 9 у.е.) и добавим три вклада A.
Дальше будем прибавлять к имеющимся средствам прибыль из расчета:

$$5 * количество\_вкладов\_A \quad + 9 * количество\_вкладов\_B$$

При этом следует отдельно сохранять получающуюся к концу года прибыль и капитал в начале года, чтобы, начиная с третьего года вычитать величину комиссии.

Пример решения на языке Pascal ABC.NET

```
var
  i, n: integer;
  k, k3, k5: integer;
  a, b: array[1..101] of uint64;

begin
  read(k, n);
  n := n + 1;
  for i := 1 to n do
    begin
    a[i] := 0;
    b[i] := 0;
    end;

  a[1] := k;
  b[1] := k;

  for i := 2 to n do
  begin
    k := a[i - 1];
    k5 := k div 5;
    case k mod 5 of
    0: k3 := 0;
    1: begin k5 := k5 - 1; k3 := 2; end;
    2: begin k5 := k5 - 2; k3 := 4; end;
    3: k3 := 1;
    4: begin k5 := k5 - 1; k3 := 3; end;
    end;
    if (i = 2) or (i = 3) then
    begin
      b[i] := k5 * 9 + k3 * 5;
      a[i] := a[i - 1] + b[i];
     end;
```

```
    if i >= 4 then
    begin
      b[i] := k5 * 9 + k3 * 5;
      a[i] := a[i - 1] + b[i]  - b[i - 3];
    end;
  end;
  write(a[n]);
end.
```

Задача E. Пирамиды для Мити

Учитывая, что в каждом следующем ряду ровно на один кубик меньше, чем в предыдущем, то количество кубиков в наборе можно представить в виде суммы подряд идущих чисел $m+1, m+2, \dots, m+n$.

Эту сумму можно преобразовать:

$$(m+1)+(m+2)+\ \dots+(m+n)=m\cdot n+(1+2+\ \dots+n)$$
$$=m\cdot n+\frac{(1+n)}{2}\cdot n=n\cdot\left(m+\frac{1+n}{2}\right)=\frac{1}{2}\cdot n\cdot(2m+n+1)$$

Очевидно, что множители n и $(2m+n+1)$ разной четности, поэтому ровно один из них нечетный. Значит, искомое количество кубиков должно иметь нечетный делитель.

Можно доказать обратное утверждение, что каждое разложение на сумму подряд идущих чисел соответствует нечетному делителю.

Поэтому количество вариантов построения пирамидки соответствует количеству нечетных делителей числа.

Осталось для каждого числа из набора вычислить количество нечетных делителей, а затем найти наименьшее число с максимальным количеством нечетных делителей.

Пример решения на языке Pascal ABC.NET

```
var n, a,  i, j, count, k, b, best, max_k: longint;
begin
  read(n);
  best := 2000000000;
  max_k := 0;
  for i  := 1 to n do
  begin
    read(a);
    b := a;
    while a mod 2 = 0 do
      a := a div 2;
    k := 0;
```

```
    for j := 1 to trunc(sqrt(a)) do
      if a mod j = 0 then k := k + 2;
    if sqr(trunc(sqrt(a))) = a then k := k - 1;
    if k > max_k then
      begin
      best := b;
      max_k := k;
      end
    else
      if k = max_k then
        if b < best then
          best := b;

  end;
 write(best, ' ', max_k);
end.
```