

Разбор задачи «Складирование ноутбуков»

автор разбора — О. Вечкасова

Нетрудно понять, что первый ноутбук надо класть в угол склада — ведь если есть способ расположить максимальное число компьютеров без соблюдения этого условия, то всегда можно передвинуть и переместить некоторые стопки так, чтобы какой-нибудь из ноутбуков попал в угол комнаты.

Считаем, что первый ноутбук уже есть на складе. Остальные будем класть согласно правилам (то есть в такой же ориентации, как и первый) вплотную друг к другу и к стенам, пока хватает размеров склада. Очевидно, что при фиксированном первом такой способ расположения самый экономный. Пусть A, B, C — размеры склада, а a, b, c — размеры ноутбука, причем стороны, соответствующие одинаковой букве, параллельны. Тогда количество поместившихся на склад компьютеров можно вычислить по формуле $(A \operatorname{div} a) * (B \operatorname{div} b) * (C \operatorname{div} c)$, где div — целочисленное деление.

Занумеруем стороны склада и первого ноутбука числами от 1 до 3. Всего вариантов положить первый ноутбук на склад ровно 6: когда его сторона 1 параллельна стороне 1 склада, 2 — 2 и 3 — 3, и так далее, последний вариант — когда его сторона 1 параллельна стороне 3 склада, 2 — 2 и 3 — 1.

Теперь чтобы решить задачу нам необходимо перебрать 6 вариантов расположения ноутбука, для каждого посчитать, сколько ноутбуков помещается на складе, и выбрать максимум из полученных результатов. Ниже приведено решение этой задачи на паскале:

```
program notebooks_ov;

{$APPTYPE CONSOLE}
{$R+}

uses
  SysUtils;

var skl, nt : array [1..3] of integer;
    i, j, m : integer;

function max(a, b : integer):integer;
begin
  if a > b then max := a else max := b;
end;

function count(i, j : integer):integer;
var a, b, c : integer;
begin
  a := skl[1] div nt[i];
  b := skl[2] div nt[j];
  c := skl[3] div nt[6-i-j];
  result := a*b*c;
```

end;

begin

```
  reset(input, 'a.in');      { b.in и b.out либо без }
```

```
  rewrite(output, 'a.out'); { файлов для лиги Б      }
```

```
  read(sk1[1], sk1[2], sk1[3], nt[1], nt[2], nt[3]);
```

```
  m := 0;
```

```
  for i := 1 to 3 do
```

```
    for j := 1 to 3 do
```

```
      if i <> j then m := max(m, count(i, j));
```

```
    writeln(m);
```

```
  close(input);
```

```
  close(output);
```

end.