

### Проект 3. Игра в монеты

Однокопеечные монеты разложены в  $n$  стопок (в стопках может быть различное количество монет), и стопки поставлены на столе в ряд слева направо. Двое игроков по очереди делают ходы. Ход состоит в том, что один из игроков берет слева направо несколько стопок подряд, не меньше одной, но не более чем перед этим взял его соперник. Первый игрок своим первым ходом берет не более  $k$  стопок. Игра заканчивается, когда все стопки будут взяты. Требуется составить программу, определяющую максимальное число монет, которое может получить первый игрок после окончания игры, если второй игрок тоже старается ходить так, чтобы получить как можно больше монет.

Исходная информация хранится в файле 'input.txt', состоящем из одной строки чисел. В строке находится число стопок  $n$ , за ним идут  $n$  чисел – количество монет в стопках слева направо, затем в строке записано число  $k$ . Числа разделены пробелами.

Ограничения:  $1 \leq n \leq 100$ ,  $1 \leq k \leq 50$ , количество монет в стопках не более 10000, время выполнения программы 1 секунда.

Программа должна вывести ответ – максимальное число монет, которое может получить первый игрок, в файл 'output.txt'.

Решение: Введем обозначения:  $s[r]$  – суммарное количество монет в стопках с  $r$  – той по  $n$  – ю;  $g[r, m]$  – максимальное количество монет, которое может получить игрок, если сейчас его ход, при условии что остались стопки с  $r$  – той по  $n$  – ю, и он может взять не более  $m$  стопок. Элементы массива  $s[1..n]$  будут использоваться для вычисления элементов массива  $g[n, k]$ , поэтому его нужно вычислить и сохранить сразу после ввода данных.

Далее вычисляем элементы массива  $g$ . Если игрок может взять все оставшиеся стопки, то он должен сделать это, иначе другой игрок возьмет не менее одной стопки из оставшихся стопок, и выигрыш игрока, чей сейчас ход, не будет максимальным. Следовательно, все числа  $g[r, m]$ , где  $m \geq n - r + 1$ , определены, в том числе все элементы  $g[n, m]$ .

$$g[r, m] = s[r]. \quad (1)$$

Будем находить элементы  $g[r, m]$  в порядке убывания  $r$ , при условии  $m < n - r + 1$ . В этой ситуации игрок может взять от одной до  $m$  стопок. Пусть он взял  $i$  стопок. Тогда другой игрок окажется в ситуации  $(r + i, i)$  – его ход, разыгрываются стопки с  $(r + i)$  – той по  $n$  – ю, и можно взять не более  $i$  стопок. Он может взять максимальное количество монет, равное  $g[r + i, i]$ . Так как в стопках с  $r$  – той по  $n$  – ю было  $s[r]$  монет, то первый игрок, взяв  $i$  стопок, получит  $s[r] - g[r + i, i]$  монет.

Поэтому максимальный выигрыш первого игрока определяется по формуле:

$$g[r, m] = \max_{1 \leq i \leq m} (s[r] - g[r + i, i]). \quad (2)$$

Ответом задачи будет число  $G[1, k]$  – столько монет может получить первый игрок, если разыгрываются стопки с первой по  $n$ -ю и можно взять не более  $k$  стопок.

Пример.

Пусть в файле 'input.txt' записана строка чисел: 7 5 7 2 3 8 9 3 2.

Вычислим элементы массива  $s[1..7]$  и запишем в таблице:

I	1	2	3	4	5	6	7
S[i]	37	32	25	23	20	12	3

Теперь заполняем таблицу  $g[7, 2]$  по формулам (1) и (2):

r \ m	1	2
1	18	24
2	19	20
3	13	13
4	12	12
5	11	17
6	9	12
7	3	3

Ответ:  $g[1, 2] = 24$ .

Рассмотренный алгоритм решения игры реализован в программе:

```

program Project1;
uses crt;
var  p,s:array[1..100]of integer;
     g:array[1..100,1..50] of integer;
     k,n,m,r,t,max,i,j:integer;
begin assign(input,'input.txt');
reset(input);read(n); for i:=1 to n do read(p[i]); read(k);close(input);
{-----massivS-----}
s[n]:=p[n];
for i:=n-1 downto 1 do s[i]:=s[i+1]+p[i];
{-----massivG-----}
for r:=n downto 1 do
for m:=k downto 1 do
if m<n-r+1 then begin max:=0;
for i:=1 to m do begin t:=s[r]-g[r+i,i];
if t>max then max:=t;
end;
g[r,m]:=max;
end
else g[r,m]:=s[r];
{-----otwet-----}
max:=0; for i:=1 to k do if g[1,i]>max then max:=g[1,i];

```

```
assign(output,'d:\output.txt'); rewrite(output); writeln(max); close(output);  
end.
```

Пример. Пусть в файле 'input.txt' записана строка чисел:

7 5 7 2 3 8 9 3 2.

Тогда программа запишет в файл 'output.txt' число 24.