

На складе имеется  $n = 5$  неделимых предметов. Для каждого предмета известна его масса (в кг.) и стоимость (в условных единицах), выражающиеся в натуральных числах:

Масса $m_i$	4	5	3	7	6
Стоимость $c_i$	5	7	4	9	8

Определить максимальную суммарную стоимость предметов, которые можно унести со склада, при условии, что суммарная масса предметов не должна превышать  $p = 15$  кг, и номера унесенных предметов.

Решение: Будем считать, что предметы пронумерованы в порядке их следования в таблице с исходными данными. Пусть  $T$  – функция, значение которой равно максимальной стоимости унесенных со склада предметов при условии, что на складе имеется  $n$  предметов, и можно унести не более  $p$  килограммов. Функция  $T$  имеет два аргумента – количество предметов и максимальную суммарную массу предметов, которую можно унести. При заданных значениях величин  $n$  и  $p$  решению задачи соответствует значение функции  $T(n, p) = T(5, 15)$ .

Для задачи  $T(n, p)$  определим подзадачи  $T(i, j)$ , где  $i$  обозначает количество начальных предметов, из которых осуществляется выбор, а  $j$  – максимально возможную суммарную массу уносимых предметов. Аргумент  $i$  задает количество предметов для подзадачи и одновременно значения их масс и стоимостей из таблицы исходных данных.

**Определим рекуррентные соотношения, позволяющие вычислять значения функции  $T$ .**

При значениях аргументов функции равно нулю	нулевых одного из значение	$T(0, 0) = 0,$ $T(0, j) = 0,$ при $j \geq 1,$ $T(i, 0) = 0,$ при $i \geq 1.$	$t[0,0]:=0;$ for $j:=1$ to $p$ do $t[0,j]:=0;$ for $i:=1$ to $n$ do $t[i,0]:=0;$
---	----------------------------	--	--

Определим возможные значения функции  $T(i, j)$  при ненулевых значениях аргументов, учитывая, что решение подзадачи, соответствующей функции  $T(i, j)$ , может быть сведено к выбору одного из двух вариантов:

<b>Предмет с номером <math>i</math> уносится со склада.</b> это уменьшает суммарную массу для $i - 1$ предметов на величину $m_i$ , увеличивая $T(i-1, j - m_i)$ на $c_i$ $m_i \leq j.$	$T(i, j) = T(i-1, j - m_i) + c_i$	<pre> for i:=1 to n do   for j:=1 to p do     if j&gt;=m[i]       then t[i,j]:=max(t[i-1,j],t[i-1,j-         m[i]]+c[i])       else t[i,j]:=t[i-1,j];           </pre>
<b>Предмет с номером <math>i</math> остается на складе.</b> решение задачи с $i$ предметами сводится к решению подзадачи с $i - 1$ предметами	$T(i, j) = T(i - 1, j).$	

Используя рекуррентное соотношение, получаем таблицу значений функции  $T$ , которая выглядит следующим образом:

$i \setminus j$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	5	5	5	5	5	5	5	5	5	5	5	5
2	0	0	0	0	5	7	7	7	7	12	12	12	12	12	12	12
3	0	0	0	4	5	7	7	9	11	12	12	12	16	16	16	16
4	0	0	0	4	5	7	7	9	11	12	12	14	16	16	18	20

5	0	0	0	4	5	7	8	9	11	12	13	15	16	17	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----

Таким образом, максимальная суммарная стоимость предметов, которые можно унести со склада,  $T(5,15) = 20$ .

Список уносимых со склада предметов будем формировать, “с конца”. Рассмотрим элементы таблицы функции  $T(5, 15)$  и  $T(4, 15)$ . Так как значения обоих этих элементов равны 20, то это значит, что можно набрать стоимость 20 (у. е.) без пятого предмета, используя в списке только предметы с номерами 1 – 4, то есть пятый предмет в список можно не включать.

Теперь рассмотрим элементы  $T(4, 15)$  и  $T(3, 15)$ . Их значения не равны,  $16 < 20$ . Это означает, что четвертый предмет должен быть обязательно включен в список уносимых предметов (без пятого предмета), а масса других предметов списка должна быть не больше  $15 - m_4 = 15 - 7 = 8$  (кг.).

Далее рассмотрим элементы  $T(3, 8)$  и  $T(2, 8)$ . Их значения не равны,  $7 \neq 11$ , значит, третий предмет включается в список, а масса других предметов списка должна быть не больше  $8 - m_3 = 8 - 3 = 5$  (кг.).

Теперь рассмотрим элементы  $T(2, 5)$  и  $T(1, 5)$ . Их значения не равны,  $7 > 5$ , поэтому второй предмет включается в список. Так как его масса равна 5 (кг.), то первый предмет не может включаться в список.

```
writeln(f,t[n,p]);
  for i:=n downto 1 do
    if t[i,sum]<>t[i-1,sum] then
      begin
        write(f,i:3);
        sum:=sum-m[i]
      end;
```

Таким образом, в искомый список входят предметы с номерами 2, 3, и 4.

Программа `rucksack` читает данные из файла ‘input.txt’, в котором в первой строке записаны два числа: количество предметов –  $n$  и суммарная масса предметов, которую можно унести со склада –  $p$ , а в следующих двух строках – массы предметов и их стоимости. Данные из файла считываются процедурой `readf`.

Кроме вывода результатов работы на экран, в программе предусмотрена их запись в текстовый файл ‘output.txt’, как это обычно требуется при решении задачи на олимпиадах по программированию.

Предполагается, что  $n \leq 20$ , а  $p \leq 100$ .

Пример. Пусть в файле ‘input.txt’ записана информация:

```
5 15
4 5 3 7 6
5 7 4 9 8
```

Программа выведет ответ:

Максимальная стоимость 20.

Берем предметы с номерами: 2 3 4.

Ответ также будет записан в файл ‘output.txt’.

```
program rucksack; {задача о рюкзаке}
const mn=20; mp=100;
var i,j,n,p,sum : integer;
    m,c:array[1..mn] of integer;
    t:array[0..mn,0..mp] of integer;
function max(x,y:integer):integer;
```

```

begin if x>y then max:=x else max:=y;
end;
procedure readf;
begin assign(input,'input.txt'); reset(input);
  readln(n,p); {количество предметов и грузоподъемность}
  writeln('n=',n,' p=',p);
  writeln('массивы весов и стоимостей предметов: ');
  for i:=1 to n do
  begin read(m[i]); write(m[i]:4);
  end; writeln;
  for i:=1 to n do
  begin read(c[i]); write(c[i]:4);
  end; writeln;
  close(input);
end;
begin clrscr; readf;
  t[0,0]:=0;
  for j:=1 to p do t[0,j]:=0;
  for i:=1 to n do t[i,0]:=0;
  for i:=1 to n do
  for j:=1 to p do
  if j>=m[i] then t[i,j]:=max(t[i-1,j],t[i-1,j-m[i]]+c[i]) else t[i,j]:=t[i-1,j];

  writeln('max_st=',t[n,p]);
  writeln('berem predmety s N:');
  sum:=p;
  assign(output,'output.txt'); rewrite(output);
  writeln(t[n,p]);
  for i:=n downto 1 do
  if t[i,sum]<>t[i-1,sum] then
  begin write(i:3);
  sum:=sum-m[i]
  end;
  close(output);
end.

```