

Всероссийская олимпиада школьников по информатике 2014-15  
Санкт-Петербург, Муниципальный этап

Разбор задач

Задача А. Сравнение комнат

Для решения этой задачи достаточно было считать входные данные, вычислить площадь комнат, перемножив для каждого прямоугольника длину на ширину, и сравнить полученные числа.

Приведем пример решения задачи на различных языках программирования.

Паскаль

```
program problemA;
var
  a, b, c, d: integer;
begin
  read(a, b, c, d);
  if (a * b) > (c * d) then
    writeln('M')
  else if (a * b) < (c * d) then
    writeln('P')
  else
    writeln('E')
end.
```

Си++

```
#include <iostream>
using namespace std;

int main()
{
  int a, b, c, d;
  cin >> a >> b >> c >> d;
  cout << ((a * b > c * d) ? "M" : (a * b == c * d) ? "E" :
    "P") << endl;
  return 0;
}
```

Python

```
a, b, c, d = map(int, input().split())

if a * b > c * d:
    print("M")
elif a * b < c * d:
    print("P")
else:
    print("E")
```

## Задача В. Рамки

Сделаем в точности то, что просят в условии. Заведем двумерный массив размером  $h$  на  $w$ . Для каждого прямоугольника заполним его рамку соответствующими символами. После этого заполненный массив по строкам.

Приведем пример решения задачи на различных языках программирования.

Паскаль

```
program problemB;
var
  i, j, k, a, b, c, d, h, w, n: integer;
  ans: array[1..80, 1..80] of char;
  cur: char;
begin
  read(h, w, n);
  for i := 1 to h do
    for j := 1 to w do
      ans[i, j] := '.';
  for k := 0 to n - 1 do begin
    cur := Chr(Ord('a') + k);
    read(a, b, c, d);
    for i := a to c do begin
      ans[i, b] := cur;
      ans[i, d] := cur
    end;
    for i := b to d do begin
      ans[a, i] := cur;
      ans[c, i] := cur
    end
  end;
  for i := 1 to h do begin
    for j := 1 to w do
      write(ans[i, j]);
    writeln();
  end
end.
```

Python

```
h, w, n = map(int, input().split())
c = [['.'] * w for i in range(h)]

for i in range(n):
    r1, c1, r2, c2 = map(lambda x: int(x) - 1, input().split())
    for p in range(c1, c2 + 1):
        c[r1][p] = chr(ord('a') + i)
        c[r2][p] = chr(ord('a') + i)
    for p in range(r1, r2 + 1):
        c[p][c1] = chr(ord('a') + i)
        c[p][c2] = chr(ord('a') + i)

for s in c:
    print("".join(s))
```

## Задача С. Урок физкультуры

Отсортируем школьников в соответствии с принципом, описанным в условии. После этого осталось найти максимум разностей соседних элементов. Сделаем это линейным проходом по массиву. Поскольку ограничения на число школьников были достаточно маленькие, можно было воспользоваться любым алгоритмом сортировки, например, сортировкой пузырьком.

Приведем пример решения задачи на различных языках программирования.

Паскаль (нет встроенной сортировки)

```
program problemC;
var
  i, j, a, h, n, bc, gc, tmp, ans: integer;
  b, g: array[1..50] of integer;
begin
  read(n);
  bc := 0;
  gc := 0;
  for i := 1 to n do begin
    read(a, h);
    if a = 0 then begin
      inc(bc);
      b[bc] := h
    end else begin
      inc(gc);
      g[gc] := h
    end
  end;
  for i := 1 to bc do
    for j := i + 1 to bc do
      if b[i] < b[j] then begin
        tmp := b[i];
        b[i] := b[j];
        b[j] := tmp
      end;
    for i := 1 to gc do
      for j := i + 1 to gc do
        if g[i] < g[j] then begin
          tmp := g[i];
          g[i] := g[j];
          g[j] := tmp
        end;
    ans := 0;
    for i := 1 to bc - 1 do
      if ans < b[i] - b[i + 1] then
        ans := b[i] - b[i + 1];
    for i := 1 to gc - 1 do
      if ans < g[i] - g[i + 1] then
        ans := g[i] - g[i + 1];
    if (bc > 0) and (gc > 0) then
      if ans < abs(g[1] - b[bc]) then
        ans := abs(g[1] - b[bc]);
    writeln(ans)
end.
```

Python (используя встроенную сортировку)

```
n = int(input())
p = []
for i in range(n):
    a, h = map(int, input().split())
    p.append((a, h))
p.sort(key = lambda v: (v[0], -v[1]))
ans = max([p[i][1] - p[i - 1][1] for i in range(1, n)])

print(ans)
```

## Задача D. Интересные разбиения

Для генерации всех интересных разбиений применим рекурсию. В качестве параметров будем передавать уже сгенерированный фрагмент разбиения, последнее использованное число и сумму, которую осталось набрать. Когда разбиение полностью готово, выводим его.

Приведем пример решения задачи на различных языках программирования.

Паскаль

```
program problemD;
var
    i, p, n: integer;
    a: array[1..80] of integer;

procedure gen(cur, len, n: integer);
var
    i: integer;
begin
    if cur = 0 then begin
        if len > 1 then begin
            write(n, '=');
            for i := 1 to len - 2 do
                write(a[i], '+');
            writeln(a[len - 1])
        end
    end else if cur > 0 then begin
        p := -1;
        if len > 1 then
            p := a[len - 1];
        for i := p + 2 to cur do begin
            a[len] := i;
            gen(cur - i, len + 1, n)
        end
    end
end;

begin
    read(n);
    gen(n, 1, n);
end.
```

Python

```
def gen(s, n, x, a):
    if x == 0:
        print(str(n) + "=" + "+".join([str(x) for x in s]))
        return
    for b in range(x, a - 1, -1):
        if x - b >= b + 2 or x == b:
            gen(s + [b], n, x - b, b + 2)

n = int(input())
gen([], n, n, 1)
```

## Задача Е. Сумма минимумов.

Используем динамическое программирование.

Обозначим как  $rmq[i][j]$  минимальный элемент в массиве между  $i$ -й и  $j$ -й позицией, включительно. Для вычисления  $rmq[i][j]$  можно использовать следующие формулы:

```
rmq[i][i] = 1
rmq[i][j] = min(rmq[i][j - 1], a[j])
```

Обозначим как  $d[r][c]$  максимальную сумму минимумов, которую можно получить на отрезке с первого по  $r$ -й элемент, если разбить его на  $c$  частей. Тогда

```
d[r][c] = min(d[r - k][c - 1] + rmq[r - k + 1][r]) для всех k от 1 до r
```

Ответ содержится в элементе  $d[n][k]$ .

Для восстановления самого разбиения необходимо также запоминать, для какого  $k$  достигается минимум, это позволяет восстановить ответ.

Следует также обратить внимание на возможность переполнения 32-битного типа данных в этой задаче. Те, кто использовал 32-битный тип данных, набирали 60 баллов.

Приведем пример решения задачи на различных языках программирования.

Паскаль

```
program problemE;
var
    i, j, n, k, l, r, cur, asize: int64;
    a, ans: array[1..300] of int64;
    rmq, d, pr: array[0..300, 1..300] of int64;
begin
    read(n, k);
    for i := 1 to n do
        read(a[i]);
    for l := 1 to n do
        for r := 1 to n do begin
            cur := a[l];
            for i:= 1 to r do
                cur := min(cur, a[i]);
            rmq[l, r] := cur
        end;
    for i := 1 to n do
        d[0, i] := m[1, i];

    for i := 1 to k - 1 do
        for j := 1 to n do
            for l := 1 to j do
                if d[i, j] < d[i - 1, l] + m[l + 1, j] then begin
                    d[i, j] := d[i - 1, l] + m[l + 1, j];
                    pr[i, j] := l
                end;
        end;
    end;
```

```

writeln(d[k - 1][n]);
i := k - 1;
j := n;
asize := 0;
while pr[i, j] <> 0 do begin
  inc(asize);
  ans[asize] := pr[i, j];
  j := pr[i, j];
  dec(i)
end;
for i := asize downto 1 do
  write(ans[i], ' ');
writeln()
end.

```

## Python

```

n, k = map(int, input().split())
a = list(map(int, input().split()))

rmq = [[-1] * n for i in range(n)]
for i in range(n):
    rmq[i][i] = a[i]

for i in range(n):
    for j in range(i + 1, n):
        rmq[i][j] = min(rmq[i][j - 1], a[j])

d = [[-1] * (k + 1) for i in range(n + 1)]
f = [[-1] * (k + 1) for i in range(n + 1)]
d[0][0] = 0
for i in range(n):
    for j in range(k):
        if d[i][j] != -1:
            for l in range(i + 1, n + 1):
                v = d[i][j] + rmq[i][l - 1]
                if v > d[l][j + 1]:
                    d[l][j + 1] = v
                    f[l][j + 1] = i

print(d[n][k])
a = []
c = n
p = k
while c > 0:
    c = f[c][p]
    p -= 1
    if c != 0:
        a.append(c)
a.reverse()
print(" ".join([str(x) for x in a]))

```