

## Разбор задач олимпиады «Полуфинал Всероссийской командной олимпиады школьников по программированию»

### Задача А. Семья

Составим и решим систему уравнений:

$$\begin{cases} x + N = y, \\ x \cdot M = y; \end{cases} \quad x \cdot M = x + N, \quad x = \frac{N}{M-1}, \quad y = \frac{M \cdot N}{M-1}.$$

### Решение на языке C++

```
#include <iostream>
#include <cstdio>

using namespace std;

int main(){
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    int n, m, t;
    cin >> t;
    while(t--){
        cin >> n >> m;
        cout << m*n/(m-1) << " " << n/(m-1) << endl;
    }
    return 0;
}
```

### Решение на языке Паскаль

```
var n, m, t : Longint;
begin
    assign(input, 'input.txt'); reset(input);
    assign(output, 'output.txt'); rewrite(output);
    Read(t);
    while t>0 do begin
        dec(t);
        Read(n, m);
        WriteLn(n*m div (m-1), ' ', n div (m-1));
    end;
end.
```

### Задача В. Паркет - 2

Из школьного курса математики известно, что сумма углов в  $n$ -угольнике равна  $180 \cdot (n-2)$  градусов, тогда один угол в правильном  $n$ -угольнике равен  $180 \cdot (n-2)/n$  градусов. Если замостить всю плоскость, то будут вершины, в которых сходятся углы  $n$ -угольников. А это значит, что угол в 360 градусов будет состоять из нескольких углов  $n$ -угольника. Другими словами 360 должно делиться на  $180 \cdot (n-2)/n$ , после упрощения получаем, что  $2 \cdot n$  должно делиться на  $n-2$ .

Либо можно было заметить, что это выполняется лишь для  $n=3$  ,  $n=4$  и  $n=6$  .

### **Решение на языке C++**

```
#include <iostream>
#include <stdio.h>

using namespace std;

int main(){
    freopen("input.txt","r",stdin);
    freopen("output.txt","w",stdout);

    int t, n;
    cin >> t;
    while(t--){
        cin >> n;
        if(2*n%(n-2) == 0) cout << "YES\n";
        else cout << "NO\n";
    }
    return 0;
}
```

### **Решение на языке Паскаль**

```
var n, t : Longint;
begin
    assign(input, 'input.txt'); reset(input);
    assign(output, 'output.txt'); rewrite(output);
    Read(t);
    while t>0 do begin
        dec(t);
        Read(n);
        if 2*n mod (n-2) = 0 then WriteLn('YES')
        else WriteLn('NO');
    end;
end.
```

## **Задача С. Обмен**

В условии задачи описана известная комбинаторная формула – количество беспорядков. Это можно посчитать различными формулами, мы приведём решение с помощью формулы:

$$!n = !(n-1) \cdot n + (-1)^n .$$

Не забываем использовать 64-битные типы данных в операциях умножения, чтобы избежать переполнения типов.

### **Решение на языке C++**

```
#include <iostream>
#include <cstdint>

using namespace std;

int main(){
    freopen("input.txt","r",stdin);
    freopen("output.txt","w",stdout);
    int t;
    cin >> t;
    while(t--){
        long long n, r=1;
        cin >> n;
```

```
        for(int i=3; i<=n; ++i)
            r = (r*i+1-2*(i&1)) % 1000000009;
        cout << r << endl;
    }
    return 0;
}
```

### **Решение на языке Паскаль**

```
var
    t, n, r : Int64;
    i : Longint;
begin
    assign(input, 'input.txt'); reset(input);
    assign(output, 'output.txt'); rewrite(output);
    Read(t);
    while t>0 do begin
        dec(t);
        Read(n);
        r := 1;
        for i:=3 to n do
            r := (r*i+1-2*(i and 1)) mod 1000000009;
        WriteLn(r);
    end;
end.
```

### **Задача D. Апельсины - 2**

Для начала узнаем количество сухого вещества в апельсинах, оно останется неизменным. А вот количество воды в апельсинах будет меняться.

Количество сухого вещества  $x = n \cdot (100 - f) \%$ .

Тогда итоговую влажность можно вычислить по обратной формуле.

### **Решение на языке C++**

```
#include <iostream>
#include <cstdio>

#define eps (1e-9)

using namespace std;

int main(){
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);

    int n, f, m, a, t;
    cin>>t;
    while(t--){
        cin>>n>>f>>m;
        double x = (100-f)/100.*n;

        for(int i=0;i<m;++i){
            cin>>a;
            n += a;
        }

        cout<<n<<" " <<int((n-x)/n*100+0.5+eps)<<endl;
    }

    return 0;
}
```

## **Решение на языке Паскаль**

```
uses math;
var
  t, n, f, m, a, i : Longint;
  x : Double;

begin
  assign(input, 'input.txt'); reset(input);
  assign(output, 'output.txt'); rewrite(output);

  Read(t);
  while t>0 do begin
    dec(t);
    Read(n, f, m);
    x := (100.0-f)/100*n;

    for i:=1 to m do begin
      Read(a);
      inc(n, a);
    end;

    WriteLn(n, ' ', floor((n-x)/n*100+0.500000001));
  end;
end.
```

## **Задача Е. Ученики**

В этой задаче нет ничего сложного. Достаточно отсортировать известным алгоритмом данные. На языке C++ всё решение сводится к написанию компаратора для стандартной функции. На Паскале придётся реализовать алгоритм гномьей сортировки.

## **Решение на языке C++**

```
#include <iostream>
#include <cstdio>
#include <string>
#include <vector>
#include <algorithm>

using namespace std;

struct w{
  string f, i, d;
  int c;
  char b;
};

bool cmp(w a, w b){
  return a.c < b.c || (a.c == b.c && (a.b < b.b || (a.b == b.b && a.f < b.f)));
}

void solve(){
  int n;
  cin>>n;
  vector<w> v;
  w t;
  for(int i=0;i<n;++i){
    cin>>t.f>>t.i>>t.c>>t.b>>t.d;
    v.push_back(t);
  }
  sort(v.begin(), v.end(), cmp);
}
```

```
for(int i=0;i<n;++i)
    cout<<v[i].c<<v[i].b<<" "<<v[i].f<<" "<<v[i].i<<" "<<v[i].d<<endl;
}

int main(){
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);

    int t;
    cin>>t;
    for(int i=1;i<=t;++i){
        cout<<"Case "<<i<<"\n";
        solve();
    }

    return 0;
}
```

### **Решение на языке Паскаль**

```
var
    ii, num, n, i, t : Longint;
    f, e, d, c : array [1..100] of String;
    m : array [1..100] of Longint;

function cmp(a, b : Longint) : Boolean;
begin
    if length(c[a]) > length(c[b]) then cmp := true
    else if length(c[a]) < length(c[b]) then cmp := false
    else if c[a] > c[b] then cmp := true
    else if c[a] < c[b] then cmp := false
    else cmp := (f[a] > f[b]);
end;

begin
    assign(input, 'input.txt'); reset(input);
    assign(output, 'output.txt'); rewrite(output);

    ReadLn(num);
    for ii:=1 to num do begin
        WriteLn('Case ',ii,':');

        ReadLn(n);

        for i:=1 to n do begin
            ReadLn(f[i]);
            ReadLn(e[i]);
            ReadLn(c[i]);
            ReadLn(d[i]);
            m[i] := i;
        end;

        i:=1;
        while i<n do
            if cmp(m[i], m[i+1]) then begin
                t := m[i];
                m[i] := m[i+1];
                m[i+1] := t;
                if i>1 then dec(i);
            end else inc(i);

        for i:=1 to n do
            WriteLn(c[m[i]],' ',f[m[i]],' ',e[m[i]],' ',d[m[i]]);
```

```
end;  
end.
```

### Задача F. Функция - 3

При дорешивании дома, либо во время он-лайн соревнования можно построить график и понять, что всё просто. Но давайте рассмотрим как же можно было решать эту задачу на очном соревновании.

В первую очередь сделаем замену  $\sqrt{x}=a$ . Тогда получаем уравнение  $a^4 - a = C$ .

Для  $a > 1$  функция  $a^4$  растёт быстрее, чем  $a$ , поэтому функция  $a^4 - a$  будет возрастающей и будет принимать значения от 0 до бесконечности.

Осталось проанализировать отрезок  $[0, 1]$ . Найдём экстремумы нашей функции. Для этого производную от неё приравняем к нулю и решим уравнение:

$$4a^3 - 1 = 0, \quad a = \sqrt[3]{0.25}, \quad a = 0.629960525, \quad x = 0.396850263.$$

Это будет точка минимума, обозначим её как  $x_m$ . Значит можно разделить отрезок  $[0, 1]$  на две части  $[0, x_m]$  и  $(x_m, 1]$ , на каждой из которых функция будет монотонна. Теперь можно применить бинпоиск.

#### Решение на языке C++

```
#include <iostream>  
#include <cstdio>  
#include <cmath>  
  
#define eps (1e-9)  
  
using namespace std;  
  
double f(double x){  
    return x*x-sqrt(x);  
}  
  
double bin_search(double l, double r, double c){  
    double m;  
    int t=0;  
    while(fabs(l-r)>eps && t<200){  
        ++t;  
        m = (l+r)*.5;  
        if(f(m)>c) l=m;  
        else r=m;  
    }  
    return r;  
}  
  
int main(){  
    freopen("input.txt", "r", stdin);  
    freopen("output.txt", "w", stdout);  
  
    int t;  
    cin>>t;  
    while(t--){  
        double c;  
        cin>>c;  
  
        double xm = 0.396850263;  
  
        if(c<eps && c>f(xm)+eps) printf("%.6lf ", bin_search(0., xm, c));
```

```
        if(c>f(xm)-eps) printf("%.6lf", bin_search(1e6, xm, c));
        if(c<f(xm)-eps) printf("No solution");
        printf("\n");
    }
    return 0;
}
```

### **Решение на языке Паскаль**

```
const eps : Double = 0.000000001;

function f(x : Double) : Double;
begin
    f := x*x - sqrt(x);
end;

function bin_search(l, r, c : Double) : Double;
var
    m : Double;
    t : Longint;
begin
    t := 0;
    while (abs(l-r)>eps) and (t<200) do begin
        inc(t);
        m := (l+r) * 0.5;
        if f(m)>c then l := m
        else r := m;
    end;
    bin_search := r;
end;

var
    c, xm : Double;
    t : Longint;
begin
    assign(input, 'input.txt'); reset(input);
    assign(output, 'output.txt'); rewrite(output);

    Read(t);
    while t>0 do begin
        dec(t);
        Read(c);
        xm := 0.396850263;

        if (c<eps) and (c>f(xm)+eps) then Write(bin_search(0.0, xm, c):0:6, ' ');
        if c>f(xm)-eps then Write(bin_search(1e6, xm, c):0:6);
        if c<f(xm)-eps then Write('No solution');
        WriteLn;
    end;
end.
```

### **Задача G. Игра - 4**

В данной задаче необходимо реализовать алгоритм поиска в ширину. Единственным отличием является поле, на котором разворачиваются действия – это соты. Поэтому ходить можно в шести направлениях. Причём в квадратном представлении эти направления отличаются для чётных и нечётных строк. Поэтому удобно создать два массива –  $dx$  и  $dy$  – в которые и записать все возможные перемещения. Для того, чтобы поиск был линеен относительно размеров поля, необходимо использовать

очередь, в которую складываются координаты всех достигнутых клеток.

### **Решение на языке C++**

```
#include <iostream>
#include <cstdio>
#include <queue>

using namespace std;

void solve(){
    int n, m, y, x;
    cin >> n >> m >> y >> x;
    int mm[1009][1009];
    char s[1009];
    for(int i=0;i<n;++i){
        cin>>s;
        for(int j=0;j<m;++j)
            mm[i][j] = s[j]-'0'-1;
    }
    queue<pair<int,int> > q;
    q.push(make_pair(y-1, x-1));

    int dx[2][6] = {{-1, 0, -1, 1, -1, 0},{ 0, 1, -1, 1, 0, 1}};
    int dy[2][6] = {{-1, -1, 0, 0, 1, 1},{-1, -1, 0, 0, 1, 1}};

    while(!q.empty()){
        y = q.front().first;
        x = q.front().second;
        if(!x || !y || x==m-1 || y==n-1){
            cout << mm[y][x]+1;
            return ;
        }
        q.pop();
        for(int i=0;i<6;++i)
            if(!mm[y+dy[y&1][i]][x+dx[y&1][i]]){
                mm[y+dy[y&1][i]][x+dx[y&1][i]] = mm[y][x]+1;
                q.push(make_pair(y+dy[y&1][i], x+dx[y&1][i]));
            }
    }
    cout << "No solution";
}

int main(){
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);

    int t;
    cin>>t;
    while(t--){
        solve();
        cout<<endl;
    }

    return 0;
}
```

### **Решение на языке Паскаль**

```
const
    dx : array [0..1,0..5] of Longint=(( 0, 1, -1, 1, 0, 1),(-1, 0, -1, 1, -1, 0));
    dy : array [0..1,0..5] of Longint=(-1, -1, 0, 0, 1, 1),(-1, -1, 0, 0, 1, 1));

var
```

```
n, m, i, j, y, x, lq : Longint;
mm : array [1..1000,1..1000] of Longint;
qx, qy : array [1..1000000] of Longint;
s : AnsiString;

procedure solve;
begin
  ReadLn(n, m);
  ReadLn(y, x);
  for i:=1 to n do begin
    ReadLn(s);
    for j:=1 to m do
      mm[i,j] := ord(s[j])-49;
    end;

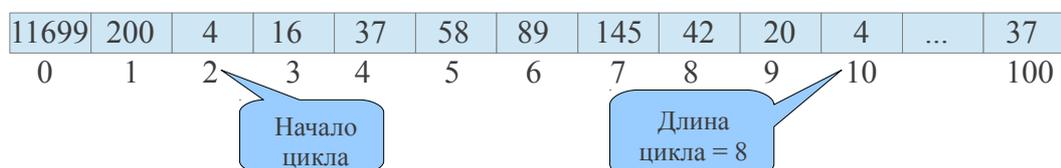
    qx[1] := x;
    qy[1] := y;
    lq := 1;
    j := 1;
    while j<=lq do begin
      y := qy[j];
      x := qx[j];
      if (x=1) or (y=1) or (x=m) or (y=n) then begin
        Write(mm[y,x]+1);
        exit;
      end;
      for i:=0 to 5 do
        if mm[y+dy[y and 1, i],x+dx[y and 1, i]] = 0 then begin
          mm[y+dy[y and 1, i],x+dx[y and 1, i]] := mm[y, x] + 1;
          inc(lq);
          qy[lq] := y+dy[y and 1, i];
          qx[lq] := x+dx[y and 1, i];
        end;
      inc(j);
    end;

    Write('No solution');
  end;

var t : Longint;
begin
  assign(input, 'input.txt'); reset(input);
  assign(output, 'output.txt'); rewrite(output);
  ReadLn(t);
  while t>0 do begin
    dec(t);
    solve;
    WriteLn;
  end;
end.
```

## Задача N. Последовательность - 6

Заметим, что все получающиеся числа не будут превышать  $9 \cdot 3 = 729$ . По принципу Дирихле какое-нибудь число при большом количестве итераций должно повториться, а это приведёт к закливанию последовательности. В этом случае можно сразу получить ответ по формуле, которую несложно получить из рисунка:



### **Решение на языке C++**

```
#include <iostream>
#include <cstdio>
#include <vector>

using namespace std;

int f(int a){
    int s=0;
    while(a){
        s += (a%10)*(a%10);
        a /= 10;
    }
    return s;
}

void solve(){
    int a, n, i, j;
    cin >> a >> n;
    vector<int> v;
    v.push_back(a);
    for(i=1; i<=n; ++i){
        a = f(a);
        for(j=0; j<v.size(); ++j)
            if(v[j] == a){
                cout << v[j+(n-j)%(i-j)] << endl;
                return ;
            }
        v.push_back(a);
    }
    cout << v.back() << endl;
}

int main(){
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);

    int t;
    cin >> t;
    while(t-->0) solve();

    return 0;
}
```

### **Решение на языке Паскаль**

```
function f(a : Longint) : Longint;
var
    s : Longint;
begin
    s := 0;
    while a>0 do begin
        inc(s, sqr(a mod 10));
        a := a div 10;
    end;
    f := s;
end;

procedure solve;
var
    n, a, i, j : Longint;
```

```
v : array [0..1000] of Longint;
begin
  Read(a, n);
  v[0] := a;
  for i:=1 to n do begin
    a := f(a);
    for j:=0 to i-1 do
      if v[j] = a then begin
        Write(v[j+(n-j) mod (i-j)]);
        exit;
      end;
    v[i] := a;
  end;
  Write(v[n]);
end;

var t : Longint;
begin
  assign(input, 'input.txt'); reset(input);
  assign(output, 'output.txt'); rewrite(output);
  Read(t);
  while t>0 do begin
    dec(t);
    solve;
    WriteLn;
  end;
end.
```

## Задача I. БИТЫ

Проверку того является k-ый бит единичкой или нет можно сделать с помощью битовой операции & (and). А именно выражением  $n \& (1 \ll (k-1))$ .

Зная это, достаточно циклом проверить каждый из битов в ведённом числе.

Не забываем использовать 64-битные типы данных.

### **Решение на языке C++**

```
#include <iostream>
#include <cstdio>
#include <cmath>

using namespace std;

int main(){
  freopen("input.txt", "r", stdin);
  freopen("output.txt", "w", stdout);
  int t;
  cin >> t;
  while(t--){
    long long n, mn=(1LL<<62), mx=0;
    cin >> n;
    for(int i=0; i<63; ++i)
      if(n&(1LL<<i)){
        mn = min(mn, 1LL<<i);
        mx = max(mx, 1LL<<i);
      }
    cout << mn << " " << mx << endl;
  }
  return 0;
}
```

## **Решение на языке Паскаль**

```
uses math;
var
  t, n, mn, mx, one : Int64;
  i : Longint;
begin
  assign(input, 'input.txt'); reset(input);
  assign(output, 'output.txt'); rewrite(output);
  Read(t);
  while t>0 do begin
    dec(t);
    Read(n);
    one := 1;
    mn := one shl 62;
    mx := 0;
    for i:=0 to 62 do
      if (n and (one shl i))>0 then begin
        mn := min(mn, one shl i);
        mx := max(mx, one shl i);
      end;
    WriteLn(mn, ' ', mx);
  end;
end.
```