

Разбор задач муниципального этапа олимпиады по информатике

1. Илья Муромец (все классы)

Тема: разбор случаев

Сложность: простая

Возможны три случая:

1) камень находится между Киевом и селом (в этом случае $B-A=Y-X$, расстояние до Киева равно $A-X$);

2) камень находится между селом и рекой (в этом случае $B-A=X+Y$, расстояние до Киева равно $A+X$);

3) камень находится между рекой и Черниговым (в этом случае $B-A=X-Y$, расстояние до Киева равно $A+X$).

Пример реализации:

```
var a,b,x,y:integer;
begin
  read(a,b,x,y);
  if (b-a=x+y) or (b-a=x-y) then
    writeln(a+x)
  else
    writeln(a-x);
end.
```

2. Алгоритм (7-9 классы)

Тема: реализация программы по схеме алгоритма

Сложность: простая

Пример реализации:

```
var a,b,c,k,i:integer;
begin
  read(a,b);
  k:=0;
  for i:=1 to 1000 do
    begin
      c:=i*i*i;
      if (a<=c) and (c<=b) then inc(k);
    end;
  writeln(k);
end.
```

2. Сотовая связь (10-11 классы)

Тема: поиск минимума

Сложность: простая

Пример реализации:

```
var kol,bx,by:array [1..10] of integer;
    n,m,ax,ay,i,j,r,minr,minb:integer;
begin
  read(n);
  for i:=1 to n do
    read(bx[i],by[i]);
  read(m);
  for i:=1 to m do
```

```

begin
  read(ax,ay);
  minr:=1000000000;
  minb:=0;
  for j:=1 to n do
  begin
    r:=sqr(ax-bx[j])+sqr(ay-by[j]);
    if r<minr then
    begin
      minr:=r;
      minb:=j;
    end;
  end;
  inc(kol[minb]);
end;
for i:=1 to n do
  writeln(kol[i]);
end.

```

3. Автомобиль-робот (7-8 классы)

Тема: исполнитель

Сложность: простая

Пример исполнителя для правого рисунка: $>^3<^4>^{\wedge}$

Универсальная программа выглядит так: $(?^{|?>^{|<^})9$

3. Похожесть последовательностей (9 класс)

Тема: поиск минимума

Сложность: простая

Пример реализации:

```

var a,b:array[1..100] of integer;
    n,i,c,d,mind:integer;
begin
  read(n);
  for i:=1 to n do
    read(a[i]);
  for i:=1 to n do
    read(b[i]);
  mind:=1000000000;
  for c:=-1000 to 1000 do
  begin
    d:=0;
    for i:=1 to n do
      d:=d+abs(a[i]+c-b[i]);
    if d<mind then mind:=d;
  end;
  writeln(mind);
end.

```

3. Количество итераций (10-11 класс)

Тема: использование стека или рекурсия

Сложность: ниже среднего

Пример реализации:

```
uses sysutils;
var st:array[0..20]of int64;
    t,n,i:integer;
    s:string;
begin
    t:=0;
    readln(n);
    for i:=1 to 2*n do
    begin
        readln(s);
        if copy(s,1,3)='for' then
        begin
            inc(t);
            st[t]:=StrToInt(Copy(s,5,Length(s)-4));
                { для накопления произведения }
            inc(t);
            st[t]:=0; { для накопления суммы }
        end
        else {end}
        begin
            if st[t]=0 then
                st[t]:=1;
            st[t-1]:=st[t-1]*st[t];
            dec(t);
            st[t-1]:=st[t-1]+st[t];
            dec(t);
        end;
    end;
    writeln(st[0]);
end.
```

3. Забор (все классы)

Тема: математика, структуры данных (10-11 класс) или сортировка (7-9 класс и частичное решение 10-11 класс)

Сложность: средняя

Минимальное количество распилов для доски получится, если доска будет распилена на части равной длины. Будем поддерживать упорядоченный по длине частей список номеров досок и количества частей из этой доски. Пока разница между минимальной и максимальной длиной превышает D , мы должны увеличивать количество частей у доски, части которой имеют наибольшую длину. После изменения длины частей, доску перемещаем в упорядоченном списке. Процесс гарантированно завершится, хотя бы когда длина всех частей станет меньше D . Для распиливания досок на части меньше D , потребуется не более $N \cdot 100000 / D$ распилов. Для ускорения работы программы для больших N можно использовать специальную структуру данных (кучу или дерево максимумов, см. разбор задачи 670. Покраска забора). Общее время работы $O(N \cdot 100000 / D \cdot \log N)$.

Пример частичного решения за $O(N^2 \cdot 100000/D)$:

```
type doska=
  record
    plen:extended;
    len:integer;
    kol:integer;
  end;
var dat:array[1..1000] of doska;
    tmp:doska;
    minlen:extended;
    kol,n,i,j,D:integer;
begin
  read(n,D);
  minlen:=1000000;
  for i:=1 to n do
  begin
    read(dat[i].len);
    dat[i].kol:=1;
    dat[i].plen:=dat[i].len;
    if dat[i].plen<minlen then minlen:=dat[i].plen;
    for j:=i-1 downto 1 do
      if dat[j].plen<dat[j+1].plen then
        begin
          tmp:=dat[j];
          dat[j]:=dat[j+1];
          dat[j+1]:=tmp;
        end
      else
        break;
    end;
  end;
  kol:=0;
  while dat[1].plen-minlen>D do
  begin
    inc(dat[1].kol);
    inc(kol);
    dat[1].plen:=dat[1].len/dat[1].kol;
    if dat[1].plen<minlen then minlen:=dat[1].plen;
    for j:=1 to n-1 do
      if dat[j].plen<dat[j+1].plen then
        begin
          tmp:=dat[j];
          dat[j]:=dat[j+1];
          dat[j+1]:=tmp;
        end
      else
        break;
    end;
  end;
  writeln(kol);
end.
```