OLIMPIADA NAȚIONALĂ DE INFORMATICĂ
ETAPA JUDEȚEANĂ
CLASA A – IX –A

Solutie_REACT_QS_PAS_09

```
{$R-,S-}
Program Reactivi;
{NU l-am dat pe asta ca sursa deoarece intr-a IX-a INCA nu se face
QuickSort}
Type Frigider = Record
                   min, max: ShortInt
                End;
Var f            : Array[1..10000] Of Frigider;
    r            : Array[1..10000] Of Frigider;
    Fisier       : Text;
    Cate, i, j, N: Integer;
    min, max     : ShortInt;

Function Maxim(x, y: ShortInt): ShortInt;
Begin
  If x>y Then
    Maxim := x
  Else
    Maxim := y
End;

Function Minim(x, y: ShortInt): ShortInt;
Begin
  If x<y Then
    Minim := x
  Else
    Minim := y
End;

Function Cauta(min, max: ShortInt): Integer;
Var i: Integer;
Begin
  Cauta := -1;
  For i := 1 To Cate Do
    Begin
      If (f[i].max>=min) And (f[i].max<=max) Then
        Begin Cauta := i; Break End
      Else
        If (f[i].min>=min) And (f[i].min<=max) Then
          Begin Cauta := i; Break End
        Else
          If (f[i].min<=min) And (f[i].max>=max) Then
            Begin Cauta := i; Break End
          Else
            If (f[i].max<min) Or (f[i].min>max) Then
              Cauta := -1
    End;
```

```
End;

Procedure Intersectie(j: Integer; min, max: ShortInt);
Begin
  f[j].min := Maxim(min, f[j].min);
  f[j].max := Minim(max, f[j].max)
End;

procedure QuickSortCresc(Lo, Hi: Integer);

  procedure Sort(Stanga, Dreapta: Integer);
  var i, j, x: integer;
      y: Frigider;
  begin
    i := Stanga; j := Dreapta; x := r[(Stanga+Dreapta) DIV 2].min;
    repeat
      while r[i].min < x do i := i + 1;
      while x < r[j].min do j := j - 1;
      if i <= j then
      begin
        y := r[i]; r[i] := r[j]; r[j] := y;
        i := i + 1; j := j - 1;
      end;
    until i > j;
    if Stanga < j then Sort(Stanga, j);
    if i < Dreapta then Sort(i, Dreapta);
  end;

begin {QuickSort};
  Sort(Lo,Hi);
end;

procedure QuickSortDescresc(Lo, Hi: Integer);

  procedure Sort(Stanga, Dreapta: Integer);
  var i, j, x: integer;
      y: Frigider;
  begin
    i := Stanga; j := Dreapta; x := r[(Stanga+Dreapta) DIV 2].max;
    repeat
      while r[i].max > x do i := i + 1;
      while x > r[j].max do j := j - 1;
      if i <= j then
      begin
        y := r[i]; r[i] := r[j]; r[j] := y;
        i := i + 1; j := j - 1;
      end;
    until i > j;
    if Stanga < j then Sort(Stanga, j);
    if i < Dreapta then Sort(i, Dreapta);
  end;

begin {QuickSort};
  Sort(Lo,Hi);
end;

Begin
```

```
  Assign(Fisier, 'reactivi.in'); Reset(Fisier);
  ReadLn(Fisier, N);
  For i := 1 To N Do
    ReadLn(Fisier, r[i].min, r[i].max);
  Close(Fisier);

  QuickSortCresc(1, N);

  i := 1; j := i+1;
  While j<=N Do
    Begin
      While (r[j].min=r[i].min) And (j<=N) Do Inc(j);
      QuickSortDescresc(i, j-1);
      i := j; j := i+1
    End;

  f[1].min := r[1].min; f[1].max := r[1].max; Cate := 1;
  For i := 2 To N Do
    Begin
      min := r[i].min; max := r[i].max;
      j := Cauta(min, max);
      If j>0 Then
        Intersectie(j, min, max)
      Else
        Begin
          Inc(Cate);
          f[Cate].min := min;
          f[Cate].max := max
        End
    End;
  Assign(Fisier, 'reactivi.out'); ReWrite(Fisier);
  WriteLn(Fisier, Cate);
  Close(Fisier)
End.
```

Soluite_EXP_PAS_9

```
Program Exp;

type vector = array[1..6000] of integer;
var x, p, y, f: vector;
    m, n, dp, i, k, j: integer;
    sw: boolean;
    ff: text;

procedure cit;
var i: integer;
    f: text;
begin
  assign(f, 'exp.in'); reset(f);
  readln(f, m);
  readln(f, n);
  for i:=1 to n do
    read(f, x[i]);
  close(f);
end;

function prim(k: integer): boolean;
var i: integer;
begin
  prim := true;
  for i := 2 to trunc(sqrt(k)) do
    if k mod i=0 then
      begin
        prim := false;
        exit;
      end;
end;

procedure prime;
var i, j: integer;
begin
  p[1] := 2;
  p[2] := 3;
  dp := 2;
  i := 5;
  while i<30000 do
    begin
      if prim(i) then
        begin
          inc(dp);
          p[dp] := i;
        end;
      i := i+2;
    end;
end;

begin
  cit;
  prime;
  for i := 1 to dp do
    f[i] := 0;
  for i := 1 to n do
    begin
      j := 1;
      while x[i]<>1 do
        begin
          while x[i] mod p[j]=0 do
            begin
              inc(f[j]);
              x[i] := x[i] div p[j];
            end;
          inc(j);
        end;
    end;
  sw := true;
  for i := 1 to dp do
    if f[i] mod m<>0 then
      begin
        sw := false;
        break;
      end;

  assign(ff, 'exp.out'); rewrite(ff);
  if not sw then
    writeln(ff, 0)
  else
    begin
      writeln(ff, 1);
      for i := 1 to dp do
        if f[i]>0 then
          writeln(ff, p[i], ' ', f[i] div m);
    end;
  close(ff);
end.
```