

## SOAL

### Soal 1. Selular Automata (LIFE)

Permainan ini adalah permainan yang mensimulasikan kehidupan. Tempat permainannya berupa persegi dengan panjang sisi 100. Setiap sel dinyatakan dalam koordinat integer di persegi itu dan dapat bernilai “mati” atau ‘hidup’. Keadaan awal dari papan itu diberikan di input. Sebuah sel berhubungan dengan 8 sel di sekitarnya, kecuali untuk sel – sel di pinggir papan.

Cara bermainnya adalah melakukan langkah berikut

1. Sebuah sel mati, yang dikelilingi tepat oleh 3 sel hidup, akan menjadi sel hidup
2. Sebuah sel hidup yang memiliki 2 atau 3 orang kawan, akan tetap hidup
3. Selain kasus di atas, sel itu akan mati

Catatan : penggantian keadaan sebuah sel harus dilakukan serentak untuk setiap langkahnya.

Bila langkah ini diulang – ulang, akan terjadi pola yang menarik yang berbeda – beda untuk setiap keadaan awal. Anda diminta membuat program yang diberikan input dan jumlah langkahnya, memberikan keadaan akhir dari papan itu.

### INPUT

Baris pertama dari input berisi 2 bilangan,  $N$  ( $1 \leq N \leq 2000$ ) dan  $M$ , di mana  $N$  adalah jumlah langkah yang dilakukan, dan  $M$  adalah jumlah sel hidup awal.  $M$  baris berikutnya berisi dua bilangan  $R$  dan  $C$ , di mana  $R$  adalah nomor baris, dan  $C$  adalah nomor kolom.  $1 \leq R, C \leq 100$

### OUTPUT

Berisi beberapa baris yang masing – masing baris terdiri dari 2 angka,  $R$  dan  $C$ , yang mencetak semua posisi sel hidup. Output harus diberikan dalam keadaan terurut menurut baris lalu menurut kolom.

#### Contoh Input 1

		10 45
		10 46
200 10		10 47
10 40	■	10 48
10 41		10 49
10 42		
10 43		
10 44		

#### Contoh output 1

8 41

**OLIMPIAD NASIONAL INFORMATIKA 2003**  
**TIM OLIMPIADE KOMPUTER INDONESIA**

---

8 48	10 49
9 40	10 50
9 41	11 40
9 48	11 41
9 49	11 48
10 39	11 49
10 40	12 41
10 41	12 48
10 48	



<b>Contoh input 2</b>	16 18
1100 5	16 19
10 40	17 18
10 39	17 19
11 40	20 11
9 40	21 6
9 41	21 10



<b>Contoh output 2</b>	21 12
1 29	22 5
1 30	22 7
2 29	22 10
2 30	22 11
8 31	23 6
8 32	27 3
9 8	28 2
9 9	28 4
9 31	29 2
9 32	29 4
10 8	30 3
10 9	99 79
10 35	99 80
11 35	100 79
12 35	100 80



## **SOLUSI**

Solusi dari soal ini adalah dengan membuat dua buah matriks 2D berukuran 100 X 100, yang masing – masing isi dari setiap matriks mewakili state dari suatu sel. Misalnya bila sel itu hidup, maka bernilai TRUE, dan bila mati bernilai FALSE.

Lalu dilakukan looping sebanyak yang diminta, untuk melakukan perubahan – perubahan state sesuai dengan aturan yang diminta. Cara melakukannya adalah memasukan keadaan dari langkah sebelumnya di matriks pertama, lalu mengisi keadaan matriks kedua berdasarkan peraturan dan data pada matriks pertama, setelah matriks kedua terisi, maka isi matriks pertama dicopy ke matriks kedua.

Setelah melakukan iterasi itu sebanyak yang diminta, berikutnya kita melakukan tracing dari matriks kedua. Bila nilainya TRUE, maka cetak nomor baris dan kolomnya.

## **ALGORITMA**

Baca input, masukan ke matriks1

-- Simulasikan langkah - langkah

For I := 1 to n do

Begin

    Matriks2 := syarat – syarat dari data matriks 1

    Matriks1 := matriks2;

End;

-- Cetak output

For I := 1 to 100 do

Begin

    For j := 1 to 100 do if matriks2[I,j] = TRUE then writeln(I, ' ',j);

End;

## **SOURCE CODE**

```
{LIFE – Widagdo Setiawan – 2/9/2002}
```

```
const Max = 100;
```

```
var i,j,k,n,m,r,c,x : longint;
```

```
    map1,map2 : array [0..max+1,0..max+1] of boolean;
```

```
begin
```

```
    assign(input,'LIFE.IN');reset(input);
```

```
    assign(output,'LIFE.OUT');rewrite(output);
```

```
    readln(n,m);
```

```
    for i := 1 to m do
```

```
        begin
```

```
            readln(r,c);
```

```
            map1[r,c] := true;
```

```
end;
for k := 1 to n do
begin
  for i := 1 to max do
  begin
    for j := 1 to max do
    begin
      x := 0;
      if map1[i-1,j-1] then inc(x);
      if map1[i-1,j] then inc(x);
      if map1[i-1,j+1] then inc(x);
      if map1[i,j-1] then inc(x);
      if map1[i,j+1] then inc(x);
      if map1[i+1,j-1] then inc(x);
      if map1[i+1,j] then inc(x);
      if map1[i+1,j+1] then inc(x);
      if (x = 3) or (map1[i,j] and (x=2)) then
        map2[i,j] := true else
        map2[i,j] := false;
    end;
  end;
  map1 := map2;
end;
for i := 1 to 100 do
begin
  for j := 1 to 100 do
  begin
    if map1[i,j] then writeln(i, ' ', j);
  end;
end;
close(OUTPUT);
end.
```

## **LATAR BELAKANG**

Permainan ini diciptakan oleh John Conway, seorang professor matematika di universitas Princeton.

<http://www.math.com/students/wonders/life/life.html>

## SOAL

### Soal 2. Pengelompokan surat (ACSL)

Untuk mempermudah pengiriman suatu pelayanan pos, suatu instansi harus membawa semua suratnya ke kantor pos dalam keadaan dikelompokkan menurut aturan tertentu dan tidak boleh ada kelompok yang berisi lebih dari 125 surat

Pengelompokan surat itu dilakukan dengan menggunakan aturan dan **urutan** sebagai berikut:

1. Kelompokkan 10 atau lebih surat yang mempunyai 5 digit kode pos yang sama dan tempelkan stiker “D” pada surat teratas.
2. Kelompokkan 10 atau lebih surat yang mempunyai 3 digit kode pos terdepan yang sama dan tempelkan stiker “T” pada surat teratas.
3. Kelompokkan 10 atau lebih surat ke Pusat Distribusi Daerah (PDD) yang sama dan tempelkan stiker “A” pada bagian atas.

PDD adalah suatu cara untuk mengelompokkan beberapa kode pos yang berbeda. Untuk mengetahui suatu surat dikirim ke PDD mana, cukup dengan membaca 3 digit terdepan dari kode posnya. Berikut ini adalah tabel yang memperlihatkan PDD yang digunakan untuk kode pos yang 3 digit pertamanya seperti yang ditunjukkan dalam tabel itu. Sebagai contoh, PDD “028” digunakan untuk kode pos yang diawali dengan 020, 023, 024, 025, ..., 029.

PDD 105	3 digit Kode POS terdepan 004, 105-109
117	005, 115, 117-119
106	006-009
110	010-017
021	018, 019, 021, 022, 055
028	020, 023-029

Bila 3 digit terdepan dari suatu kode pos tidak masuk dalam PDD manapun, maka surat dengan kode pos itu tidak dapat dikelompokkan dengan cara ini.

4. Kelompokkan surat yang lain dan tempelkan stiker “M”

2 buah kelompok dikatakan memiliki karakteristik sama apabila mempunyai stiker sama **DAN** memenuhi

- bila stiker D, maka memiliki kode pos yang sama
- bila stiker T, maka memiliki 3 digit terdepan kode pos yang sama
- bila stiker A, maka memiliki PDD yang sama

Diperbolehkan ada lebih dari 1 kelompok dengan karakteristik sama. Tetapi hanya 1 kelompok dari suatu karakteristik yang **boleh** beranggotakan kurang dari 125 surat, sehingga kelompok lain dengan karakteristik itu, mempunyai 125 anggota.

Misal ada 507 surat dengan kode pos 11603 maka surat itu dikelompokkan dalam 4 kelompok dengan aturan A yang masing – masing berisi 125 anggota, dan 7 surat sisanya dikelompokkan dengan aturan lain. Bila ada 515 surat dengan kode pos 11603, maka surat itu

**OLIMPIADENASIONAL INFORMATIKA 2003**  
**TIM OLIMPIADE KOMPUTER INDONESIA**

---

dikelompokkan dalam 5 kelompok dengan aturan A yang masing – masing berisi 125, 125, 125, 125, 15.

**INPUT**

Baris pertama berisi integer positif  $N$  ( $1 \leq N \leq 15000$ ) yang menyatakan banyaknya kelompok kode pos.  $N$  baris selanjutnya berisi 2 bilangan. Bilangan pertama menyatakan jumlah surat dengan suatu kodepos ( $1 \leq X \leq 60000$ ) dan bilangan kedua menyatakan kodeposnya ( $00001 \leq \text{kodepos} \leq 20000$ ). Sebagai contoh, pada baris pertama dari contoh menyatakan bahwa ada 8 surat yang menuju ke kode pos 02910 dan pada baris kedua menyatakan ada 6 surat yang menuju ke kode pos 01845.

**OUTPUT**

Output satu baris berisi 4 integer  $D, T, A, M$  yang masing – masing dipisahkan dengan 1 spasi. Dimana :

$D \rightarrow$  jumlah kelompok dengan aturan 1  
 $T \rightarrow$  jumlah kelompok dengan aturan 2  
 $A \rightarrow$  jumlah kelompok dengan aturan 3  
 $M \rightarrow$  jumlah kelompok dengan aturan 4

**Contoh Input 1**

```
2
8 02910
6 01845
```

**Contoh output 1**

```
0 0 0 1
```

**Contoh input 2**

```
8
8 02910
16 02920
2 01824
6 01834
9 01845
5 01937
5 02244
15 02736
```

**Contoh Output 2**

```
2 1 1 1
```

## **SOLUSI**

Cara termudah untuk menyelesaikan masalah ini adalah dengan membrute force. Pertama kita membuat array dari 1..20000, yang isinya jumlah dari surat dalam kode pos itu. Lalu kita berusaha mengelompokkan sebanyak mungkin surat dengan aturan D, lalu dengan aturan T, lalu dengan aturan A, dan terakhir dengan aturan M.

## **ALGORITMA**

Baca input dan masukan dalam array *Jum*

```
-- Aturan D
for I := 1 to 20000 { I adalah kodepos }
begin
  jika JUM[i] > 10 maka kelompokkan ke D sampai JUM[i] < 10
end;

-- Aturan T
for I := 1 to 200 do { I adalah 3 digit terdepan }
begin
  For J := 0 to 99 do { J adalah 2 digit belakang }
  Begin
    K := I * 100 + j { K adalah kodeposnya }
    Kelompokkan untuk I yang sama dengan aturan T;
  End;
end;

-- Aturan A
Sama seperti aturan T, tapi bukan untuk I yang sama, melainkan untuk setiap I dalam PDD yang sama.

-- Aturan M
Kelompokkan sisanya dengan aturan ini.

-- Cetak
Writeln(D, 'T, 'A, 'M);
```

## **SOURCE CODE**

```
{ACSL – Widagdo Setiawan – 2/9/2002}
Const PDD : array [1..6,1..8] of integer = (
  (4,105,106,107,108,109,201,201),
  (5,115,117,118,119,201,201,201),
  (6,7,8,9,201,201,201,201),
  (10,11,12,13,14,15,16,17),
  (18,19,21,22,55,201,201,201),
```

```
(20,23,24,25,26,27,28,29));
var i,j,k,n,D,T,A,M,l,p: longint;
    Jumlah : array [1..20000] of integer;
    Jum3 : array [0..200] of integer;
    ada : boolean;
begin
assign(input,'ACSL.IN');reset(input);
assign(output,'ACSL.OUT');rewrite(output);
readln(n);
for i := 1 to n do
begin
    readln(j,k);
    inc(jumlah[k],j);
end;
for i := 1 to 20000 do
begin
    if jumlah[i] < 10 then continue;
    d := d + jumlah[i] div 125;
    jumlah[i] := jumlah[i] - (jumlah[i] div 125) * 125;
    if jumlah[i] >= 10 then
    begin
        jumlah[i] := 0;
        inc(d);
    end;
end;
for i := 0 to 200 do
begin
    k := 0;
    for j := i * 100 to i * 100 + 99 do
    begin
        k := k + jumlah[j];
    end;
    if k >= 10 then
    begin
        t := t + k div 125;
        l := 0;
        if k - k div 125 >= 10 then
        begin
            inc(t);
            for j := i * 100 to i * 100 + 99 do
            begin
                jumlah[j] := 0;
            end;
        end else begin
            for j := i * 100 to i * 100 + 99 do
            begin
```



```
    l := l + jumlah[j];
    if l > (k div 125) * 125 then
    begin
        jumlah[j] := l - (k div 125) * 125;
        break;
    end else begin
        jumlah[j] := 0;
    end;
end;
end;
end;
for p := 1 to 6 do
begin
    k := 0;
    for i := 0 to 200 do
    begin
        ada := false;
        for j := 1 to 8 do
        begin
            if pdd[p,j] = i then ada := true;
        end;
        if not ada then continue;
        for j := i * 100 to i * 100 + 99 do
        begin
            k := k + jumlah[j];
        end;
    end;
end;

if k >= 10 then
begin
    a := a + k div 125;
    l := 0;
    if k - k div 125 >= 10 then
    begin
        inc(a);
        for i := 0 to 200 do
        begin
            ada := false;
            for j := 1 to 8 do
            begin
                if pdd[p,j] = i then ada := true;
            end;
            if not ada then continue;
            for j := i * 100 to i * 100 + 99 do
            begin
```

```
    jumlah[j] := 0;
  end;
end else begin
  for i := 0 to 200 do
  begin
    ada := false;
    for j := 1 to 8 do
    begin
      if pdd[p,j] = i then ada := true;
    end;
    if not ada then continue;
    for j := i * 100 to i * 100 + 99 do
    begin
      l := l + jumlah[j];
      if l > (k div 125) * 125 then
      begin
        jumlah[j] := l - (k div 125) * 125;
        break;
      end else begin
        jumlah[j] := 0;
      end;
    end;
  end;
end;
end;
end;
end;
end;
k := 0;
for i := 1 to 20000 do
begin
  k := j + jumlah[i];
end;
m := m + k div 125;
if k - (k div 125) * 125 > 1 then inc(m);
writeln(d, 't', a, m);
close(output);
end.
```



## SOLUSI

Cara mudah untuk menyelesaikan soal ini adalah dengan memprecount dulu semua kemungkinan garis yang dibuat dari kubus itu. Langkah selanjutnya tinggal mencocokkan keadaan data, apakah ada membentuk garis sesuai hasil precounnya. Untuk setiap garis hasilnya disimpan dulu, lalu diurutkan, dan terakhir dicetak.

## ALGORITMA

Baca Input

Bandingkan Input dengan table garis, bila ada, maka catat hasilnya

Sort hasil yang tadi didapat sesuai aturan di soal

Cetak hasil

## SOURCE CODE

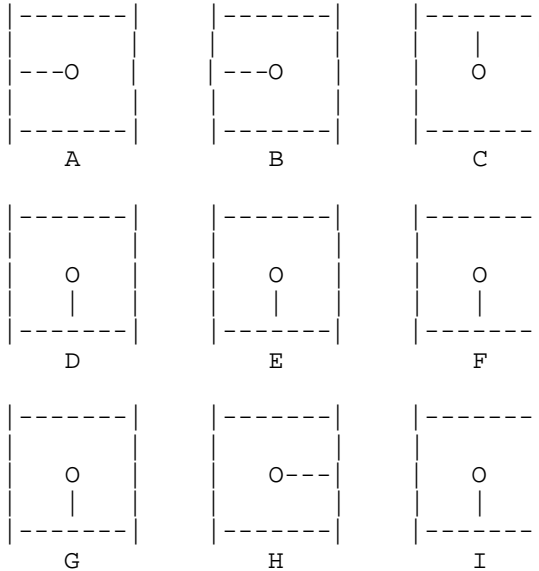
```
{TTT3D – Widagdo Setiawan – 2/9/2002}
const win : array [1..49,1..3] of longint = (
  ( 1, 2, 3), ( 4, 5, 6 ), ( 7, 8, 9 ),
  ( 10,11,12 ), ( 13, 14,15 ), ( 16, 17, 18 ),
  ( 19,20,21), ( 22,23,24), ( 25,26,27),
  ( 1,4,7), ( 2,5,8), ( 3,6,9),
  ( 10,13,16), ( 11,14,17), ( 12,15,18),
  ( 19,22,25), ( 20,23,26), ( 21,24,27),
  ( 1,10,19), ( 2,11,20), ( 3,12,21),
  ( 4,13,22), ( 5,14,23), ( 6,15,24),
  ( 7,16,25), ( 8,17,26), ( 9,18,27),
  ( 1,5,9), ( 3,5,7), ( 10,14,18),
  ( 12,14,16), ( 19,23,27), ( 21,23,25),
  ( 1,13,25), ( 19,13,7),
  ( 2,14,26), ( 20,14,8),
  ( 3,15,27), ( 21,15,9),
  ( 1,11,21), ( 3,11,19),
  ( 4,14,24), ( 6,14,22),
  ( 7,17,27), ( 9,17,25),
  ( 1,14,27), ( 3,14,25), ( 9,14,19), ( 7,14,21));
var i,j,k,nh,t : longint;
    map : array [1..27] of boolean;
    hasil : array [1..48,1..3] of longint;
    ch : char;
begin
  assign(input,'TTT3D.IN');reset(input);
  assign(output,'TTT3D.OUT');rewrite(output);
  for i := 1 to 27 do
  begin
    read(ch);
    if ch = 'X' then map[i] := true;
  end;
  readln;
```

```
readln(k);
map[k] := true;
for i := 1 to 49 do
begin
  if (win[i,1]<>k) and (win[i,2]<>K) and (win[i,3]<>k) then continue;
  if map[win[i,1]] and map[win[i,2]] and map[win[i,3]] then
  begin
    inc(nh);
    hasil[nh,1] := win[i,1];
    hasil[nh,2] := win[i,2];
    hasil[nh,3] := win[i,3];
  end;
end;
for i := 1 to nh do
begin
  for j := 1 to 2 do
  begin
    for k := j + 1 to 3 do
    begin
      if hasil[i,j] > hasil[i,k] then
      begin
        t := hasil[i,j];
        hasil[i,j] := hasil[i,k];
        hasil[i,k] := t;
      end;
    end;
  end;
end;
for i := 1 to nh - 1 do
begin
  for j := i + 1 to nh do
  begin
    if (hasil[i,1] > hasil[j,1]) or
      ((hasil[i,1] = hasil[j,1]) and (hasil[i,2] > hasil[j,2])) then
    begin
      hasil[48] := hasil[i];
      hasil[i] := hasil[j];
      hasil[j] := hasil[48];
    end;
  end;
end;
for i := 1 to nh do
begin
  writeln(hasil[i,1], ' ', hasil[i,2], ' ', hasil[i,3]);
end;
if nh = 0 then writeln(0);
close(output);
end.
```

**SOAL**

**Soal 4. Clock (CLOCK)**

**IOI'94 : The Clocks**



Ada 9 jam dengan bentuk matriks 3 X 3. Tujuan dari permainan ini adalah mengatur agar semua posisi jam menunjuk arah jam 12, dengan langkah sesedikit mungkin. Ada 9 tombol yang berguna untuk memutar jam, dengan karakteristik berbeda – beda. Karakteristiknya yaitu :

Tombol            Jam yang berubah

- |   |       |
|---|-------|
| 1 | ABDE  |
| 2 | ABC   |
| 3 | BCEF  |
| 4 | ADG   |
| 5 | BDEFH |
| 6 | CFI   |
| 7 | DEGH  |
| 8 | GHI   |
| 9 | EFHI  |

Setiap langkah, akan memutar jam sebesar 90'.

**Input**

Ada sembilan integer dari 0..3. Yang artinya 0=12 o'clock, 1=3 o'clock, 2=6 o'clock, 3=9 o'clock.

**Output**

Tampilkan langkah terpendek dalam urutan poemencetan tombol. Output harus diberikan secara terurut dari kecil ke besar, dan bila ada beberapa solusi untuk jumlah langkah minimum, dipilih solusi yang menampilkan urutan angka terkecil.

**Contoh Input**

```
3 3 0
2 2 2
2 1 2
```

**Contoh output**

4589

Contoh gerakan yang dilakukan dalam setiap langkah

```
0 = 12 o'clock
1 = 3 o'clock
2 = 6 o'clock
3 = 9 o'clock
```

```
3 3 0      3 0 0      3 0 0      0 0 0      0 0 0
2 2 2  5-> 3 3 3  8-> 3 3 3  4 -> 0 3 3  9-> 0 0 0
2 1 2      2 2 2      3 3 3      0 3 3      0 0 0
```

## **SOLUSI**

Bila dianalisa, terlihat bahwa tidak ada gunanya menekan tombol yang sama lebih dari 3 kali, jadi hanya ada 4 kemungkinan untuk setiap tombol, yaitu 0 kali, 1 kali, 2 kali, dan 3 kali. Lalu terlihat pula bahwa, urutan menekan tombol tidak menjadi masalah. Jadi hanya ada  $4^9$  cara penekanan tombol yang mungkin (262.144 cara). Angka itu cukup kecil, jadi kita dapat mencobanya satu persatu, dan melihat jawaban yang menghasilkan semua 0. Bila menemukan jawaban yang lebih pendek kita memakai jawaban itu, dan bila menemukan jawaban yang sama panjang, kita memakai jawaban yang menggunakan angka kecil terbanyak, seperti yang dikatakan di soal.

## **ALGORITMA**

Baca input

```
For setiap kemungkinan penekanan tombol do
  if hasil = semua nol then
    if jumlah langkah < jumlah langkah sebelum then
      catat hasil
```

Cetak hasil

## **SOURCE CODE**

```
{ CLOCK – Widagdo Setiawan – 2/9/2002 }
const tabel : array [1..9,1..9] of boolean = (
  (TRUE,TRUE,FALSE,TRUE,TRUE,FALSE,FALSE,FALSE,FALSE),
  (TRUE,TRUE,TRUE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE),
  (FALSE,TRUE,TRUE,FALSE,TRUE,TRUE,FALSE,FALSE,FALSE),
  (TRUE,FALSE,FALSE,TRUE,FALSE,FALSE,TRUE,FALSE,FALSE),
  (FALSE,TRUE,FALSE,TRUE,TRUE,TRUE,FALSE,TRUE,FALSE),
  (FALSE,FALSE,TRUE,FALSE,FALSE,TRUE,FALSE,FALSE,TRUE),
  (FALSE,FALSE,FALSE,TRUE,TRUE,FALSE,TRUE,TRUE,FALSE),
  (FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,TRUE,TRUE,TRUE),
  (FALSE,FALSE,FALSE,FALSE,TRUE,TRUE,FALSE,TRUE,TRUE));
var i,j,k,min : longint;
    map : array [1..9] of longint;
    langkah,hasil : array [1..9] of longint;
Procedure putar(x:longint);
var i : longint;
begin
  for i := 1 to 9 do
    begin
      if not tabel[x,i] then continue;
      inc(map[i]);
```



```
    if map[i] = 4 then map[i] := 0;
  end;
end;
Procedure rekursi (lev : longint);
var i,j,k : longint;
begin
  if lev = 10 then
  begin
    for i := 1 to 9 do if map[i] > 0 then exit;
    k := 0;
    for i := 1 to 9 do k := k + langkah[i];
    if min > k then
    begin
      min := k;
      hasil := langkah;
    end else if min = k then begin
      for i := 1 to 9 do
      begin
        if hasil[i] > langkah[i] then
        begin
          hasil := langkah;
          break;
        end;
      end;
    end;
    exit;
  end;
  rekursi(lev+1);
  inc(langkah[lev]);putar(lev);rekursi(lev+1);
  inc(langkah[lev]);putar(lev);rekursi(lev+1);
  inc(langkah[lev]);putar(lev);rekursi(lev+1);
  langkah[lev] := 0; putar(lev);
end;
begin
  assign(input,'CLOCK.IN');reset(input);
  assign(output,'CLOCK.OUT');rewrite(output);
  for i := 1 to 9 do read(map[i]);
  min := 10000;
  rekursi(1);
  for i := 1 to 9 do
  begin
    for j := 1 to hasil[i] do write(i);
  end;
  writeln;
  close(output);
end.
```