

ĐỀ THI CHÍNH THỨC

Môn thi: TIN HỌC  
Buổi thi thứ nhất: Sáng 12/01/2020

**HƯỚNG DẪN CHẤM THI**  
(Bản hướng dẫn này gồm 06 trang)

**I. Hướng dẫn chung**

- 1) Nếu thí sinh làm bài không theo cách nêu trong đáp án nhưng đúng thì Tổ chấm thi trao đổi, thống nhất thang điểm và vẫn cho đủ số điểm từng phần như hướng dẫn quy định;
- 2) Việc chi tiết hoá (nếu có) thang điểm trong hướng dẫn chấm phải đảm bảo không làm sai lệch hướng dẫn chấm và phải được thống nhất thực hiện trong Tổ chấm thi.

**II. Đáp án và thang điểm**

BÀI	ĐÁP ÁN		ĐIỂM
<b>Bài 1</b>	<b>Chữ số</b> Lời giải: Sử dụng mảng thống kê tke['0'..'9'] để thống kê số lượng các chữ số xuất hiện trong F		<b>6,0</b>
<b>Test</b>	CHUSO.INP	CHUSO.OUT	
1	6	1 1 2 1	Mỗi Test 0.5 điểm
2	10	5 2	
3	1000	0 4 5 2	
4	5324	0 1 1 3 4 1 5 2 7 1	
5	File chuso05.inp	File chuso05.out	
6	File chuso06.inp	File chuso06.out	
7	File chuso07.inp	File chuso07.out	
8	File chuso08.inp	File chuso08.out	
9	File chuso09.inp	File chuso09.out	
10	File chuso10.inp	File chuso10.out	
11	File chuso11.inp	File chuso11.out	
12	File chuso12.inp	File chuso12.out	
<b>Bài 2</b>	<b>Hoán vị</b> Lời giải: - Thống kê các loại chữ số xuất hiện trong N, ví dụ tke['0'..'9'] - In mảng tke từ phải sang trái, in chỉ số 'i' tke['i'] lần		<b>7,0</b>

<b>Test</b>	<b>HOANVI.INP</b>	<b>HOANVI.OUT</b>	
1	22311	32211	Mỗi Test 0.5 điểm
2	750249	975420	
3	1000000	1000000	
4	37696799	99977663	
5	File hoanvi05.inp	File hoanvi05.out	
6	File hoanvi06.inp	File hoanvi06.out	
7	File hoanvi07.inp	File hoanvi07.out	
8	File hoanvi08.inp	File hoanvi08.out	
9	File hoanvi09.inp	File hoanvi09.out	
10	File hoanvi10.inp	File hoanvi10.out	
11	File hoanvi11.inp	File hoanvi11.out	
12	File hoanvi12.inp	File hoanvi12.out	
13	File hoanvi13.inp	File hoanvi13.out	
14	File hoanvi14.inp	File hoanvi14.out	
<b>Bài 3</b>	<b>Robot</b> Lời giải: Tìm đường đi ngắn nhất trên mảng hai chiều (Dijkstra)		<b>7,0</b>
<b>Test</b>	<b>ROBOT.INP</b>	<b>ROBOT.OUT</b>	
1	4 3 0 3 2 2 5 1 5 3 1 3 1 1 3 2	10	Mỗi Test 0.5 điểm
2	4 3 0 3 2 2 5 1 5 3 1 3 1 1 2 3	6	
3	5 5 0 1 5 5	8	
4	5 5 0 1 2 1 1 1 2 1 1 1 1 1 3 1 1 1 1 1 1 3 1 1 1 1 1 4 5	9	
5	File robot05.inp	File robot05.out	
6	File robot06.inp	File robot06.out	
7	File robot07.inp	File robot07.out	
8	File robot08.inp	File robot08.out	

9	File robot09.inp	File robot09.out	
10	File robot10.inp	File robot10.out	
11	File robot11.inp	File robot11.out	
12	File robot12.inp	File robot12.out	
13	File robot13.inp	File robot13.out	
14	File robot14.inp	File robot14.out	

-----**HẾT**-----

## Code tham khảo

### Bài 1. Chữ số

```
var tke: array[0..9] of longint;  
    n, du: longint;  
    f: longint;  
//-----  
procedure xuli;  
var i: longint;  
begin  
    fillchar(tke,sizeof(tke),0);  
    f := 0;  
    for i:=1 to n do f := f+i;  
    while (f>0) do  
        begin  
            du := f mod 10; inc(tke[du]);  
            f := f div 10;  
        end;  
  
    for i:=0 to 9 do  
        if (tke[i]>0) then writeln(i,' ',tke[i]);  
end;  
//-----  
BEGIN {Main}  
    assign(input,'chuso.inp'); reset(input);  
    assign(output,'chuso.out'); rewrite(output);  
    read(n);  
    xuli;  
    close(input); close(output);  
END.
```

### Bài 2. Hoán vị

```
var tke: array['0'..'9'] of longint;  
    n: ansistring;  
//-----  
procedure xuli;  
var i,j: longint; k: char;  
begin  
    fillchar(tke,sizeof(tke),0);  
    for i:=1 to length(n) do inc(tke[n[i]]);  
  
    for k:='9' downto '0' do  
        if (tke[k]>0) then  
            for j:=1 to tke[k] do write(k);  
    end;  
//-----
```

```

BEGIN {Main}
  assign(input,'hoanvi.inp'); reset(input);
  assign(output,'hoanvi.out'); rewrite(output);
  read(n);
  xuli;
  close(input); close(output);
END.

```

### Bài 3. Robot

```

uses math;
const maxM = 100; MaxN = 100;
  Vx : array[1..4] of -1..1 =(-1,0,1,0);
  Vy : array[1..4] of -1..1 =(0,1,0,-1);

var C: array[1..maxM,1..maxN] of longint;
  F: array[1..maxN,1..maxN] of int64;
  free: array[1..maxM,1..maxN] of boolean;
  K,M,N,a,b: longint;
//-----
procedure Dijkstra;
var i,j,ux,uy,min,p,u,v: longint;
begin
  fillchar(free,sizeof(free),true);
  for i:=1 to M do
    for j:=1 to N do F[i,j] := maxlongint;

  F[1,1] := C[1,1];
  free[1,1] := false;
  F[1,2] := F[1,1] + C[1,2];
  F[2,1] := F[1,1] + C[2,1];

  repeat
    ux := 0; uy:=0; min := maxlongint;
    for i := 1 to M do
      for j := 1 to N do
        if Free[i,j] and (F[i,j] < min) then
          begin min := F[i,j]; ux := i; uy := j; end;

    if ((ux=a) and (uy=b)) or ((ux = 0) and (uy=0)) then Break;
    Free[ux,uy] := False;

    for p:=1 to 4 do
      begin
        u := ux + Vx[p];
        v := uy + Vy[p];
        if (u>=1) and (u<=M) and (v>=1) and (v<=N) then
          begin
            if free[u,v] and (F[ux,uy]+C[u,v] < F[u,v]) then F[u,v] := F[ux,uy] + C[u,v];

```

```

        end;
    end;
until False;
end;
//-----
procedure nhap;
var i,u,v: longint;
begin
    readln(M,N);
    for u:=1 to M do
        for v:= 1 to N do read(C[u,v]);
    readln(a,b);
    Dijkstra;
    writeln(F[a,b]);
end;
//-----
BEGIN {Main}
assign(input,'robot.inp'); reset(input);
assign(output,'robot.out'); rewrite(output);
nhap;
close(input); close(output);
END.

```