

ĐỀ THI CHÍNH THỨC

Môn thi: TIN HỌC
Buổi thi thứ nhất: Sáng 11/10/2020

HƯỚNG DẪN CHẤM THI
(Bản hướng dẫn này gồm 07 trang)

I. Hướng dẫn chung

1) Mỗi bài thi của thí sinh sẽ được chấm bằng cách:

- Chạy chương trình của thí sinh với bộ dữ liệu vào của hướng dẫn chấm; chương trình của thí sinh sẽ xuất ra các kết quả (gọi là kết quả của thí sinh) tương ứng các dữ liệu vào.
- Đối chiếu sự trùng khớp giữa các kết quả của thí sinh với các kết quả của hướng dẫn chấm.

2) Việc chạy chương trình của thí sinh và đối chiếu sự trùng khớp kết quả có thể thực hiện thủ công hoặc bằng phần mềm chấm thi. Ngoài ra, phần mềm chấm thi sẽ tính thời gian chạy chương trình nếu đề bài có yêu cầu.

3) Trường hợp bài làm của thí sinh có lỗi (không biên dịch được), phần tổ chức dữ liệu và nhập dữ liệu được chấm tối đa 25% số điểm của bài.

II. Đáp án và thang điểm

BÀI	ĐÁP ÁN		ĐIỂM
Bài 1	Sơ nền Bài toán quy về tìm phần tử thứ n của dãy Fibonacci với: $F[1] = 1; F[2] = 1;$ Với $n \geq 3, F[n] = F[n-2] + F[n-1]$		6,0
Test	SONNEN.INP	SONNEN.OUT	
1	7	13	Mỗi Test 0.5 điểm
2	9	34	
3	10	55	
4	13	233	
5	16	987	
6	18	2584	
7	19	4181	
8	20	6765	
9	35	9227465	
10	46	1836311903	
11	87	679891637638612258	
12	90	2880067194370816120	

Bài 2	Nghi vấn Duyệt toàn bộ phiếu trả lời trắc nghiệm theo từng câu, với mỗi câu, kiểm tra 4 ô thuộc câu đó xem có ô nào thuộc diện nghi vấn không? Nếu có thì phân tích các ô còn lại để xác định câu này thuộc loại nào: Nghi vấn chưa tô? Nghi vấn tô đúng?		7,0
Test	NGHIVAN.INP	NGHIVAN.OUT	
1	3 0 0 0 90 125 150 0 0 250 0 0 80	2 1 1	Mỗi Test 0.5 điểm
2	7 250 0 0 0 0 0 0 95 100 90 80 70 250 0 0 80 0 0 55 0 0 250 255 250 255 0 20 45 85	5 2 3	
3	8 200 0 0 9 8 0 0 90 125 150 0 0 200 0 0 80 7 0 15 0 250 255 250 255 15 20 0 100 0 0 40 0	5 3 2	
4	10 250 0 0 9 8 0 0 100 125 150 0 0 250 0 0 80 0 0 15 0 0 250 255 250 255 15 20 0 210 0 0 180 0 0 0 10 0 0 0 180 150	3 1 2	
5	File_input05	File_output 05	
6	File_input06	File_output 06	
7	File_input07	File_output 07	
8	File_input08	File_output 08	
9	File_input09	File_output 09	
10	File_input10	File_output 10	
11	File_input11	File_output 11	
12	File_input12	File_output 12	
13	File_input13	File_output 13	
14	File_input14	File_output 14	

Bài 3	Diễn đàn Cách 1: Sinh dãy nhị phân độ dài n - Sinh dãy nhị phân độ dài n, mỗi dãy có M phần tử mang giá trị 1 - Với mỗi dãy nhị phân được sinh ra, tìm khoảng cách tối thiểu giữa hai phần tử mang giá trị 1, ghi nhận khoảng cách này vào biến Smin Cách 2: Tìm kiếm nhị phân - Tìm khoảng cách D lớn nhất sao cho có thể bố trí đủ M doanh nghiệp. - D được tìm trong phạm vi $\{D_{min} = 1; D_{max} = (x_N - x_0)\}$ bằng phương pháp tìm kiếm nhị phân. while $D_{min} \leq D_{max}$ do begin $D := (D_{min} + D_{max}) \text{ div } 2;$ if kiểmtra(D) then begin $S_{min} := D;$ $D_{min} := D + 1;$ end else $D_{max} := D - 1;$ end;		7,0
Test	DIENDAN.INP	DIENDAN.OUT	
1	2 2 2 1 10 6 6 8 6 10 2 4 11	9 1	Mỗi Test 0,5 điểm
2	2 2 2 0 1000000 5 3 0 200000 400000 600000 800000	1000000 400000	
3	2 5 3 1000 3000 5000 7000 9000	4000 2000	

	4 4 3000 5000 7000 9000		
4	1 15 8 200000 400000 600000 800000 1000000 20000 40000 60000 80000 100000 2000 4000 6000 8000 10000	40000	
5	File_input05	File_output 05	
6	File_input06	File_output 06	
7	File_input07	File_output 07	
8	File_input08	File_output 08	
9	File_input09	File_output 09	
10	File_input10	File_output 10	
11	File_input11	File_output 11	
12	File_input12	File_output 12	
13	File_input13	File_output 13	
14	File_input14	File_output 14	

III. Code tham khảo

Bài 2. Nghi vấn

```
uses math;
const maxN = 1000000;
var F: array[1..maxN,1..4] of byte;
    mucto: array[0..2] of byte;
    m,ans: longint;

//-----
procedure nhap;
var i,j: longint;
begin
    readln(m);
    for i:=1 to m do
        for j:=1 to 4 do
            read(F[i,j]);
end;
//-----
procedure xuli;
var i, j, chuato, todup: longint;
begin
    chuato := 0; todup := 0;
    for i:=1 to m do
        begin
            fillchar(mucto,sizeof(mucto),0);
            for j:=1 to 4 do
                begin
                    if (F[i,j] <= 10) then inc(mucto[0]);
                    if (F[i,j] > 10) and (F[i,j] < 100) then inc(mucto[1]);
                    if (F[i,j] >= 100) then inc(mucto[2]);
                end;
            if (mucto[1] = 1) and (mucto[2] = 0) then inc(chuato);
            if ((mucto[1] > 1) and (mucto[2] < 2)) or ((mucto[1] = 1) and (mucto[2] = 1)) then
inc(todup);
        end;
        write(chuato + todup, ' ', chuato, ' ', todup);
    end;
//-----
BEGIN {Main}
    assign(input,'nghivan.inp'); reset(input);
    assign(output,'nghivan.out'); rewrite(output);
    nhap;
    xuli;
    close(input); close(output);
END.
```

Bài 3. Diễn đàn

```
uses math;
const
    MaxN = 100000;    maxL = 1000000000;
    fi = 'diendan.inp';    fo = 'diendan.out';
var diadiem: array[0..MaxN+1] of longint;
    free: array[0..MaxN+1] of boolean;
    N,M,T,dem,luui,d: longint;

function Kiemtra(d:longint): boolean;
var i,j: longint; kc: longint;
begin
    fillchar(free,sizeof(free),true);
    free[0] := false; free[n+1] := false;
    free[1] := false; luui := 1;
    dem := 1;
    for i:=1 to n do
        if (free[i]=false) then
            begin
                for j:=i+1 to n do
                    if ((free[j]) and (diadiem[j]-diadiem[luui]>=d)) then
                        begin
                            free[j] := false;        inc(dem);        luui := j;        break;
                        end;
                end;
                if dem<m then exit(false) else exit(true);
            end;
    end;
//-----
procedure xuli;
var i,j: longint; Dmax,Dmin,Smin: longint;
begin
    Dmax := diadiem[n]-diadiem[1];
    Dmin := 1;
    Smin := 0;
    while Dmin<=Dmax do // tìm D bằng cách tìm kiếm nhị phân
        begin
            D := (Dmin + Dmax) div 2;
            if kiemtra(D) then
                begin
                    Smin := D;
                    Dmin := D + 1;
                end
            else Dmax := D - 1;
        end;
    writeln(Smin);
end;
```

```

//-----
procedure DoiCho(var a,b:longint);
var tam: longint;
begin
    tam := a;  a := b;  b := tam;
end;
//-----
Procedure QuickSort(L,H:longint);
Var i,j, Chot: longint;
Begin
    If L>=H then Exit;
    Chot := diadiem[(L+H) div 2];
    i := L;
    j := H;
    Repeat
        While diadiem[i] < Chot do inc(i);
        While diadiem[j] > Chot do dec(j);
        If i<=j then
            Begin
                If i<j then DoiCho(diadiem[i],diadiem[j]);
                Inc(i); Dec(j);
            End;
        Until i>j;

        QuickSort(L,j);
        QuickSort(i,H);
    End;
//-----
procedure nhap;
var i,j: longint;
begin
    readln(T);
    for i:=1 to T do
        begin
            readln(n,m);
            for j:= 1 to n do readln(diadiem[j]);
            QuickSort(1,n);
            xuli;
        end;
    end;
//-----
BEGIN
    assign(input,fi); reset(input); assign(output,fo); rewrite(output);
    nhap;
    close(input); close(output);
END.

```