

Tổng quan đề thi:

Tên bài	Tên tệp chương trình	Tên tệp dữ liệu vào	Tên tệp dữ liệu ra
Bài 1. Đếm ước số	BL1.*	UOCSO.INP	UOCSO.OUT
Bài 2. Đua Robot	BL2.*	ROBOT.INP	ROBOT.OUT
Bài 3. Tuyển dò ngang	BL3.*	TUYENDO.INP	TUYENDO.OUT

**Ghi chú:** dấu \* đại diện cho phân mở rộng, tùy theo ngôn ngữ lập trình có thể là PAS hoặc CPP. Thời gian thực hiện chương trình không quá 1 giây.

**Câu 1. (6,0 điểm) ĐẾM ƯỚC SỐ**

Bạn Nam rất thích học toán, đặc biệt là các bài toán về tổ hợp. Các công thức tổ hợp như đếm số hoán vị, số chỉnh hợp hay số tổ hợp đều cần phải tính các giá trị giai thừa. Như ta đã biết, giai thừa của một số nguyên dương  $N$  (ký hiệu là  $N!$ ) bằng tích của các số tự nhiên từ 1 đến  $N$ , chẳng hạn  $4! = 1 \times 2 \times 3 \times 4 = 24$ , hay  $6! = 1 \times 2 \times 3 \times 4 \times 5 \times 6 = 720$ . Khi tính toán các giá trị giai thừa như vậy, bạn Nam nhận thấy kết quả tính được có số lượng ước số nhiều hơn số thừa số tạo ra kết quả đó. Ví dụ,  $4!$  là tích của bốn thừa số 1, 2, 3 và 4 nhưng kết quả là 24 thì lại có tới 8 ước số là 1, 2, 3, 4, 6, 8, 12, 24.

**Yêu cầu:** Với một số nguyên dương  $N$ , bạn hãy lập trình giúp bạn Nam tính xem giá trị  $N!$  có bao nhiêu ước số.

**Dữ liệu vào:** Cho từ tệp văn bản UOCSO.INP gồm một dòng ghi số nguyên dương  $N$ .

**Kết quả:** Ghi vào tệp văn bản UOCSO.OUT gồm một dòng ghi kết quả tính được.

**Ví dụ:**

UOCSO.INP	UOCSO.OUT
4	8
6	30

**Giới hạn dữ liệu:**

- \* Có 70% số điểm ứng với giá trị  $N \leq 10$
- \* Có 20% số điểm ứng với giá trị  $N \leq 15$
- \* Có 10% số điểm ứng với giá trị  $N \leq 20$

**Câu 2. (7,0 điểm) ĐUA ROBOT**

Câu lạc bộ Robocon của trường vừa tổ chức cuộc đua Robot. Có  $N$  đội tham gia cuộc đua, mỗi đội sẽ lập trình cho Robot di chuyển trên một đường thẳng. Ban đầu tất cả Robot của các đội đặt tại vạch xuất phát được đánh tọa độ là 0. Robot có thể được lập trình để di chuyển về phía trước hoặc đi ngược về phía sau. Dọc theo đường đi, ban tổ chức đặt  $M$  trạm kiểm soát, trạm kiểm soát thứ  $i$  đặt tại vị trí có tọa độ  $V_i$  ( $|V_i| \leq 10^9$ ,  $i=1..M$ ). Sau thời gian di chuyển, Robot thứ  $j$  dừng lại tại vị trí có tọa độ  $R_j$  ( $|R_j| \leq 10^9$ ,  $j=1..N$ ) và điểm được tính bằng tổng khoảng cách từ Robot đến tất cả các trạm kiểm soát.

**Yêu cầu:** Hãy lập trình tính điểm cao nhất của Robot tại vị trí dừng lại.

**Dữ liệu vào:** Cho từ tệp văn bản ROBOT.INP có dạng:

- \* Dòng đầu ghi hai số nguyên dương  $N$  và  $M$
- \* Dòng thứ hai ghi  $N$  số nguyên  $R_1, R_2, \dots, R_N$
- \* Dòng thứ ba ghi  $M$  số nguyên  $V_1, V_2, \dots, V_M$

Các số trên cùng một dòng ghi cách nhau ít nhất một khoảng cách.

**Kết quả:** Ghi vào tệp văn bản ROBOT.OUT gồm một dòng ghi điểm số cao nhất tính được.

**Ví dụ:**

ROBOT.INP	ROBOT. OUT
3 3 -2 3 2 2 -3 -1	11

**Giải thích:** Robot thứ 2 có điểm số cao nhất là 11

- Robot thứ 1 ở vị trí -2 cách trạm 1 là 4, cách trạm 2 là 1, cách trạm 3 là 1, tổng khoảng cách 6.
- Robot thứ 2 ở vị trí 3 cách trạm 1 là 1, cách trạm 2 là 6, cách trạm 3 là 4, tổng khoảng cách 11.
- Robot thứ 3 ở vị trí 2 cách trạm 1 là 0, cách trạm 2 là 5, cách trạm 3 là 3, tổng khoảng cách 8.

**Giới hạn dữ liệu:**

- \* Có 60% số điểm ứng với giá trị  $N, M \leq 10^4$
- \* Có 40% số điểm ứng với giá trị  $N, M \leq 10^6$

### Câu 3. (7,0 điểm) TUYẾN ĐÒ NGANG

Sông Tiền là một trong hai nhánh sông lớn của dòng sông MeKong. Trước đây, việc giao thông qua lại giữa hai bờ sông chủ yếu bằng phà và các tuyến đò ngang. Từ năm 2000, cầu Mỹ Thuận đã được đưa vào sử dụng và sắp tới đây sẽ tiếp tục khánh thành cầu Cao Lãnh nối liền hai bờ sông Tiền. Có thể nói, những công trình này cùng với cầu Cần Thơ và cầu Vàm Cống (cũng sắp hoàn thành) đã góp phần rất lớn trong việc phát triển kinh tế của vùng ĐBSCL, kết nối vùng ĐBSCL gần hơn với TP.HCM và các tỉnh miền Đông Nam bộ. Mặc dù vậy, chúng ta cũng không thể phủ nhận vai trò của những tuyến đò ngang hiện nay vì sự tiện lợi của nó thay vì phải đi vòng những đoạn đường xa để qua sông bằng cầu.

Hiện tại, dọc theo sông Tiền có  $N$  tuyến đò ngang đang hoạt động, mỗi tuyến đò đi từ một bến đò ở bờ Bắc sông Tiền sang một bến ở bờ Nam sông Tiền và ngược lại. Các bến đò ở mỗi bờ được đánh số từ 1 đến  $N$  theo thứ tự từ thượng nguồn xuống hạ nguồn. Biết rằng, nếu tuyến đò xuất phát từ bến đò  $i$  ở bờ Bắc thì sẽ đi sang bến đò  $a_i$  ở bờ Nam ( $i=1..N, a_i = 1..N, a_i \neq a_j \forall i \neq j$ ). Vấn đề phát sinh là giữa một số tuyến đò có hành trình khi sang sông bị giao cắt nhau và như vậy có thể không an toàn cho hành khách vì có thể xảy ra tai nạn va chạm trên sông. Chính vì vậy chính quyền địa phương dự định sẽ cho dừng hoạt động một số tuyến đò, chỉ giữ lại những tuyến đò mà hành trình của nó không giao cắt với hành trình của những tuyến khác được giữ lại.

**Yêu cầu:** Hãy tìm một phương án dừng hoạt động một số ít nhất các tuyến đò sao cho những tuyến đò hoạt động còn lại có hành trình không giao cắt nhau.

**Dữ liệu vào:** Cho từ tệp văn bản TUYENDO.INP có dạng:

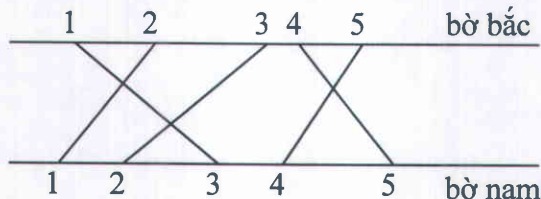
- \* Dòng thứ nhất ghi số nguyên dương  $N$
- \* Dòng thứ hai ghi  $N$  số nguyên dương  $a_1, a_2, \dots, a_N$ . Giữa các số cách nhau một khoảng cách.

**Kết quả:** Ghi vào tệp văn bản TUYENDO.OUT gồm một dòng ghi một số nguyên là số tuyến đò ít nhất cần phải dừng hoạt động.

**Ví dụ:**

TUYENDO.INP	TUYENDO. OUT
5 3 1 2 5 4	2

**Giải thích:** Có ít nhất 2 tuyến đò phải dừng hoạt động là tuyến 1 và 4 hoặc tuyến 1 và 5.



**Giới hạn dữ liệu:**

- Có 20% số điểm ứng với giá trị  $N \leq 20$
- Có 50% số điểm ứng với giá trị  $N \leq 10^3$
- Có 30% số điểm ứng với giá trị  $N \leq 10^5$

--- HẾT ---

Họ và tên thí sinh: \_\_\_\_\_

Số báo danh: \_\_\_\_\_

Chữ ký GT1: \_\_\_\_\_

Chữ ký GT2: \_\_\_\_\_

### I. Hướng dẫn chung

1) Bài thi được chấm trên máy tính bằng phần mềm Themis – chương trình chấm bài tự động. Giám khảo cũng có thể chấm bài thủ công theo bộ test. Máy tính chấm bài là loại máy có tốc độ CPU từ 3.0GHz trở lên, sử dụng hệ điều hành Windows XP trở lên.

- Thời gian thực hiện chương trình không quá 01 giây/test.
- Bộ test chấm bài được lưu trong đĩa kèm theo.
- Sử dụng trình biên dịch Free Pascal phiên bản 2.4.0, Code::blocks 11 trở lên.

2) Việc chi tiết hóa thang điểm (nếu có) trong hướng dẫn chấm phải bảo đảm không làm sai lệch hướng dẫn chấm và phải được thống nhất thực hiện trong tổ chấm.

### II. Đáp án và thang điểm

#### Câu 1. (6,0 điểm)

Test	UOCSO.INP	UOCSO.OUT	Điểm
1	1	1	0,6
2	2	2	0,6
3	5	16	0,6
4	7	60	0,6
5	8	96	0,6
6	9	160	0,6
7	10	270	0,6
8	14	2592	0,6
9	15	4032	0,6
10	20	41040	0,6

#### Câu 2. (7,0 điểm)

Test	ROBOT.INP	ROBOT.OUT	Điểm
1	N=60 M=50	1717	0,7
2	N=300 M=700	310.555	0,7
3	N=1.000 M=600	3.975.465	0,7
4	N=4.000 M=8.000	75.931.468	0,7
5	N=7.000 M=5.000	1.003.622.481	0,7
6	N=10.000 M=10.000	19.955.474.404	0,7

7	N=600.000 M=300.000	300.128.946.173.054	0,7
8	N=500.000 M=500.000	500.118.224.892.429	0,7
9	N=100.000 M=1.000.000	1.000.276.525.088.960	0,7
10	N=1.000.000 M=1.000.000	1.000.043.915.280.042	0,7

**Câu 3. (7,0 điểm)**

Test	TUYENDO.INP	TUYENDO.OUT	Điểm
1	10 7 3 6 10 9 8 1 5 4 2	7	0,7
2	20 14 12 11 15 20 19 5 16 18 2 10 13 4 8 7 1 6 17 9 3	16	0,7
3	N = 700	655	0,7
4	N = 800	750	0,7
5	N = 900	845	0,7
6	N = 950	0	0,7
7	N = 1.000	942	0,7
8	N = 80.000	79438	0,7
9	N = 90.000	89406	0,7
10	N = 100.000	99378	0,7

--- HẾT---

## CHƯƠNG TRÌNH THAM KHẢO

**Bài 1.**

```

Program UOCSO;
Const fin = 'UOCSO.INP';
      fout= 'UOCSO.OUT';
Var A,B:Array[1..50] of Longint;
    k,i,t,N,j:Longint;
    S:Int64;
    f:Text;
Begin
  Assign(f,fin);
  Reset(f);
  Read(f,N);
  Close(f);
  k:=0;
  For i:=2 to N do
    Begin
      t:=i;
      For j:=1 to k do
        While t mod A[j]=0 do
          Begin

```

```

        Inc(B[j]);
        t:=t div A[j];
    End;
    If t>1 then
    Begin
        Inc(k);
        A[k]:=t;
        B[k]:=1;
    End;
    End;
    S:=1;
    For i:=1 to k do S:=S*(B[i]+1);
    Assign(f,fout);
    ReWrite(f);
    Write(f,S);
    Close(f);
End.

```

---

## Bài 2.

```

Program ROBOT;
Const fin ='ROBOT.INP';
      fout='ROBOT.OUT';
Var Left,Right,R,V:Array[0..1000001] of Int64;
    n,m,i,dau,giua,cuoi:Longint;
    S,max:Int64;
    f:Text;
Procedure QuickSort(L,R:Longint);
    Var i,j,temp,gt:Longint;
    Begin
        i:=L; j:=R; gt:=V[(L+R) div 2];
        Repeat
            While V[i]<gt do Inc(i);
            While gt<V[j] do Dec(j);
            If i<=j then
                Begin
                    temp:=V[i];
                    V[i]:=V[j];
                    V[j]:=temp;
                    Inc(i);
                    Dec(j);
                End;
            Until i>j;
            If L<j then QuickSort(L,j);
            If i<R then QuickSort(i,R);
        End;
    Begin
        Assign(f,fin);
        Reset(f);
        Readln(f,N,M);
        For i:=1 to N do Read(f,R[i]);
        Readln(f);
        For i:=1 to M do Read(f,V[i]);
        Close(f);

        QuickSort(1,M);
    End;

```

```

Left[1]:=0;
For i:=2 to M do Left[i]:=Left[i-1] + (V[i]-V[i-1])*(i-1);
Right[M]:=0;
For i:=M-1 downto 1 do Right[i]:=Right[i+1] + (V[i+1]-V[i])*(M-i);

max:=0;
V[0]:=-trunc(1E9+5);
V[M+1]:=trunc(1E9+5);
For i:=1 to N do
  Begin
    dau:=0;
    cuoi:=M+1;
    Repeat
      giua:=(dau+cuoi) div 2;
      If V[giua]<=R[i] then dau:=giua
      Else cuoi:=giua;
    Until dau+1=cuoi;
    S:=0;
    If dau>0 then S:=S + Left[dau] + (R[i]-V[dau])*dau;
    If cuoi<=M then S:=S + Right[cuoi] + (V[cuoi]-R[i])*(M-dau);
    If max<S then max:=S;
  End;
Assign(f, fout);
ReWrite(f);
Write(f,max);
Close(f);
End.

```

---

### Bài 3.

```

Program TUYENDO;
Const fin ='TUYENDO.INP';
      fout='TUYENDO.OUT';
Var a,T:Array[0..100000] of Longint;
    n,i,dau,giua,cuoi,k:Longint;
    f:Text;
Begin
  Assign(f,fin);
  Reset(f);
  Readln(f,n);
  For i:=1 to N do Read(f,a[i]);
  Close(f);
  k:=1;
  T[1]:=a[1];
  For i:=2 to N do
    Begin
      dau:=0;
      cuoi:=k+1;
      Repeat
        giua:=(dau+cuoi) div 2;
        If a[i]<T[giua] then cuoi:=giua
        Else dau:=giua;
      Until dau+1=cuoi;
      If cuoi<=k then T[cuoi]:=a[i]
    Else
      Begin

```