**CHUYÊN ĐỀ 4. MẢNG 2 CHIỀU**

**a. Nhập mảng**

*Mảng a có n hàng, m cột*

cin>>m>>n;

for(i=1;i<=n;i++)

for(j=1;j<=m;j++) cin>>a[i][j];

**b. Xuất mảng**

for(i=1;i<=n;i++)

{

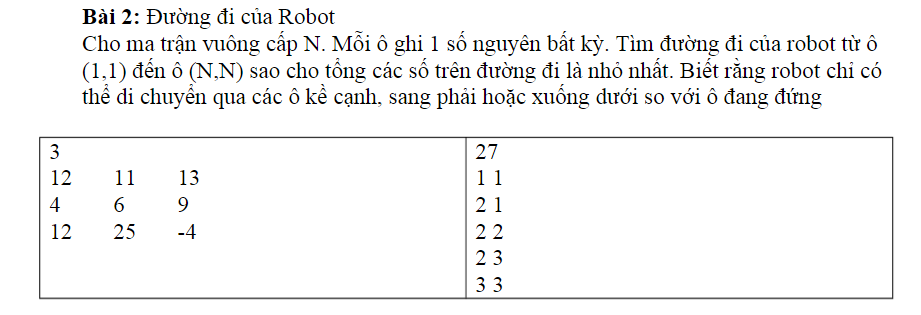
for(j=1;j<=m;j++) cout<<a[i][j]<<" ";

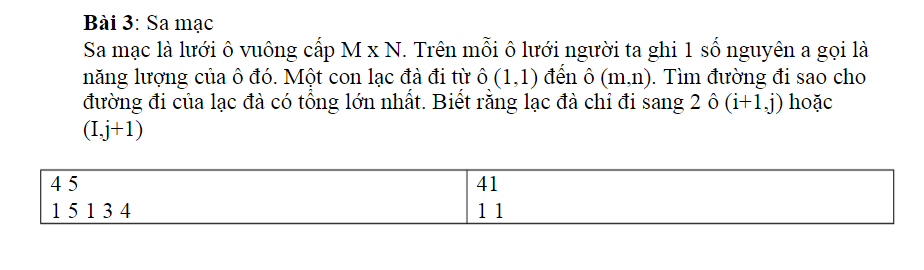
cout<<endl<<endl;

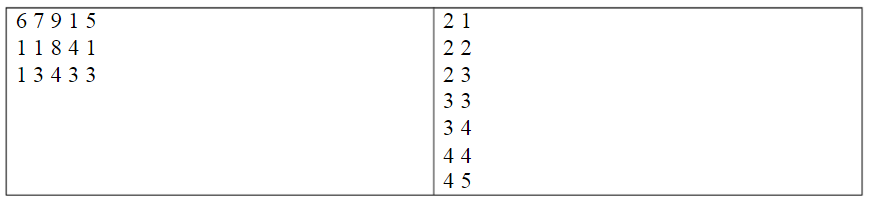
}

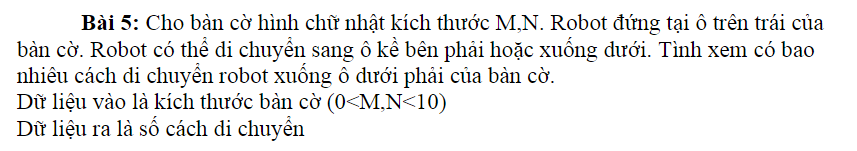
**c. Thao tác trên mảng:**

***Tìm đường đi thỏa mãn điều kiện:***







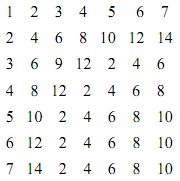


**LUYỆN ĐỀ:**

**Câu 1** (6 điểm):

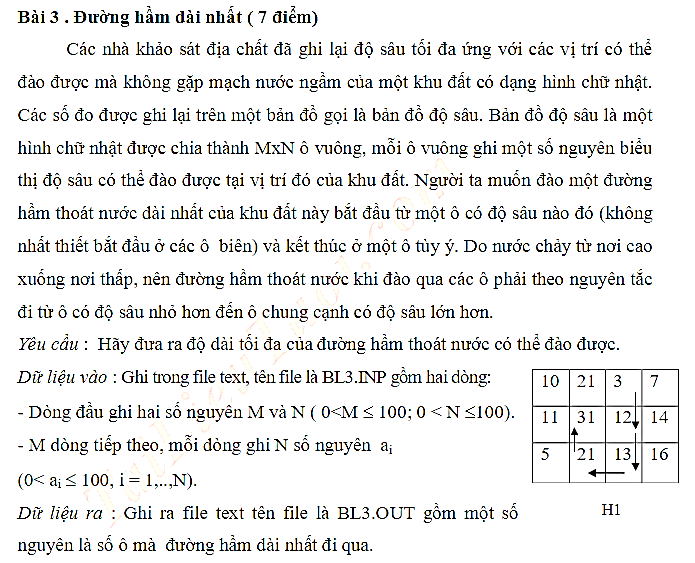
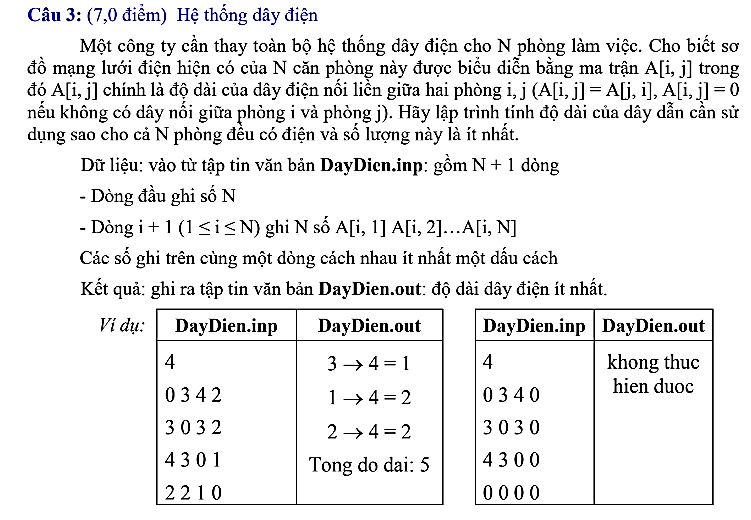
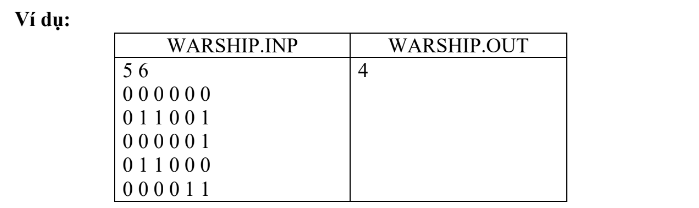
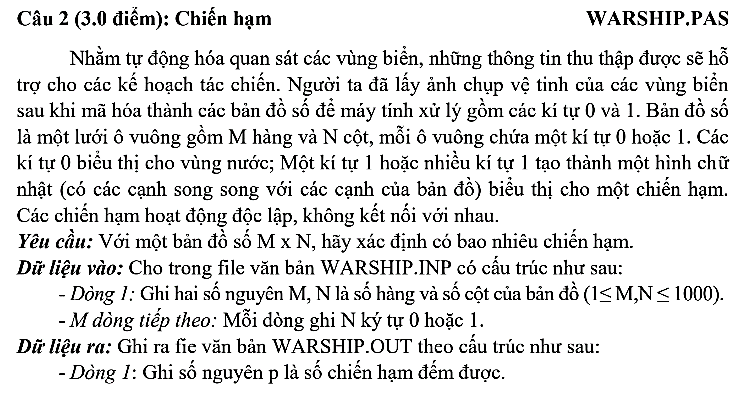
Cho một số nguyên dương N bất kì (N ≤ 20).

Hãy viết chương trình tạo mảng N x N phần tử nguyên dương (giá trị lớn nhất trong bảng là 2\*N) theo quy luật cho trong ví dụ sau với N = 7

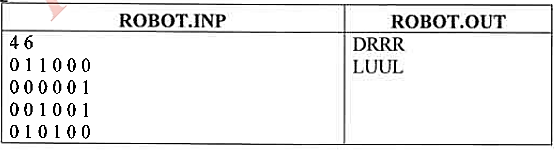
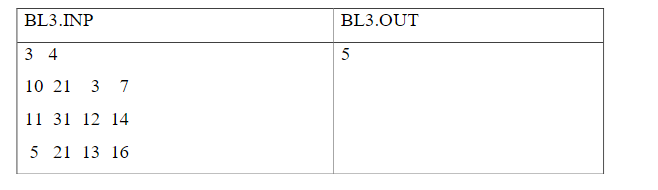
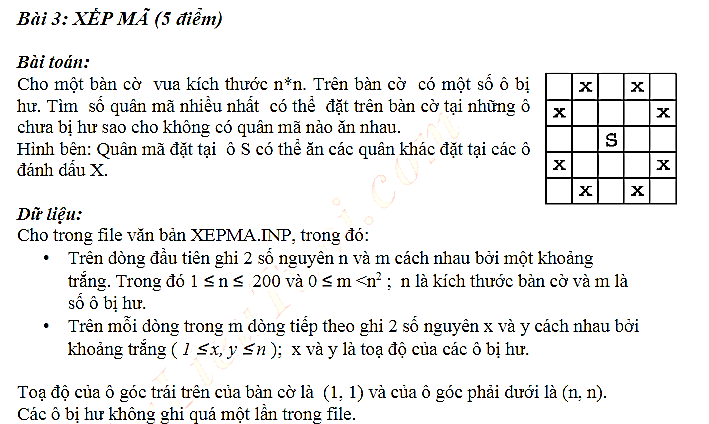
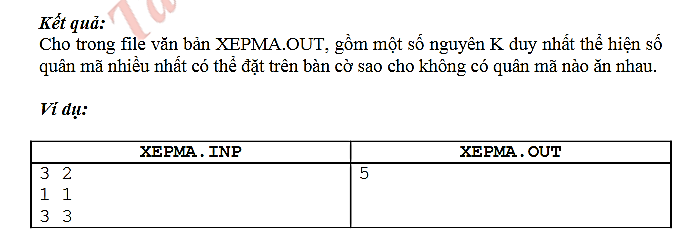
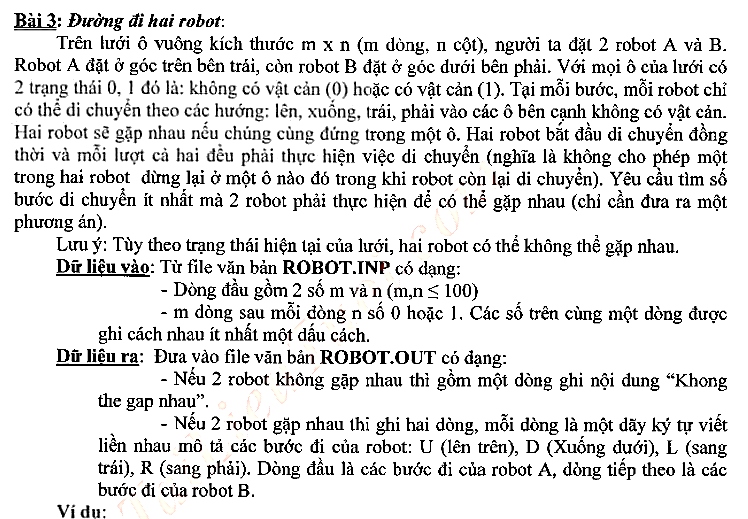


- Dữ liệu vào: Giá trị của N được cho trong tập tin **InB1.txt**

- Dữ liệu ra: Ghi trong tập tin **OutB1.txt** mảng kết quả

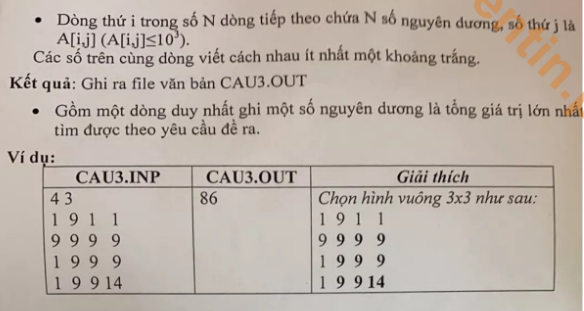
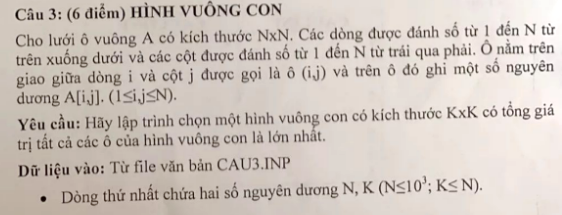


**Câu 4**

****

**Câu 5**

**Câu 6**



**Câu 7**

**7. Di chuyển**

**7.1. Mô hình**

Cho bảng A gồm M∗N ô. Từ ô (i,j)có thể di chuyển sang 3 ô (i+1,j), (i+1,j−1) và (i+1,j+1). Hãy xác định một lộ trình đi từ hàng 1 đến hàng M sao cho tổng các ô đi qua là lớn nhất.

**7.2. Công thức**

Gọi F(i,j) là giá trị lớn nhất có được khi di chuyển đến ô (i,j). Có 3 ô có thể đi đến ô (i,j) là (i−1,j), (i−1,j−1) và (i−1,j+1). Do đó ta có công thức QHĐ như sau:

* F[1,j]=A[1,j]   
  F[i,j]=max(F[i−1,j],F([i−1,j−1],F[i−1,j+1])+A[i,j] với i>1

**7.3. Cài đặt**

Bảng phương án là bảng 2 chiều F[0..m,0..n]. (*Tất cả các ô trên biên đều cho giá trị bằng 0).*

**Quá trình tính như sau:**

*for i:=1 to m do  
for j := 1 to n do*

*F[i,j]=max(F[i-1,j],F[i-1,j-1],F[i-1,j+1]]+A[i,j]);*

**Cách cài đặt này cho độ phức tạp bộ nhớ và thời gian đều là O(n2). Ta có thể tiết kiệm không gian nhớ bằng cách tính trực tiếp trên mảng A.**

*include<bits/stdc++.h>*

*using namespace std;*

*int i,j,m,n,a[1000][1000],b[1000][1000],ii[1000],jj[1000],k=0;*

*int main()*

*{*

*cin>>m>>n;*

*for(i=1;i<=m;i++)*

*for(j=1;j<=n;j++) cin>>a[i][j];*

*for(i=0;i<=m;i++) b[0][i]=0;*

*for(i=0;i<=m;i++) b[i][0]=0;*

*for(i=1;i<=m;i++)*

*for(j=1;j<=n;j++)*

*b[i][j]=max(max(b[i-1][j],b[i-1][j-1]),b[i][j-1])+a[i][j];*

*for(i=1;i<=m;i++)*

*{*

*for(j=1;j<=n;j++) cout<<b[i][j]<<" ";*

*cout<<endl<<endl;*

*}*

*i=m;j=n; ii[1]=i;*

*jj[1]=j;*

*while(i>1||j>1)*

*{*

*k++;*

*ii[k]=i;*

*jj[k]=j;*

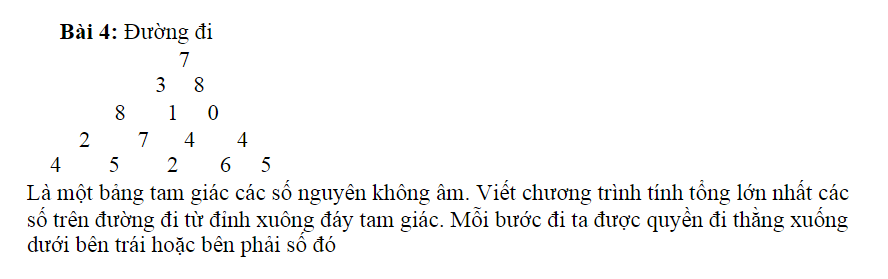
*if(b[i][j]==a[i][j]+b[i-1][j])i--; else if(b[i][j]==a[i][j]+b[i-1][j-1]){i--;j--;}else if(b[i][j]==a[i][j]+b[i][j-1])j--;*

*}*

*cout<<b[m][n]<<endl;*

*cout<<1<<" "<<1<<endl;*

*for(i=k;i>=1;i--) cout<<ii[i]<<" "<<jj[i]<<endl;*

*}*

*#include<bits/stdc++.h>*

*using namespace std;*

*int i,j,m,n,a[1000][1000],b[1000][1000],ii[1000],jj[1000],k=0;*

*int main()*

*{*

*cin>>n;*

*for(i=1;i<=n;i++)*

*for(j=1;j<=i;j++) cin>>a[i][j];*

*//for(i=0;i<=m;i++) b[0][i]=0;*

*//for(i=0;i<=m;i++) b[i][0]=0;*

*b[1][0]=-1;*

*b[1][1]=a[1][1];*

*for(i=1;i<=n;i++)*

*for(j=1;j<=i;j++)*

*b[i][j]=max(b[i-1][j],b[i-1][j-1])+a[i][j];*

*int ans = b[n][1];*

*for(i=2;i<=n;i++) if(ans<b[n][i]) ans=b[n][i];*

*for(i=1;i<=n;i++)*

*{*

*for(j=1;j<=i;j++) cout<<b[i][j]<<" ";*

*cout<<endl<<endl;*

*}*

*cout<<ans;*

*}*