#include<stdio.h>

#define MAX 50

//

struct PS

{

int tu, mau;

};

//

void XuatPS(PS x);

void XuatMangPS(PS A[] ,int n);

int GCD(int a, int b);

PS operator+(PS const &a, PS const &b);

void ChinhDau(PS &x);

void RutGon(PS &x);

void NhapPS(PS &x);

PS operator\*(PS const &a, PS const &b);

PS operator-(PS const &a, PS const &b);

PS operator/(PS const &a, PS const &b);

void NhapMangPS(PS A[], int &n);

//

int main()

{

PS A[MAX];

PS tong,hieu,tich,thuong;

int n;

NhapMangPS(A,n);

tong = A[1]+A[2];

hieu = A[1]-A[2];

tich = A[1]\*A[2];

XuatMangPS(A,n);

XuatPS(tong);

return 0;

}

PS operator\*(PS const &a, PS const &b)

{

PS c;

c.tu = a.tu\*b.tu;

c.mau = a.mau\*b.mau;

return c;

}

PS operator/(PS const &a, PS const &b)

{

PS c;

c.tu = a.tu\*b.mau;

c.mau = a.mau\*b.tu;

return c;

}

PS operator-(PS const &a, PS const &b)

{

PS c;

c.tu = a.tu\*b.mau-a.mau\*b.tu;

c.mau = b.mau\*a.mau;

return c;

}

PS operator+(PS const &a, PS const &b)

{

PS c;

c.tu = a.tu\*b.mau+a.mau\*b.tu;

c.mau = b.mau\*a.mau;

return c;

}

void XuatMangPS(PS A[] ,int n)

{

for(int i =0 ;i<n;i++)

XuatPS(A[i]);

}

void XuatPS(PS x)

{

RutGon(x);

ChinhDau(x);

if(x.tu==0||x.mau==1)

printf("%d\t",x. tu);

else

printf("%d %d\t",x.tu,x.mau);

}

void ChinhDau(PS &x)

{

if(x.mau < 0)

{

x.mau = x.mau\*-1;

x.tu = x.tu\*-1;

}

}

int GCD(int a, int b)

{

if(b==0) return a;

return GCD(b,a%b);

}

void RutGon(PS &x)

{

int gcd = GCD(x.tu,x.mau);

x.tu = x.tu/gcd;

x.mau = x.mau/gcd;

}

void NhapMangPS(PS A[], int &n)

{

do

{

scanf("%d",&n);

}while(n<=0);

for(int i = 0 ; i < n;i++)

NhapPS(A[i]);

}

void NhapPS(PS &x)

{

int tu, mau;

printf("Nhap tu mau: ");

scanf("%d%d",&tu,&mau);

x.tu = tu;

x.mau = mau;

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#include<stdio.h>

//

struct Time

{

int h,m,s;

};

//

bool BgreatthanA(int x, int y, Time a, Time b);

void swap(Time &a, Time &b);

Time operator-(Time a, Time b);

int Minus(int a, int b);

void XuatTime(Time x);

int Minus(int a, int b, int &c);

void Setup(Time &a, Time &b);

void NhapTime(Time \*p);

//

int main()

{

Time t1,t2,t3,\*pt;

pt = &t1;

NhapTime(pt);

pt = &t2;

NhapTime(pt);

t3 = t1-t2;

XuatTime(t3);

return 0;

}

void XuatTime(Time x)

{

printf("%d:%d:%d",x.h,x.m,x.s);

}

Time operator-(Time a, Time b)

{

Time c;

c.m = 0;

c.h = 0;

Setup(a,b);

c.s = Minus(a.s,b.s,c.m);

c.m =c.m+ Minus(a.m,b.m,c.h);

c.h =c.h+ Minus(a.h,b.h);

return c;

}

int Minus(int a, int b)

{

if(a-b<0)

{

return (a+60)-b;

}

return a- b;

}

int Minus(int a, int b, int &c)

{

if(a-b<0)

{

c= -1;

return (a+60)-b;

}

return a- b;

}

void Setup(Time &a, Time &b)

{

if(BgreatthanA(a.h,b.h,a,b))

swap(a,b);

}

void swap(Time &a, Time &b)

{

Time t=a;

a=b;

b=t;

}

bool BgreatthanA(int x, int y, Time a, Time b)

{

int A[3];

A[0] = a.h- b.h;

A[1] = a.m - b.m;

A[2] = a.s - b.s;

for(int i =0; i<3;i++)

if(A[i]!=0)

{

if(A[i]>0)

{

return false;

}

else

{

return true;

}

}

return true;

}

void NhapTime(Time \*p)

{

int h,m,s;

printf("Nhap thoi gian: ");

scanf("%d%d%d",&h,&m,&s);

p->h = h;

p->m = m;

p->s = s;

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#include<stdio.h>

//

void XuatMang(int A[], int n);

void EnterXtoSearch(int &x);

int BinarySearch(int A[], int n, int left, int right, int x);

void XuatKq(int Kq);

void SapXep(int A[], int n);

void NhapMang(int A[], int &n);

void swap(int &a, int &b);

//

int main()

{

int A[50], n,x;

NhapMang(A,n);

SapXep(A,n);

XuatMang(A,n);

EnterXtoSearch(x);

int Kq = BinarySearch(A,n,0,n-1,x);

XuatKq(Kq);

}

{

void EnterXtoSearch(int &x)

printf("Enter number to search: ");

scanf("%d",&x);

}

void XuatKq(int Kq)

{

if(Kq==-1)

printf("Number is not in array");

else

printf("%d",Kq);

}

int BinarySearch(int A[], int n, int left, int right, int x)

{

if(left>right)

return -1;

else

{

int mid = (left+right)/2;

if(A[mid]==x)

return mid;

if(A[mid]>x)

return BinarySearch(A,n,left,mid-1,x);

else

return BinarySearch(A,n,mid+1,right,x);

}

}

void NhapMang(int A[], int &n)

{

do

{

printf("Enter length of array: ");

scanf("%d",&n);

}while(n<=0);

printf("Enter element: ");

for(int i=0;i<n;i++)

scanf("%d",&A[i]);

}

void SapXep(int A[], int n)

{

for(int i=0;i<n-1;i++)

for(int j=i;j<n;j++)

if(A[i]>A[j])

swap(A[i],A[j]);

}

void swap(int &a, int &b)

{

int t;

t=a;

a=b;

b=t;

}

void XuatMang(int A[], int n)

{

for(int i = 0;i<n;i++)

printf("%d ",A[i]);

printf("\n");

}

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#include<stdio.h>

#include<iostream>

using namespace std;

//

void Nhap(int &n);

void ConvertTo2(int n);

//

int main()

{

int n;

Nhap(n);

ConvertTo2(n);

}

void Nhap(int &n)

{

printf("Enter number: ");

scanf("%d",&n);

}

void ConvertTo2(int n)

{

for(int i =31;i>=0;i--)

cout<<((n>>i)&1);

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#include<stdio.h>

//

void Nhap(int &n);

float TinhS(int n );

int TinhSadd(int n);

//

int main()

{

int n;

Nhap(n);

float Kq = TinhS(n);

printf("%.2f",Kq);

}

float TinhS(int n )

{

if(n==1) return 1;

return float(1)/TinhSadd(n)+TinhS(n-1);

}

int TinhSadd(int n )

{

if(n==1) return 1;

return n+TinhSadd(n-1);

}

void Nhap(int &n)

{

printf("Enter number: ");

scanf("%d",&n);

}

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#include<stdio.h>

//

void SapXep(int A[]);

void swap(int &a, int &b);

void Sort(int A[], int l, int r, int mid);

void Xuat(int A[]);

//

int main()

{

int A[3] = {3,2,1};

SapXep(A);

Xuat(A);

}

void Xuat(int A[])

{

for(int i = 0; i< 3;i++)

printf("%d",A[i]);

}

void SapXep(int A[])

{

Sort(A,0,2,1);

}

void Sort(int A[], int l, int r,int mid)

{

if(A[l]>A[r])

{

if(A[mid]>A[l])

{

swap(A[l],A[mid]);

return Sort(A,l,r,mid);

}

if(A[mid]<A[r])

{

swap(A[r],A[mid]);

return Sort(A,l,r,mid);

}

swap(A[r],A[l]);

}

else

{

if(A[mid]>A[r])

swap(A[mid],A[r]);

if(A[mid]<A[l])

swap(A[mid],A[l]);

}

}

void swap(int &a, int &b)

{

int t = a;

a = b;

b = t;

}

///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#include<stdio.h>

#include<string.h>

#define MAX 200

//

void Dao(char S[]);

void Add(char S[], char C[], int pos, int &count, int &k);

void Tach(char S[]);

void swap(char &a, char &b);

//

int main()

{

char S[MAX];

fgets(S,200,stdin);

Tach(S);

}

void Xuat(char X[])

{

printf("%s",X);

}

void Tach(char S[])

{

char C[MAX];

int k =0;

for(int i = 0;i<strlen(S);i++)

{

if(S[i]!=' ')

{

C[k++] = S[i];

}

else

{

if(k>0)

{

C[k] = '\0';

Dao(C);

Xuat(C);

k = 0;

}

printf("%c",S[i]);

}

}

C[k-1] = '\0'; //vi dung fgets

Dao(C);

Xuat(C);

}

void Add(char S[], char C[], int pos, int &count, int &k)

{

for(int i = 0;i<=count;i++)

{

C[k++] = S[pos-count];

}

C[k++] = ' ';

count = 0;

}

void Dao(char S[])

{

float mid = float(strlen(S)-1)/2;

for(int i = 0; i< mid ;i++)

{

swap(S[i],S[strlen(S)-1-i]);

}

}

void swap(char &a, char &b)

{

char c = a;

a = b;

b = c;

}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#include<stdio.h>

//

int Fn(int n);

int TimN(int M);

//

//

int main()

{

int M;

scanf("%d",&M);

int Kq = TimN(M);

printf("%d",Kq);

}

int TimN(int M)

{

for(int i = 1; i<M;i++)

{

if(Fn(i)>M)

return i-1;

}

}

int Fn(int n)

{

switch(n)

{

case 0:

return 1;

case 1:

return 2;

case 2:

return 3;

}

return Fn(n-3)+2\*Fn(n-2)+3\*Fn(n-1);

}

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#include<stdio.h>

#include<string.h>

#define MAX 101

//

//

int Max(int L[], int n);

void Setup(int L[], int len);

int Duyet(char S[], int L[], int len);

int XuLy(char S[]);

//

int main()

{

char S[MAX];

fgets(S,MAX,stdin);

S[strlen(S)-1] = '\0';

int Kq = XuLy(S);

printf("%d",Kq);

}

int XuLy(char S[])

{

const int len = strlen(S);

int L[len+1];

Setup(L,len+1);

for(int i = 0 ;i< len; i++)

{

if(S[i]=='(')

L[i+1] = L[i] +1;

if(S[i]==')')

L[i+1] = L[i] -1;

}

if(L[len]!=0||S[len-1]=='(')

return -1;

else

return Max(L,len+1);

}

int Duyet(char S[], int L[], int len)

{

for(int i = 1;i<len;i++)

{

if(S[i]=='(')

L[i] = L[i-1] + 1;

else

L[i] = L[i-1] - 1;

}

if(L[len-1]!=0&&len>=2)

return -1;

else

return Max(L,len);

}

void Setup(int L[], int len)

{

for(int i = 0 ;i<len ;i++)

L[i] = 0;

}

int Max(int L[], int n)

{

int max = L[0];

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

if(L[i]>max)

max = L[i];

return max;

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#include<stdio.h>

#define MAX 101

//

void NhapMang2(int A[][MAX], int &n, int &m);

int XuLy(int A[][MAX], int n, int m);

int Tong(int A[][MAX], int y, int x);

void Setup(int A[][MAX], int n, int m);

//

int main()

{

int A[MAX][MAX], n, m;

NhapMang2(A,n,m);

Setup(A,n,m);

int Kq = XuLy(A,n,m);

printf("%d",Kq);

}

int XuLy(int A[][MAX], int n, int m)

{

int maxS = 0;

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

for(int j = 1;j<=m;j++)

{

int tong = Tong(A,i,j);

if(tong>maxS)

maxS = tong;

}

return maxS;

}

int Tong(int A[][MAX], int y, int x)

{

int S = A[y][x];

int Y[8] ={-1,-1,-1,0,1,1,1,0};

int X[8] ={-1,0,1,1,1,0,-1,-1};

for(int i = 0 ;i<8;i++)

S = S + A[y+Y[i]][x+X[i]];

return S;

}

void NhapMang2(int A[][MAX], int &n, int &m)

{

scanf("%d%d",&n,&m);

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

for(int j = 1;j<=m;j++)

scanf("%d",&A[i][j]);

}

void Setup(int A[][MAX], int n, int m)

{

for(int i = 0;i<=n+1;i++)

{

A[i][0] = 0;

A[i][m+1] = 0;

}

for(int j = 1; j<=m;j++)

{

A[0][j] = 0;

A[n+1][j] = 0;

}

}

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#include<stdio.h>

#define MAX 101

//

void NhapMang(int A[], int &n);

void Setup(int L[], int len);

void Duyet(int A[], int L[], int pos, char c);

void Duyet(int A[], int L[], int pos);

int Max(int L[], int len);

int XuLy(int A[], int const n);

//

//

int main()

{

int A[MAX], n;

NhapMang(A,n);

int Kq = XuLy(A,n);

printf("%d",Kq);

}

int XuLy(int A[], int const n)

{

int Lup[n+1], Ldown[n+1];

Setup(Lup,n+1);

Setup(Ldown,n+1);

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

{

Duyet(A,Lup,i,'c');

Duyet(A,Ldown,i);

}

int maxup = Max(Lup,n+1);

int maxdown = Max(Ldown,n+1);

return maxup>=maxdown?maxup:maxdown;

}

int Max(int L[], int len)

{

int max = L[0];

for(int i = 1 ;i<len;i++)

if(L[i]>max)

max =L[i];

return max;

}

void Duyet(int A[], int L[], int pos)

{

int lmax = 0;

for(int i =1 ;i<pos;i++)

{

if(A[pos]<=A[i])

if(L[i]>lmax)

lmax = L[i];

}

L[pos] = lmax +1;

}

void Duyet(int A[], int L[], int pos, char c)

{

int lmax = 0;

for(int i = 0 ; i<pos;i++)

{

if(A[pos]>=A[i])

if(L[i]>lmax)

lmax = L[i];

}

L[pos] = lmax + 1;

}

void Setup(int L[], int len)

{

for(int i = 0 ;i<len;i++)

L[i] = 0;

}

void NhapMang(int A[], int &n)

{

scanf("%d",&n);

A[0] = 0;

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

scanf("%d",&A[i]);

}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#include<stdio.h>

#define MAX 101

//

int DemMin(int A[][MAX], int y, int x);

void NhapMang2(int A[][MAX], int &n, int &m);

void XuatMang2(int L[][MAX], int n, int m);

void XuLy(int A[][MAX], int n, int m);

void Setup(int A[][MAX], int n, int m);

//

int main()

{

int A[MAX][MAX], n, m;

NhapMang2(A,n,m);

Setup(A,n,m);

XuLy(A,n,m);

}

void Setup(int A[][MAX], int n, int m)

{

for(int i = 0;i<=n+1;i++)

{

A[i][0] = 0;

A[i][m+1] = 0;

}

for(int j = 1; j<=m;j++)

{

A[0][j] = 0;

A[n+1][j] = 0;

}

}

void XuLy(int A[][MAX], int n, int m)

{

int L[MAX][MAX];

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

for(int j =1 ;j<=m;j++)

if(A[i][j] == 0)

L[i][j] = DemMin(A,i,j);

else

L[i][j] = 9;

XuatMang2(L,n,m);

}

void XuatMang2(int L[][MAX], int n, int m)

{

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

{

for(int j =1;j<=m;j++)

{

printf("%d ",L[i][j]);

}

printf("\n");

}

}

int DemMin(int A[][MAX], int y, int x)

{

int Y[8] ={-1,-1,-1,0,1,1,1,0};

int X[8] ={-1,0,1,1,1,0,-1,-1};

int count = 0;

for(int i =0;i<8;i++)

{

if(A[y+Y[i]][x+X[i]]==1)

count++;

}

return count;

}

void NhapMang2(int A[][MAX], int &n, int &m)

{

scanf("%d%d",&n,&m);

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

for(int j = 1;j<=m;j++)

scanf("%d",&A[i][j]);

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#include<stdio.h>

int main()

{

// nhap vao n

int n;

printf("Nhap vao n : ");

scanf("%d", &n);

printf("\nCac bo hoan vi la :\n");

// khoi tao mang gom n phan tu

int mang[n];

// gan gia tri cho cac phan tu cua mang va in ra bo hoan vi dau tien

int i;

for(i = 0; i < n; i++)

{

mang[i] = i + 1;

printf("%d ", mang[i]);

}

// xu ly de in ra cac bo hoan vi tiep theo

for(i = n - 1; i > 0; i--)

{

// neu gap mang[i] > mang[i - 1]

if(mang[i] > mang[i - 1])

{

int j;

// tim trong bo cuoi giam dan phan tu vua du lon hon mang[i - 1] sau do swap.

for(j = n - 1; j >= i; j--)

{

if(mang[j] > mang[i - 1])

{

int temp = mang[j];

mang[j] = mang[i - 1];

mang[i - 1] = temp;

break;

}

}

// dao nguoc bo cuoi

for(j = n - 1; j > ((n - 1 + i) / 2); j--)

{

int temp = mang[i + n - 1 - j];

mang[i + n - 1 - j] = mang[j];

mang[j] = temp;

}

// in ra bo hoan vi moi

printf("\n");

for(j = 0; j < n; j++)

{

printf("%d ", mang[j]);

}

// i = n de khi thoat vong lap hien tai i-- thanh i = n - 1; va vong lap chay lai tu dau.

i = n;

}

}

return 0;

}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#include<stdio.h>

//

void ChuyenThanhO(int A[], int n, int pos);

void Xuat(int A[], int n);

void Setup(int A[], int n);

void Xuly(int n);

//

int main()

{

int n;

scanf("%d",&n);

Xuly(n);

}

void Xuly(int n)

{

int A[n];

Setup(A,n);

Xuat(A,n);

for(int i = n-1;i>=0;i--)

{

if(A[i]==0)

{

A[i] = 1;

ChuyenThanhO(A,n,i);

Xuat(A,n);

i=n;

}

}

}

void ChuyenThanhO(int A[], int n, int pos)

{

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

A[i]=0;

}

void Xuat(int A[], int n)

{

for(int i =0;i<n;i++)

printf("%d",A[i]);

printf("\n");

}

void Setup(int A[], int n)

{

for(int i = 0;i<n;i++)

A[i]=0;

}

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#include<stdio.h>

#define MAX 100

//

void ChuyenThanhO(int A[], int n, int pos);

void Nhap(int A[], int &n);

void Xuattapcon(int A[], int B[], int n);

void Xuat(int A[], int n);

void Setup(int A[], int n);

void Xuly(int A[], int n);

//

int main()

{

int A[MAX], n;

Nhap(A,n);

Xuly(A,n);

}

void Nhap(int A[], int &n)

{

scanf("%d",&n);

for(int i = 0;i<n;i++)

scanf("%d",&A[i]);

}

void Xuly(int A[], int n)

{

int B[n];

Setup(B,n);

Xuat(B,n);

for(int i = n-1;i>=0;i--)

{

if(B[i]==0)

{

B[i] = 1;

ChuyenThanhO(B,n,i);

Xuattapcon(A,B,n);

i=n;

}

}

}

void Xuattapcon(int A[], int B[], int n)

{

for(int i = 0;i<n;i++)

if(B[i]==1)

printf("%d",A[i]);

printf("\n");

}

void ChuyenThanhO(int A[], int n, int pos)

{

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

A[i]=0;

}

void Xuat(int A[], int n)

{

for(int i =0;i<n;i++)

printf("%d",A[i]);

printf("\n");

}

void Setup(int A[], int n)

{

for(int i = 0;i<n;i++)

A[i]=0;

}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#include<stdio.h>

int main()

{

// nhap n va k

int n;

printf("Nhap vao n : ");

scanf("%d", &n);

int k;

printf("\nNhap vao k: ");

scanf("%d", &k);

// khoi tao mang k phan tu

int mang[k];

printf("\nTap con liet ke duoc :\n");

// in ra tap con dau tien

int i;

for(i = 0; i < k; i++)

{

mang[i] = i + 1; // khoi tao mang[i] = i + 1;

printf("%d ", mang[i]);

}

printf("\n");

// xu ly

for(i = k - 1; i >= 0; i--)

{

if(mang[i] < n - k + i + 1) // neu phan tu vi tri i nho hon gia tri max tai vi tri nay

{

mang[i]++; // tang len 1 don vi

int j;

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

{

mang[j] = mang[j-1] + 1; // gan lai gia tri cho cac phan tu phia sau

}

// in ra bo so moi tao duoc

for(j = 0; j < k; j++)

{

printf("%d ", mang[j]);

}

printf("\n");

i = k; // gan i = k de khi het vong lap i-- nen i se = k - 1, tu do chay lai tu vi tri cuoi.

// gan i = k - 1 la sai vi khi het vong lap hien tai i-- se thanh k - 2.

}

}

return 0;

}