**• Set Theory**

1. **Start with a NULL set and add elements one-by-one: Use different ways of implementing sets and understand the pros and cons of each of these methods**
2. **Given an element value, check whether it is a member of the set or not**
3. **Find out the number of elements of a given set.**
4. **Complement of a set; Union, Intersection.**
5. **Test whether a given set X is a subset of the set A or not.**
6. **Test whether two given sets are equal or not.**
7. **Difference and Symmetric Difference of two sets.**

#include<stdio.h>

void addelement()

{

int a[5],i,n;

printf("Enter Array size:");

scanf("%d",&n);

printf("Enter Values: \n");

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

{

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

}

printf("Set Element :\n");

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

{

printf("%d\n",a[i]);

}

}

void findelement()

{

int a[5],i,n,r,c=0;

printf("Enter Array size:");

scanf("%d",&n);

printf("Enter Values: \n");

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

{

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

}

printf("Enter Values which u want to find: \n");

scanf("%d",&r);

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

{

if(r==a[i])

{

c++;

}

}

if(c==1)

{

printf("it is a member");

}

else

{

printf("it is Not a member");

}

}

void union\_set()

{

int a[15],b[15],c[30],i,j,k=0,n,m;

printf("Enter Set size:\n");

scanf("%d",&n);

printf("Enter Set A\n");

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

{

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

c[k]=a[i];

k++;

}

printf("Enter Set size:\n");

scanf("%d",&m);

printf("Enter Set B\n");

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

{

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

}

printf("Set A is: { ");

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

{

if(i==(n-1))

{

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

}

else

{

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

}

}

printf("}\n");

printf("Set B is: { ");

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

{

if(i==(m-1))

{

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

}

else

{

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

}

}

printf("}\n");

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

{

int flag=0;

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

{

if(b[i]==c[j])

{

flag=1;

break;

}

}

if(flag==0)

{

c[k]=b[i];

k++;

}

}

printf("Union set C: { ");

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

{

if(i==(k-1))

{

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

}

else

{

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

}

}

printf("}");

}

void intersect\_set()

{

int a[10],b[10],c[10],i,j,k,n,m,p;

printf("Enter Set size:\n");

scanf("%d",&n);

printf("Enter Array A\n");

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

{

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

}

printf("Enter Set size:\n");

scanf("%d",&m);

printf("\nEnter Array B\n");

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

{

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

}

printf("\nArray A:{ ");

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

{

if(i==(n-1))

{

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

}

else

{

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

}

}

printf("}");

printf("\nArray B:{ ");

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

{

if(i==(m-1))

{

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

}

else

{

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

}

}

printf("}");

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

{

int flag=0;

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

{

if(a[i]==b[j])

{

flag=1;

break;

}

}

if(flag)

{

c[k]=a[i];

k++;

}

}

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

{

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

{

if(c[i]==c[j])

{

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

{

c[p]=c[p+1];

}

k--;

}

}

}

if(k==0)

{

printf("\nThere are No common elements:\n");

}

else

{

printf("\nIntersect Two sets Elements are { ");

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

{

if(i==(k-1))

{

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

}

else

{

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

}

}

printf("}");

}

}

void subset()

{

int x[10],a[10],i,j,f,c=0,n,m;

printf("Enter Set size:\n");

scanf("%d",&n);

printf("Enter Set X:\n");

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

{

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

}

printf("Enter Set size:\n");

scanf("%d",&m);

printf("Enter Set A:\n");

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

{

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

}

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

{

f=0;

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

{

if(x[i]==a[j])

{

f=1;

break;

}

}

if(f==0)

{

c++;

}

}

if(c==0)

{

printf("X is subset of A");

}

else

{

printf("X is not Subset of A");

}

}

void equalset()

{

int a[10],b[10],i,j,f,c=0,n,m,l=0;

printf("Enter Set size:\n");

scanf("%d",&n);

printf("Enter Set A:\n");

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

{

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

}

printf("Enter Set size:\n");

scanf("%d",&m);

printf("Enter Set B:\n");

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

{

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

}

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

{

f=0;

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

{

if(a[i]==b[j])

{

f=1;

break;

}

}

if(f==0)

{

c++;

}

}

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

{

f=0;

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

{

if(b[i]==a[j])

{

f=1;

break;

}

}

if(f==0)

{

l++;

}

}

if(c==0 && l==0)

{

printf("Equal Set");

}

else

{

printf("Not Equal Set");

}

}

void Difference()

{

int a[10],b[10],c[10],d[10],i,j,k=0,e=0,n,m;

printf("Enter Set Size:");

scanf("%d",&n);

printf("Enter Set A:\n");

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

{

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

}

printf("Enter Set Size:");

scanf("%d",&m);

printf("Enter Set B:\n");

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

{

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

}

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

{

int f=0;

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

{

if(a[i]==b[j])

{

f=1;

break;

}

}

if(f==0)

{

c[k]=a[i];

k++;

}

}

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

{

int f=0;

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

{

if(b[i]==a[j])

{

f=1;

break;

}

}

if(f==0)

{

d[e]=b[i];

e++;

}

}

printf("A-B: { ");

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

{

if(i==(k-1))

{

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

}

else

{

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

}

}

printf(" }");

printf("\nB-A: { ");

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

{

if(i==(e-1))

{

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

}

else

{

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

}

}

printf(" }");

}

void Symmetric\_Difference()

{

int a[10],b[10],c[10],d[10],z[10],i,j,k=0,e=0,y=0,n,m;

printf("Enter Set Size:");

scanf("%d",&n);

printf("Enter Set A:\n");

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

{

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

}

printf("Enter Set Size:");

scanf("%d",&m);

printf("Enter Set B:\n");

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

{

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

}

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

{

int f=0;

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

{

if(a[i]==b[j])

{

f=1;

break;

}

}

if(f==0)

{

c[k]=a[i];

z[y]=c[k];

y++;

k++;

}

}

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

{

int f=0;

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

{

if(b[i]==a[j])

{

f=1;

break;

}

}

if(f==0)

{

d[e]=b[i];

z[y]=d[e];

y++;

e++;

}

}

printf("A-B: { ");

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

{

if(i==(k-1))

{

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

}

else

{

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

}

}

printf(" }");

printf("\nB-A: { ");

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

{

if(i==(e-1))

{

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

}

else

{

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

}

}

printf(" }");

printf("\nA-B U B-A: { ");

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

{

if(i==(y-1))

{

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

}

else

{

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

}

}

printf(" }");

}

void main()

{

while("true")

{

int n,ch;

char ch2;

printf("\n");

printf("(1) Start with a NULL set and add elements one-by-one:\n");

printf("(2) Given an element value, check whether it is a member of the set or not\n");

printf("(3) Find out the number of elements of a given set.\n");

printf("(4) Complement of a set; Union, Intersection.\n");

printf("(5) Test whether a given set X is a subset of the set A or not.\n");

printf("(6) Test whether two given sets are equal or not.\n");

printf("(7) Difference and Symmetric Difference of two sets.\n");

printf("\n");

printf("Enter your choices: ");

scanf("%d",&n);

if(n==1)

{

addelement();

}

else if(n==2)

{

findelement();

}

else if(n==3)

{

printf("In progress...");

}

else if(n==4)

{

while("true")

{

printf("\n");

printf("1.Union\n");

printf("2.Intersection\n");

printf("Enter Your choice..!\n");

scanf("%d",&ch);

switch(ch)

{

case 1:

union\_set();

break;

case 2:

intersect\_set();

break;

default:

printf("valid choice..!");

}

printf("\nDo u want to continues This two Operation..? (Press Y or y): ");

ch2=getch();

if(ch2=='y' || ch2=='Y')

{

continue;

}

else

{

break;

}

}

}

else if(n==5)

{

subset();

}

else if(n==6)

{

equalset();

}

else if(n==7)

{

while("true")

{

printf("\n");

printf("1.Difference\n");

printf("2.Symmetric Difference\n");

printf("Enter Your choice..!\n");

scanf("%d",&ch);

switch(ch)

{

case 1:

Difference();

break;

case 2:

Symmetric\_Difference();

break;

default:

printf("valid choice..!");

}

printf("\nDo u want to continues This two Operation..? (Press Y or y): ");

ch2=getch();

if(ch2=='y' || ch2=='Y')

{

continue;

}

else

{

break;

}

}

}

else

{

printf("Invalid..!\n");

}

printf("\nDo u want to continues..? (Press Y or y): ");

ch2=getch();

if(ch2=='y' || ch2=='Y')

{

continue;

}

else

{

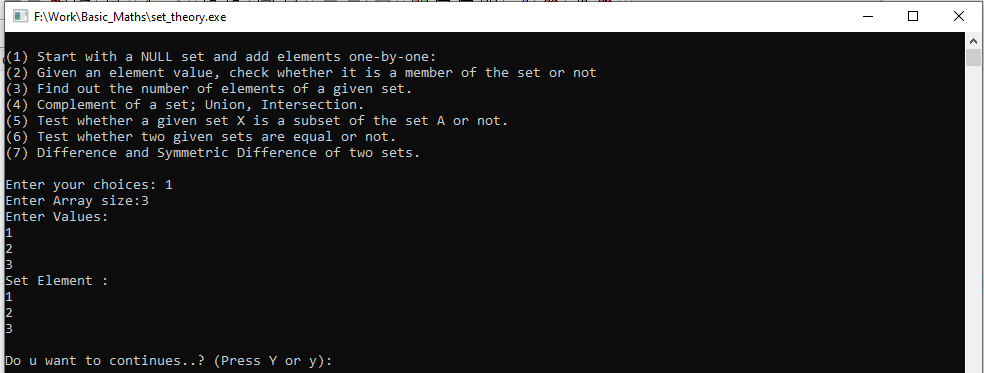
break;

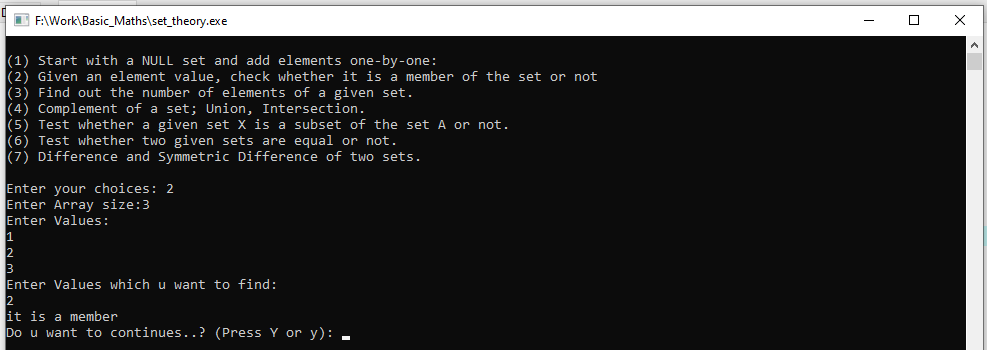
}

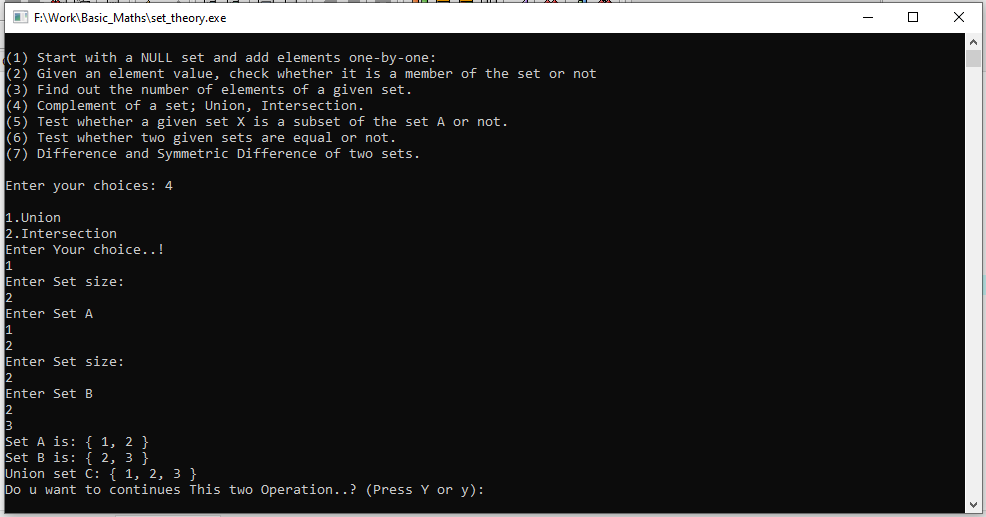
}

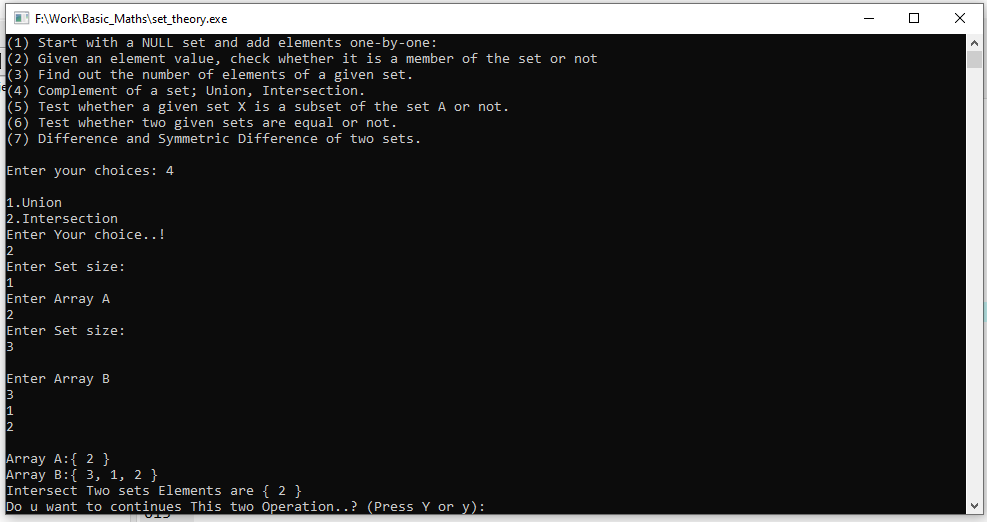
}

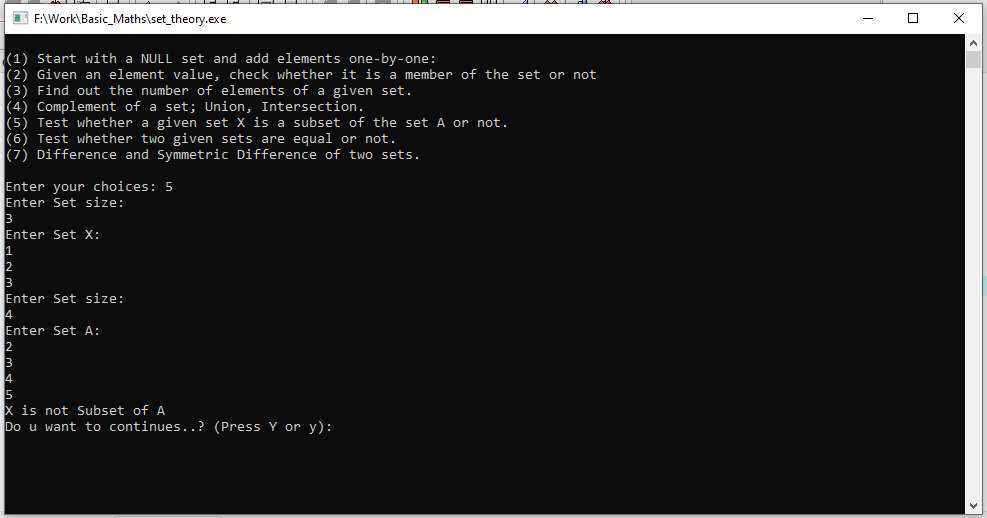
**OUTPUT:-**

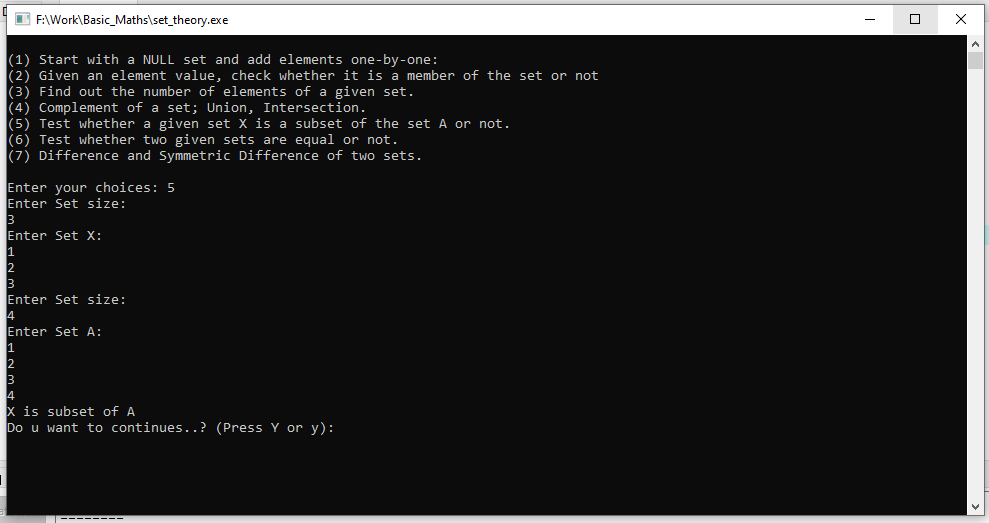


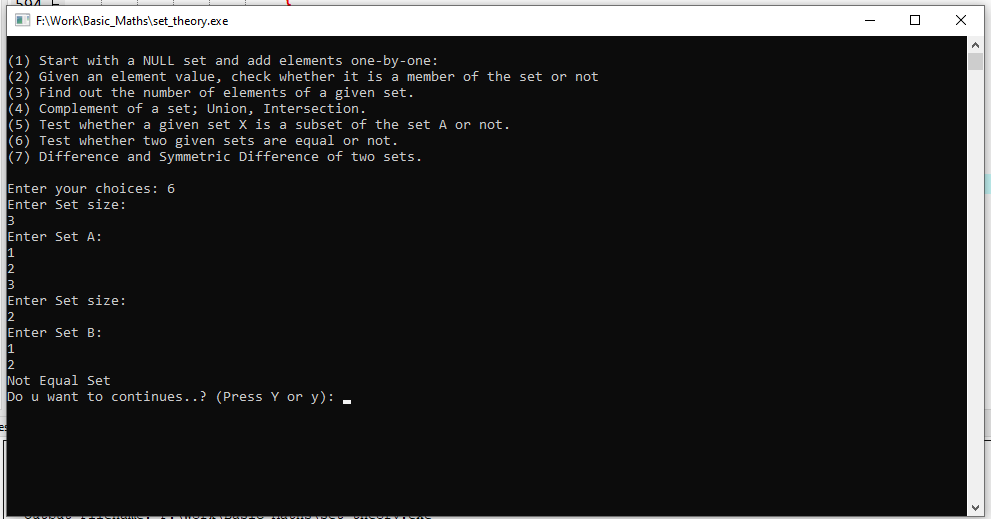


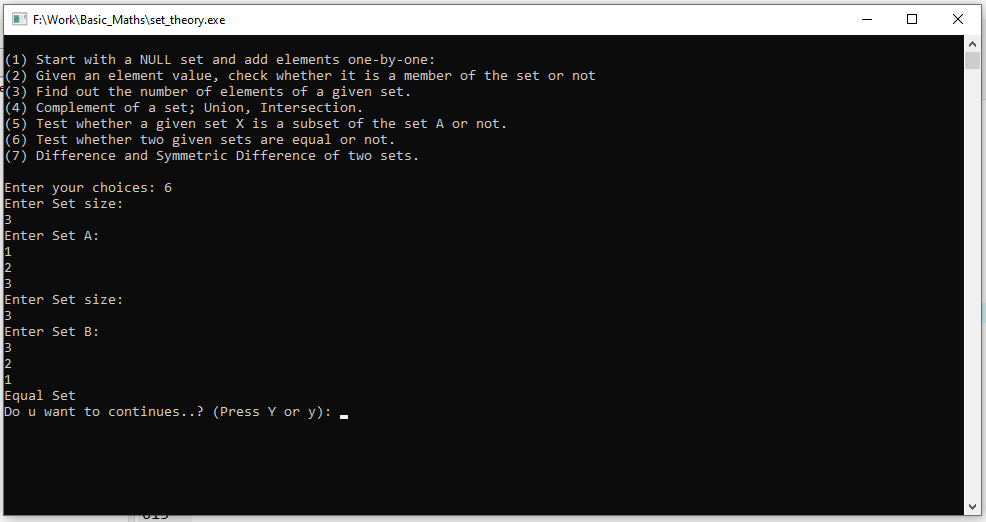


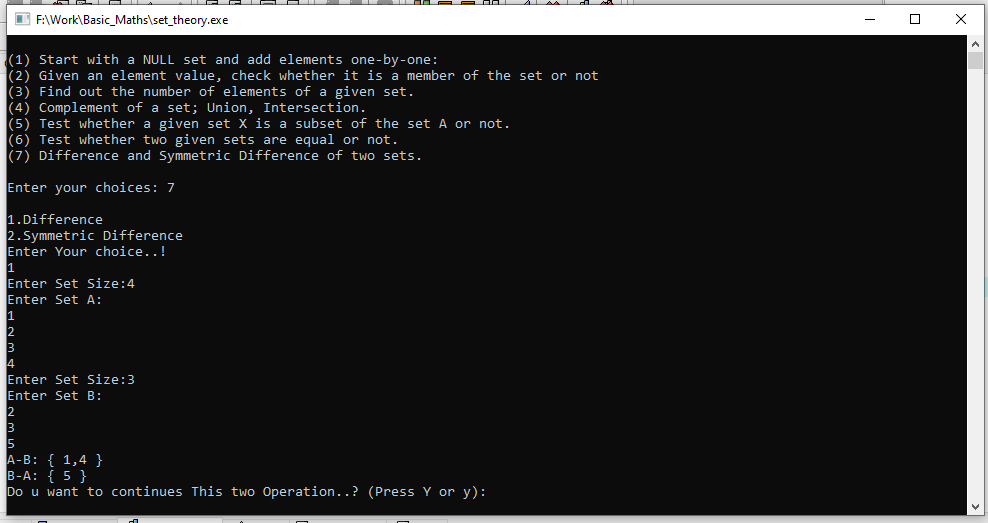


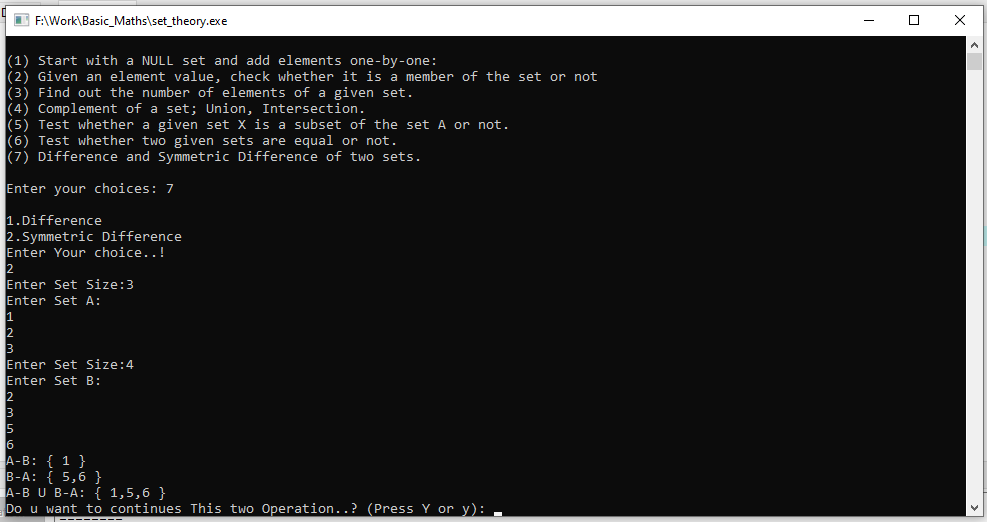












**Matrices:**

1. **Write a function to Create a Matrix of size m x n, and another function to Print a**

**Matrix of size m x n.**

**2. Create 2 matrices A and B of size (m x n). Find (A + B) and (A – B).**

**3. Find Transpose of matrix A and of matrix B. Find (AT + BT) and (AT– BT).**

**Check whether (AT + BT) = (A + B)Tand (AT– BT) = (A – B)T.**

**4. Add matrix A and null matrix. Subtract null matrix from matrix A.**

**5. Create a unit matrix of size (n x n) and a unit matrix of size (n x n).**

**Multiply matrix A with a unit matrix.**

**6. Create matrix A of size (m x n) and matrix B of size (n x p). Multiply matrix A and**

**matrix B to get matrix C of size (m x p).**

**7. Create a symmetric matrix A. Find matrix AT. Check whether A = AT?**

**8. Evaluate Scalar Product of a Matrix A: For example, k A, where k is a constant (number)**

**9. Take as input two matrices, A & B and print (A \* B) and (B \* A). First check**

**which ones out of (A \* B) and (A \* B) are possible to compute.**

#include<stdio.h>

#include<conio.h>

void matrix\_display()

{

int i,j,a[10][10];

printf("Enter Matirx: \n");

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

{

for(j=0;j<3;j++)

{

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

}

}

printf("Matrix is:\n");

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

{

for(j=0;j<3;j++)

{

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

}

printf("\n");

}

}

void Matrixab()

{

int i,j,m,n;

int a[10][10],b[10][10],c[10][10];

printf("Enter Row Size:");

scanf("%d",&m);

printf("Enter Col Size:");

scanf("%d",&n);

printf("Enter Matirx A\n");

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

{

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

{

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

}

}

printf("Enter Matirx B\n");

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

{

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

{

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

c[i][j]=a[i][j]+b[i][j];

}

}

printf("Matirx A\n");

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

{

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

{

printf("%d\t",a[i][j]);

}

printf("\n");

}

printf("Matirx B\n");

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

{

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

{

printf("%d\t",b[i][j]);

}

printf("\n");

}

printf("(A + B) Matrix is:\n");

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

{

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

{

printf("%d\t",c[i][j]);

}

printf("\n");

}

printf("(A - B) Matrix is:\n");

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

{

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

{

c[i][j]=a[i][j]-b[i][j];

printf("%d\t",c[i][j]);

}

printf("\n");

}

}

void matrixtras()

{

int i,j,m,n;

int a[10][10],b[10][10],c[10][10];

printf("Enter Row Size:");

scanf("%d",&m);

printf("Enter Col Size:");

scanf("%d",&n);

printf("Enter Matirx A\n");

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

{

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

{

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

}

}

printf("Enter Matirx B\n");

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

{

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

{

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

}

}

printf("Matirx A\n");

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

{

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

{

printf("%d\t",a[i][j]);

}

printf("\n");

}

printf("Transpose of Matirx A\n");

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

{

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

{

printf("%d\t",a[j][i]);

}

printf("\n");

}

printf("\n========================================\n");

printf("Matirx B\n");

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

{

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

{

printf("%d\t",b[i][j]);

}

printf("\n");

}

printf("Transpose of Matirx B\n");

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

{

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

{

printf("%d\t",b[j][i]);

c[i][j]=a[j][i]+b[j][i];

}

printf("\n");

}

printf("(AT + BT) Matrix is:\n");

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

{

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

{

printf("%d\t",c[i][j]);

}

printf("\n");

}

printf("(AT - BT) Matrix is:\n");

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

{

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

{

c[i][j]=a[j][i]-b[j][i];

}

}

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

{

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

{

printf("%d\t",c[i][j]);

}

printf("\n");

}

printf("\n========================================\n");

printf("(A + B)T Matrix is:\n");

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

{

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

{

c[i][j]=a[i][j]+b[i][j];

}

}

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

{

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

{

printf("%d\t",c[j][i]);

}

printf("\n");

}

printf("\n========================================\n");

printf("(A - B)T Matrix is:\n");

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

{

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

{

c[i][j]=a[i][j]-b[i][j];

}

}

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

{

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

{

printf("%d\t",c[j][i]);

}

printf("\n");

}

}

void nullsubmatrix()

{

int i,j,m,n,k;

int a[10][10],b[10][10]={{0,0,0},{0,0,0},{0,0,0}};

int c[10][10];

printf("Enter Row Size:");

scanf("%d",&m);

printf("Enter Col Size:");

scanf("%d",&n);

printf("Enter Matirx A\n"); //Input A

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

{

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

{

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

}

}

printf("Matirx A\n");//Print A

printf("\n");

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

{

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

{

printf("%d\t",a[i][j]);

}

printf("\n");

}

printf("Subtract null matrix from matrix A\n");

printf("\n");

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

{

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

{

c[i][j]=a[i][j]-b[i][j];

printf("%d\t",c[i][j]);

}

printf("\n");

}

}

void matrix5()

{

int i,j,m,n,k;

int a[10][10],b[10][10],c[10][10];

printf("Enter Row Size:");

scanf("%d",&m);

printf("Enter Col Size:");

scanf("%d",&n);

printf("Enter Matirx A\n"); //Input A

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

{

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

{

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

}

}

printf("Enter Matirx B\n");//Input B

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

{

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

{

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

}

}

printf("Matirx A\n");//Print A

printf("\n");

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

{

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

{

printf("%d\t",a[i][j]);

}

printf("\n");

}

printf("Matirx B\n");//Print B

printf("\n");

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

{

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

{

printf("%d\t",b[i][j]);

}

printf("\n");

}

printf("Matirx A X B\n");

printf("\n");

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

{

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

{

c[i][j]=0;

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

{

c[i][j]+=a[i][k]\*b[k][j];

}

printf("%d\t",c[i][j]);

}

printf("\n");

}

}

void mulmatrix()

{

int i,j,m,n,k;

int a[10][10],b[10][10],c[10][10];

printf("Enter Row Size:");

scanf("%d",&m);

printf("Enter Col Size:");

scanf("%d",&n);

printf("Enter Matirx A\n"); //Input A

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

{

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

{

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

}

}

printf("Enter Matirx B\n");//Input B

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

{

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

{

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

}

}

printf("Matirx A\n");//Print A

printf("\n");

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

{

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

{

printf("%d\t",a[i][j]);

}

printf("\n");

}

printf("Matirx B\n");//Print A

printf("\n");

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

{

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

{

printf("%d\t",b[i][j]);

}

printf("\n");

}

printf("Matirx A X B\n");

printf("\n");

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

{

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

{

c[i][j]=0;

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

{

c[i][j]+=a[i][k]\*b[k][j];

}

printf("%d\t",c[i][j]);

}

printf("\n");

}

}

void matrix9()

{

int a[20][20],b[20][20],c[30][30],d[30][30],e=0;;

int i,j,n,m,k=0;

printf("Enter Rows :");

scanf("%d",&n);

printf("Enter Colunm :");

scanf("%d",&m);

printf("Enter A Matrix:\n");

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

{

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

{

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

}

}

printf("A Matrix:\n");

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

{

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

{

printf("%d \t",a[i][j]);

}

printf("\n");

}

printf("Enter B Matrix:\n");

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

{

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

{

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

}

}

printf("B Matrix:\n");

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

{

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

{

printf("%d \t",b[i][j]);

}

printf("\n");

}

printf("(A \* B) Multiply:\n");

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

{

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

{

c[i][j]=0;

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

{

c[i][j]+=a[i][k]\*b[k][j];

}

}

}

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

{

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

{

printf("%d \t",c[i][j]);

}

printf("\n");

}

printf("(B \* A) Multiply:\n");

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

{

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

{

d[i][j]=0;

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

{

d[i][j]+=b[i][k]\*a[k][j];

}

}

}

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

{

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

{

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

}

printf("\n");

}

printf("check which ones out of (A \* B) and (B \* A) are possible to compute :\n");

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

{

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

{

if(c[i][j]!=d[i][j])

{

e++;

}

}

}

if(e==0)

{

printf("possible to compute \n");

}

else

{

printf("Not possible to compute \n");

}

}

void matrix7()

{

int a[20][20],e=0;;

int i,j,n,m;

printf("Enter Rows :");

scanf("%d",&n);

printf("Enter Colunm :");

scanf("%d",&m);

printf("Enter A Matrix:\n");

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

{

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

{

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

}

}

printf("A Matrix:\n");

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

{

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

{

printf("%d \t",a[i][j]);

}

printf("\n");

}

printf("Transpose Matrix A: \n");

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

{

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

{

printf("%d \t",a[j][i]);

}

printf("\n");

}

printf("A = AT ? :\n");

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

{

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

{

if(a[i][j]!=a[j][i])

{

e++;

}

}

}

if(e==0)

{

printf("Matrix is A = AT \n");

}

else

{

printf("Matrix is Not A = AT \n");

}

}

void matrix8()

{

int a[20][20],k=2;

int i,j,n,m;

printf("Enter Rows :");

scanf("%d",&n);

printf("Enter Colunm :");

scanf("%d",&m);

printf("Enter A Matrix:\n");

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

{

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

{

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

}

}

printf("A Matrix:\n");

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

{

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

{

printf("%d \t",a[i][j]);

}

printf("\n");

}

printf("e Scalar Product of a Matrix A: \n");

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

{

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

{

a[i][j] = k \* a[i][j];

}

}

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

{

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

{

printf("%d \t",a[i][j]);

}

printf("\n");

}

}

void main()

{

while("true")

{

int n,ch;

char ch2;

printf("\n");

printf("(1) Write a function to Create a Matrix of size m x n, and another function to Print a Matrix of size m x n.\n");

printf("(2) Create 2 matrices A and B of size (m x n). Find (A + B) and (A - B).\n");

printf("(3) Find Transpose of matrix A and of matrix B. Find (AT + BT) and (AT- BT).Check whether (AT + BT) = (A + B)Tand (AT- BT) = (A - B)T.\n");

printf("(4) Add matrix A and null matrix. Subtract null matrix from matrix A..\n");

printf("(5) Create a unit matrix of size (n x n) and a unit matrix of size (n x n).Multiply matrix A with a unit matrix.\n");

printf("(6) Create matrix A of size (m x n) and matrix B of size (n x p). Multiply matrix A andmatrix B to get matrix C of size (m x p).\n");

printf("(7) Create a symmetric matrix A. Find matrix AT. Check whether A = AT?.\n");

printf("(8) Evaluate Scalar Product of a Matrix A: For example, k A, where k is a constant (number).\n");

printf("(9) Take as input two matrices, A & B and print (A \* B) and (B \* A). First check which ones out of (A \* B) and (A \* B) are possible to compute.\n");

printf("\n");

printf("Enter your choices: ");

scanf("%d",&n);

if(n==1)

{

matrix\_display();

}

else if(n==2)

{

Matrixab();

}

else if(n==3)

{

matrixtras();

}

else if(n==4)

{

nullsubmatrix();

}

else if(n==5)

{

matrix5();

}

else if(n==6)

{

mulmatrix();

}

else if(n==7)

{

matrix7();

}

else if(n==8)

{

matrix8();

}

else if(n==9)

{

matrix9();

}

else

{

printf("Invalid..!\n");

}

printf("\nDo u want to continues..? (Press Y or y): ");

ch2=getch();

if(ch2=='y' || ch2=='Y')

{

continue;

}

else

{

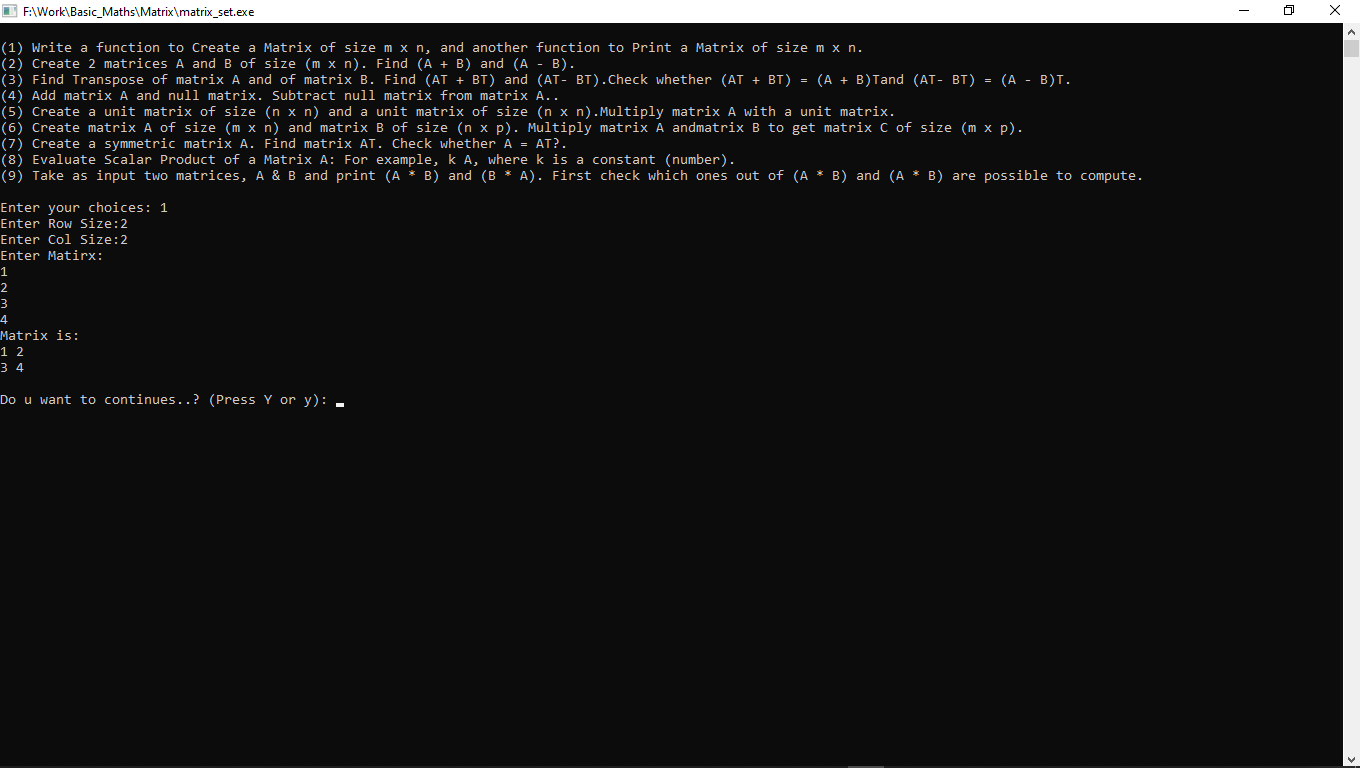
break;

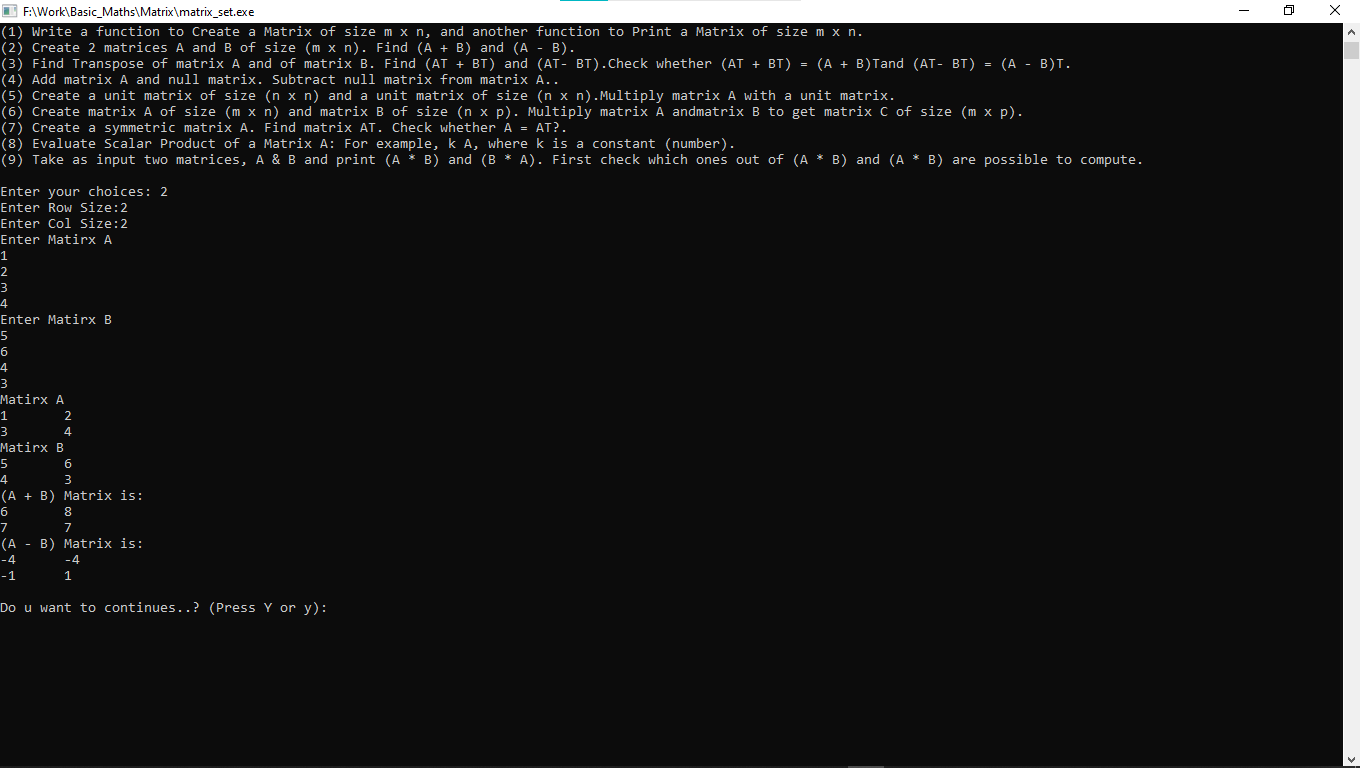
}

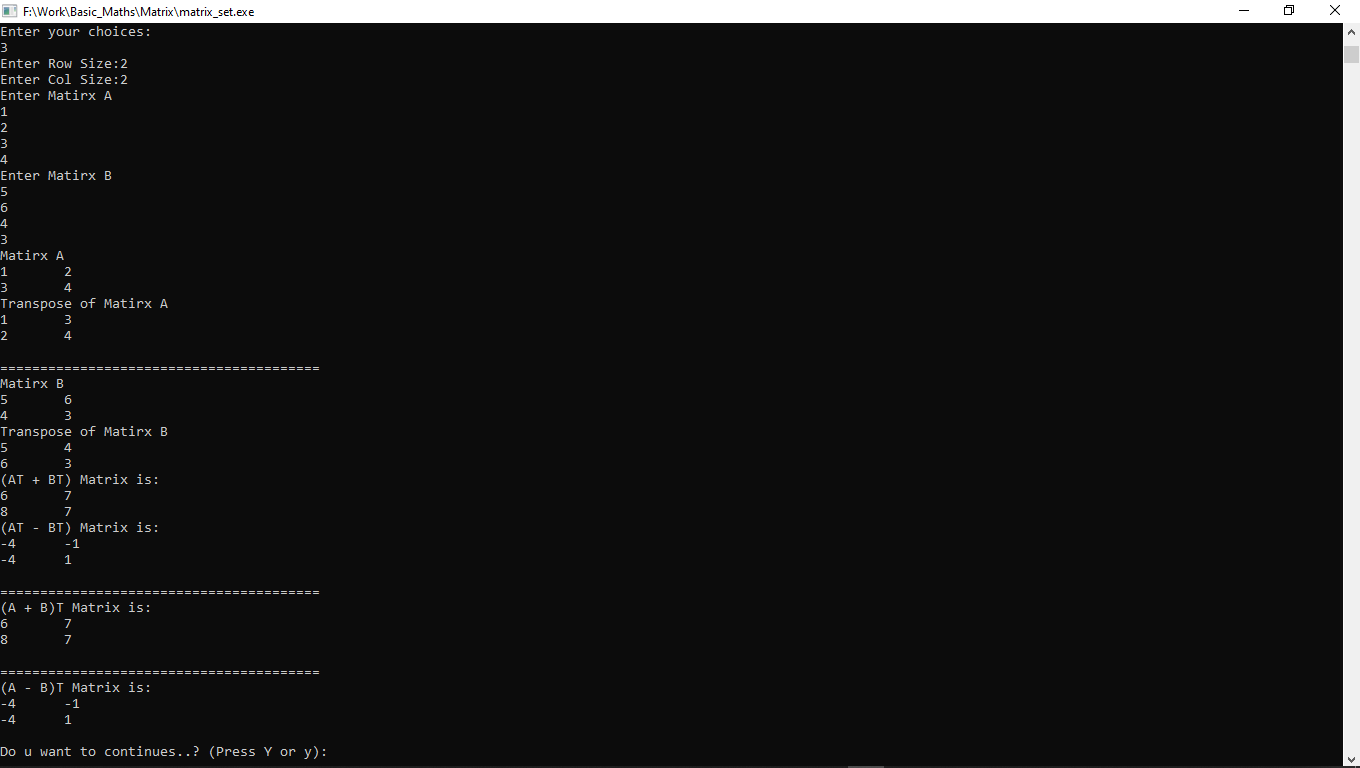
}

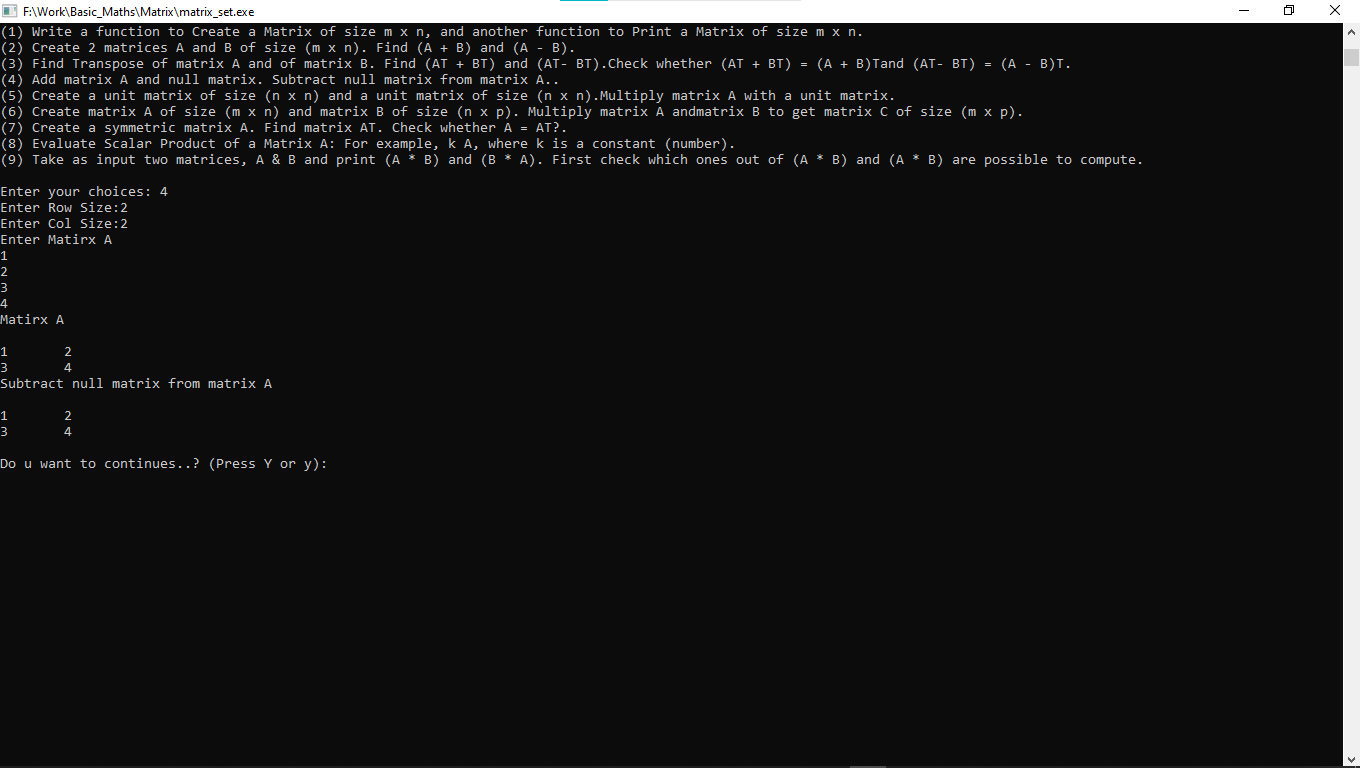
}

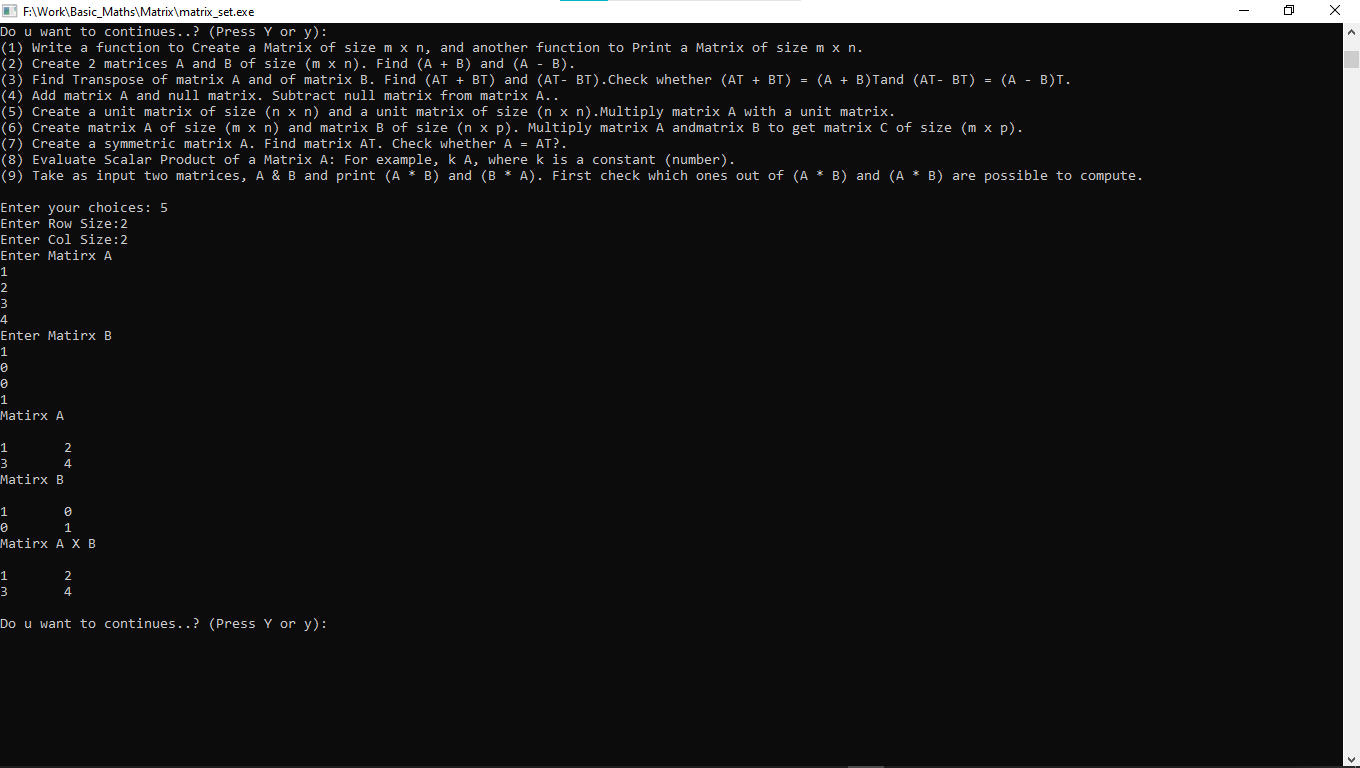
**OUTPUT:-**

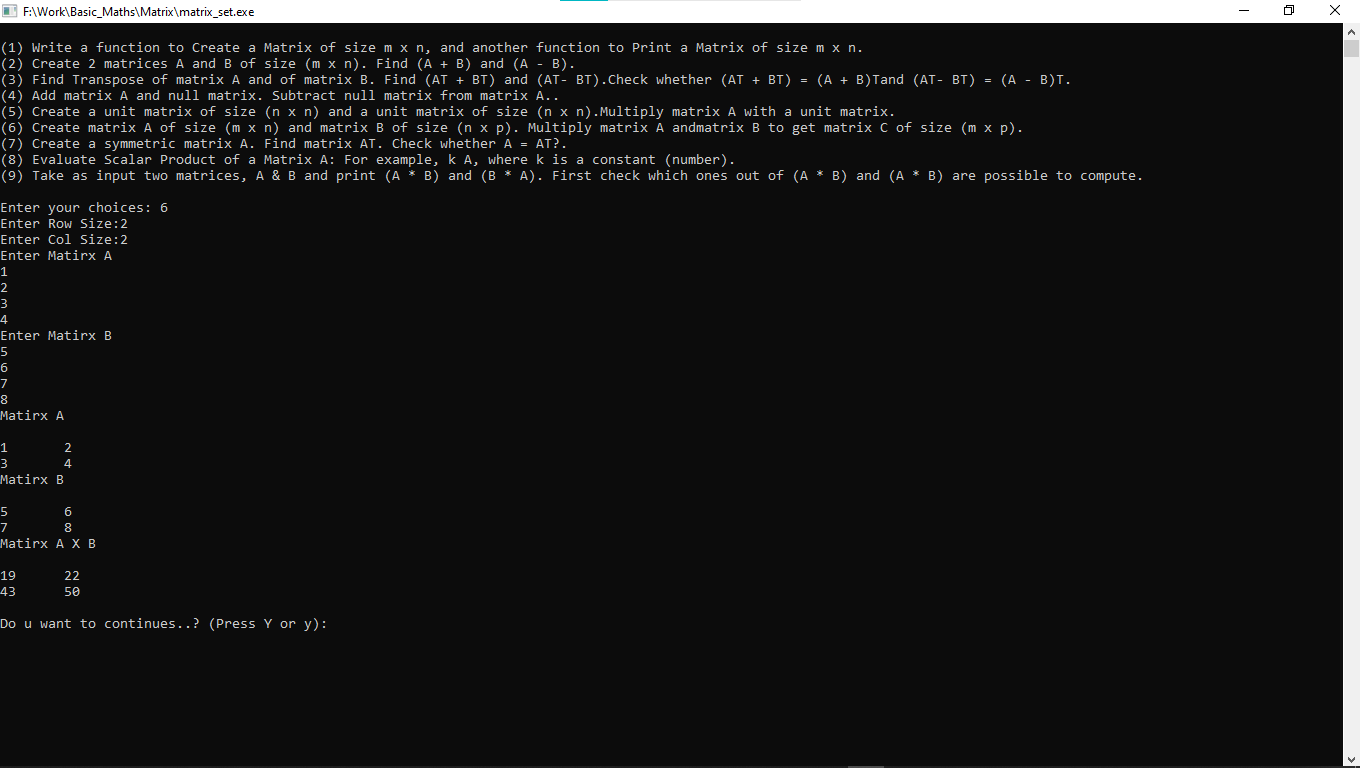


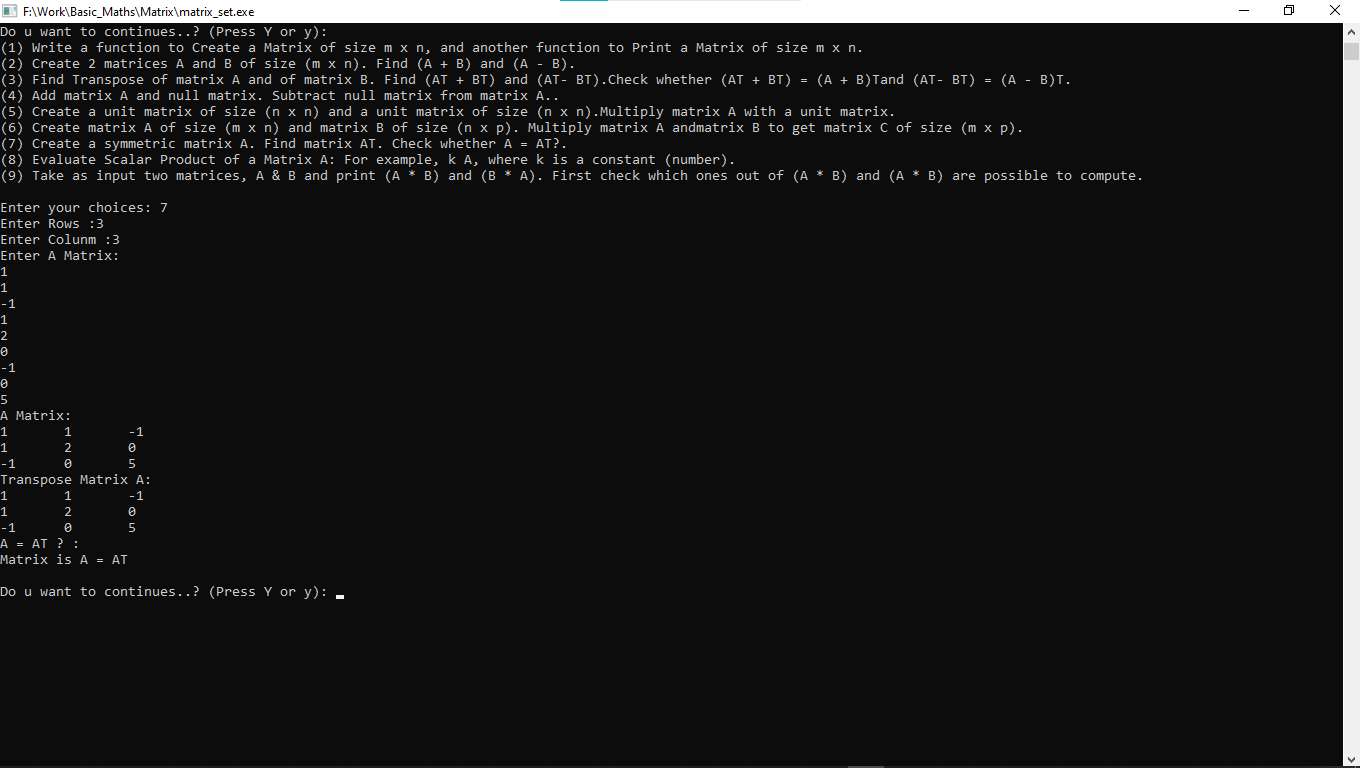


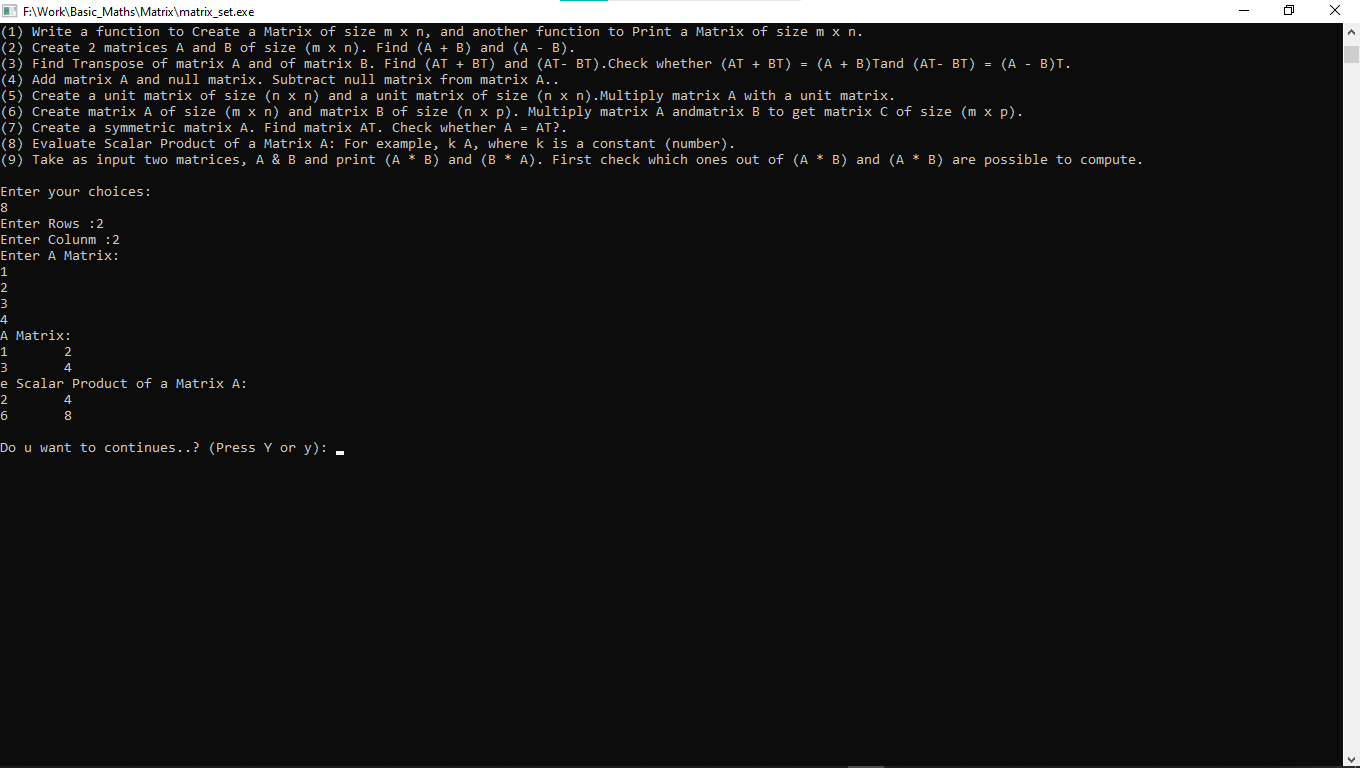


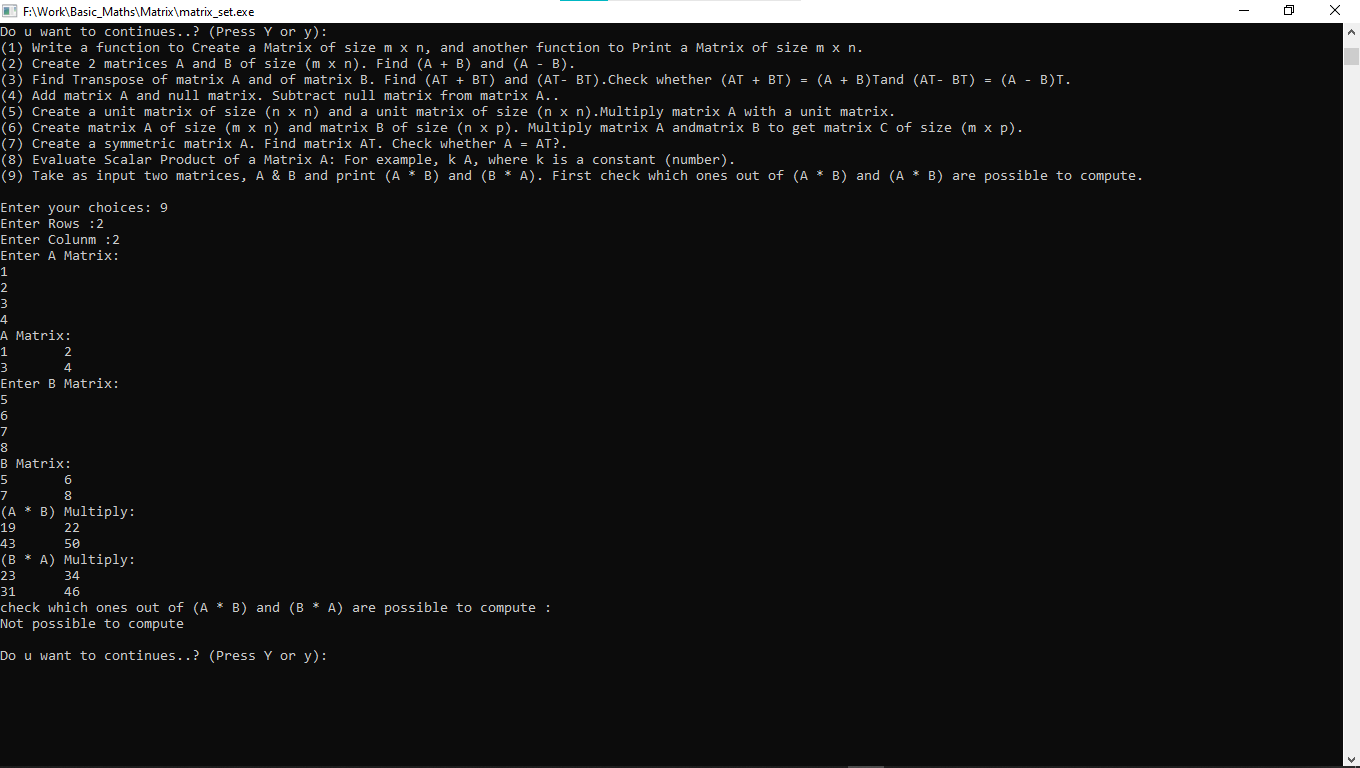












**Integers:**

**• Given a positive integer, find its divisors. Example: Divisors of 36 are 2, 3, 4, 6, 9, 12, 18.**

**• Given a positive integer n, represent n as product of its divisors. Example: 36 = 2 x 2 x 3 x 3.**

**• Given a positive integer, find whether it is a prime number or a composite number. Write an efficient algorithm.**

**• Given two positive integers, m and n, find whether they are relative prime numbers or not. • Given two positive integers, a and b, find Least Common Multiples (LCM) of a and b.**

**• Given two positive integers a and b, find GCD (Greatest Common Divisor) of a and b.**

#include<stdio.h>

#include<conio.h>

#include<math.h>

void Divisors()

{

int i,a;

printf("Enter A: ");

scanf("%d",&a);

for(i=2;i<a;i++)

{

if(a%i==0)

{

printf("\n%d",i);

}

}

}

void pdivisors()

{

int n,i,a=2;

printf("Enter Number: ");

scanf("%d",&n);

while (n%2 == 0)

{

printf("%d ",a);

n = n/a;

}

for ( i = 3; i <= sqrt(n); i = i+2)

{

while (n%i == 0)

{

printf("%d ", i);

n = n/i;

}

}

if (n > 2)

{

printf ("%d ", n);

}

}

void findprime()

{

int n,i;

printf("Enter N: ");

scanf("%d",&n);

for(i=2;i<n;i++)

{

if(n%i==0)

{

break;

}

}

if(i==n)

{

printf("prime number %d ",n);

}

else

{

printf("composite number %d",n);

}

}

int relativeprime()

{

int n1, n2, min, i, flag = 1;

printf("Enter 1 positive numbers\n");

scanf("%d", &n1);

printf("Enter 2 positive numbers\n");

scanf("%d",&n2);

min = n1 < n2 ? n1 : n2;

for(i = 2; i <= min; i++)

{

if( n1 % i == 0 && n2 % i == 0 )

{

flag = 0;

break;

}

}

if(flag)

{

printf("%d and %d are relative prime number\n", n1, n2);

}

else

{

printf("%d and %d are not relative prime number\n", n1, n2);

}

}

void lcm()

{

int i,a,b,gcd,lcm;

printf("Enter A: ");

scanf("%d",&a);

printf("Enter B: ");

scanf("%d",&b);

for(i=1;i<=a && i<=b;++i)

{

if(a%i==0 && b%i==0)

{

gcd=i;

}

}

lcm=(a\*b)/gcd;

printf("\nLCM of %d and %d is = %d",a,b,lcm);

}

void gcd()

{

int i,a,b,gcd;

printf("Enter A: ");

scanf("%d",&a);

printf("Enter B: ");

scanf("%d",&b);

for(i=1;i<=a && i<=b;++i)

{

if(a%i==0 && b%i==0)

{

gcd=i;

}

}

printf("GCD of %d and %d is = %d",a,b,gcd);

}

void main()

{

while("true")

{

int n,ch;

char ch2;

printf("\n");

printf("(1) Given a positive integer, find its divisors.\n");

printf("(2) Given a positive integer n, represent n as product of its divisors.\n");

printf("(3) Given a positive integer, find whether it is a prime number or a composite number.\n");

printf("(4) Given two positive integers, m and n, find whether they are relative prime numbers or not.\n");

printf("(5) Given two positive integers, a and b, find Least Common Multiples (LCM) of a and b.\n");

printf("(6) Given two positive integers a and b, find GCD (Greatest Common Divisor) of a and b.\n");

printf("\n");

printf("Enter your choices: ");

scanf("%d",&n);

if(n==1)

{

Divisors();

}

else if(n==2)

{

pdivisors();

}

else if(n==3)

{

findprime();

}

else if(n==4)

{

relativeprime();

}

else if(n==5)

{

lcm();

}

else if(n==6)

{

gcd();

}

else

{

printf("Invalid..!\n");

}

printf("\nDo u want to continues..? (Press Y or y): ");

ch2=getch();

if(ch2=='y' || ch2=='Y')

{

continue;

}

else

{

break;

}

}

}

**OUTPUT:-**

