***Assignment -1***

1. **WAP to implement Magic Square**.

#include<stdio.h>

int main ()

{

int n;

printf ("Enter order of magic square:");

scanf ("%d”, &n);

int i, j, x, y;

int arr[20][20] = {0};

int cnt=1;

i=x=0;

j=y=n/2;

while(cnt<=n\*n)

{

if (cnt%n==1&&cnt! =1)

{

i=x+1;

j=y;

}

if(i<0&&j<0)

{

i=x+1;

j=y;

}

if(i<0&&j>=0)

{

i=n-1;

}

if(i>=0&&j<0)

{

j=n-1;

}

arr[i][j] =cnt;

x=i;

y=j;

i--;

j--;

cnt++;

}

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

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

{

printf("%d”, arr[i][j]);

printf(" ");

}

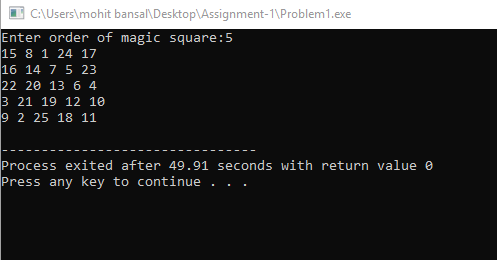
printf("\n");

}

return 0;

}

**OUTPUT :**



1. **WAP to implement Call by Value and Call by Reference mechanisms.**

#include<stdio.h>

void callbyvalue(int a,int b)

{

int c;

c=a;

a=b;

b=c;

}

void callbyreference(int \*c,int \*d)

{

int e;

e = \*c;

\*c = \*d;

\*d = e;

}

int main()

{

int a,b;

printf("Enter numbers a & b which are to be swapped");

scanf("%d%d",&a,&b);

printf("Before call by value:\n a=%d\tb=%d\n\n",a,b);

callbyvalue(a,b);

printf("After call by value:\n a=%d\tb=%d\n\n",a,b);

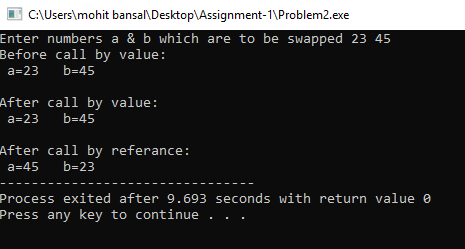
callbyreference(&a,&b);

printf("After call by referance:\n a=%d\tb=%d",a,b);

return 0;

}

**OUTPUT :**



1. **WAP to implement three different ways to swap two variables without using a third variable.**

**#**include<stdio.h>

void add(int a,int b)

{

a-=-(b-=a+=b);

printf("a=%d\tb=%d\n\n",a,-b);

}

void mul(int a,int b)

{

a=a\*b;

b=a/b;

a=a/b;

printf("a=%d\tb=%d\n\n",a,b);

}

void xor(int a,int b)

{

a^=b^=a^=b;

printf("a=%d\tb=%d\n\n",a,b);

}

int main()

{

int a,b;

printf("Enter two numbers a & b which are to be swapped:");

scanf("%d%d",&a,&b);

printf("Using addition:\n");

add(a,b);

printf("Using multipication:\n");

mul(a,b);

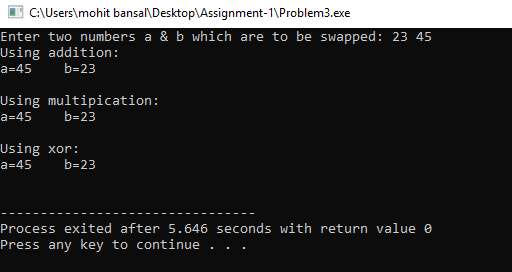
printf("Using xor:\n");

xor(a,b);

return 0;

}

**OUTPUT :**



1. **WAP to implement the following programs using recursion.**

**a. Factorial**

#include<stdio.h>

int fac(int n)

{

if(n==1)

return 1;

return n\*fac(n-1);

}

int main()

{

int n;

printf("Enter the number for which factorial is to be calculated:\n");

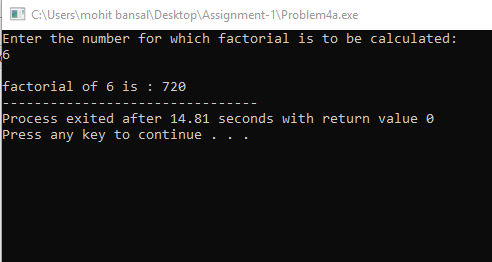
scanf("%d",&n);

printf("\nfactorial of %d is : %d",n,fac(n));

return 0;

}

**OUTPUT :**



**b. Fibonacci Series**

#include<stdio.h>

int arr[10000];

int fib(int n)

{

if(n==0)

return 0;

if(n==1)

return 1;

if(arr[n]!=0)

return arr[n];

arr[n]=fib(n-1)+fib(n-2);

return arr[n];

}

int main()

{

int n;

printf("Enter number of element of series you want to display\n");

scanf("%d",&n);

if(n==1||n==2)

printf("1");

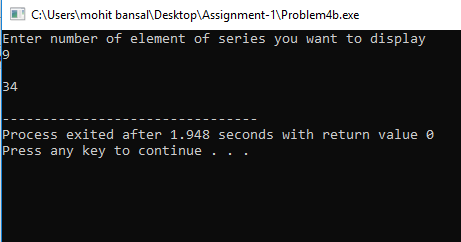
else

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

return 0;

}

**OUTPUT :**



**c. Greatest Common Divisor**

#include<stdio.h>

int gcd(int a,int b)

{

if(b==0)

return a;

return gcd(b,a%b);

}

int main()

{

int a,b;

printf("Enter two numbers:\n");

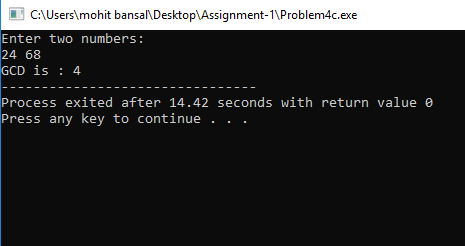
scanf("%d%d",&a,&b);

printf("GCD is : %d",gcd(a,b));

return 0;

}

**OUTPUT :**



**d. Linear Search**

#include<stdio.h>

int arr[10000]={0};

int n,x;

int ls(int i)

{

if(i>=n)

return -1;

else if(arr[i]==x)

return i;

else

return ls(i+1);

}

int main()

{

printf("Enter size of array:");

scanf("%d",&n);

int i;

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

{

printf("Enter %dth element of array:",i);

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

}

printf("enter element to be searched:");

scanf("%d",&x);

int res = ls(0);

if(res==-1)

printf("not found\n");

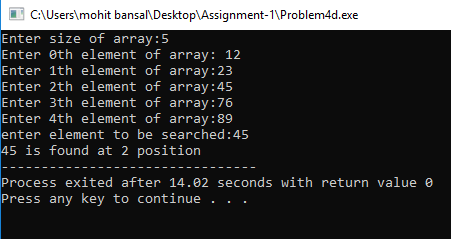
else

printf("%d is found at %d position",x,res);

return 0;

}

**OUTPUT :**



**e. Binary Search**

#include<stdio.h>

int arr[10000];

int index[100000];

int n,x;

void sort()

{

int i,j,temp;

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

{

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

{

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

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}}}

int bs(int low,int high)

{

if(low==high)

{

if(arr[low]==x)

return low;

else

return -1;

}

int mid;

mid = (low+high)/2;

if(arr[mid]>x)

return bs(low,mid-1);

else if(arr[mid]<x)

return bs(mid+1,high);

else

return mid;

}

int main()

{

printf("Enter size of array:");

scanf("%d",&n);

int i;

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

{

printf("enter %dth element of array: ",i);

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

}

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

index[arr[i]] = i;

sort();

printf("Enter element to be searched:");

scanf("%d",&x);

int res = bs(0,n-1);

if(res==-1)

printf("not found");

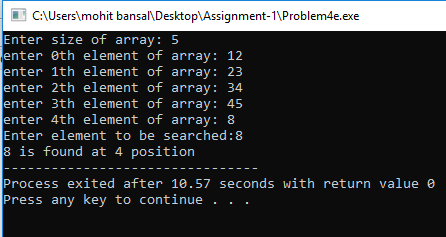
else

printf("%d is found at %d position",x,index[x]);

return 0;

}

**OUTPUT :**



**f. Tower of Hanoi**

#include<stdio.h>

void toh(int n,char src,char aux,char dest)

{

if(n==0)

return ;

toh(n-1,src,dest,aux);

printf("%c -> %c\n",src,dest);

toh(n-1,aux,src,dest);

}

int main()

{

int n;

printf("Enter number of disks:");

scanf("%d",&n);

toh(n,'S','A','D');

return 0;

}

**OUTPUT :**

