# 1.BINARY SEARCH

#include<stdio.h>

int main()

{

int i,n,a[100],flag=0,key,pos,f,l,mid;

printf("enter n & key:");

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

printf("enter the array elements");

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

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

f=0;

l=n-1;

while (f<=l)

{

mid=(f+l)/2;

if (a[mid]==key)

{

flag=1;

break;

}

else if(a[mid]<key)

f=mid+1;

else if(a[mid]>key)

l=mid-1;

}

if(flag==1)

printf("the key element %d found in %d position",key,pos);

else

printf("not found");

}

OUTPUT:

Enter number of elementsn4

Enter 4 integersn5

8

6

2

Enter value to findn7

Not found! 7 isn't present in the list.n

--------------------------------

Process exited after 15.31 seconds with return value 0

Press any key to continue . . .

## 2.LINEAR SEARCH

#include <stdio.h>

int main()

{

int array[100], search, c, n;

printf("Enter number of elements in array\n");

scanf("%d", &n);

printf("Enter %d integer(s)\n", n);

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

scanf("%d", &array[c]);

printf("Enter a number to search\n");

scanf("%d", &search);

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

{

if (array[c] == search)

{

printf("%d is present at location %d.\n", search, c+1);

break;

}

}

if (c == n)

printf("%d isn't present in the array.\n", search);

return 0;

}

OUTPUT:

Enter number of elements in array

4

Enter 4 integer(s)

2

4

8

6

Enter a number to search

4

4 is present at location 2.

--------------------------------

Process exited after 16.21 seconds with return value 0

Press any key to continue . . .

### 3.MATRIX MULTIPLICATION

#include<stdio.h>

int main()

{

int i,j,r,c,a[10][10],b[10][10],mul[10][10],k;

printf("enter the no of rows=");

scanf("%d",&r);

printf("enter the no of columns=");

scanf("%d",&c);

printf("enter the first matrix elements=");

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

{

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

{

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

}

}

printf("enter the second matrix elements=");

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

{

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

{

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

}

}

printf("multiply of matrix=\n");

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

{

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

{

mul[i][j]=0;

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

{

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

}

}

}

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

{

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

{

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

}

printf("\n");

}

return 0;

}

OUTPUT:

enter the no of rows=3

enter the no of columns=3

enter the first matrix elements=

1 2 3

3 4 5

7 6 8

enter the second matrix elements=

2 3 4

4 5 6

9 8 7

multiply of matrix=

37 37 37

67 69 71

110 115 120

--------------------------------

Process exited after 53.47 seconds with return value 0

Press any key to continue . . .

### 4.FACTORIAL USING RECURSION

#include<stdio.h>

long factorial(int n)

{

if (n == 0)

return 1;

else

return(n \* factorial(n-1));

}

int main()

{

int number;

long fact;

printf("Enter a number: ");

scanf("%d", &number);

fact = factorial(number);

printf("Factorial of %d is %ld\n", number, fact);

return 0;

}

OUTPUT:

Enter a number: 4

Factorial of 4 is 24

--------------------------------

Process exited after 13.6 seconds with return value 0

Press any key to continue . . .

### 5.FACTORIAL WITHOUT RECURSION

#include<stdio.h>

int main()

{

int i,fact=1,number;

printf("Enter a number: ");

scanf("%d",&number);

for(i=1;i<=number;i++){

fact=fact\*i;

}

printf("Factorial of %d is: %d",number,fact);

return 0;

}

OUTPUT:

Enter a number: 3

Factorial of 3 is: 6

--------------------------------

Process exited after 2.839 seconds with return value 0

Press any key to continue . . .

6.FIBONACCI USING RECURSION

#include<stdio.h>

int Fibonacci(int);

int main()

{

int n, i = 0, c;

scanf("%d",&n);

printf("Fibonacci series\n");

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

{

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

i++;

}

return 0;

}

int Fibonacci(int n)

{

if ( n == 0 )

return 0;

else if ( n == 1 )

return 1;

else

return ( Fibonacci(n-1) + Fibonacci(n-2) );

}

OUTPUT:

5

Fibonacci series

0

1

1

2

3

--------------------------------

Process exited after 3.235 seconds with return value 0

Press any key to continue . . .

### 7.FIBONACCI WITHOUT RECURSION

#include<stdio.h>

int main()

{

int n1=0,n2=1,n3,i,number;

printf("Enter the number of elements:");

scanf("%d",&number);

printf("\n%d %d",n1,n2);

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

{

n3=n1+n2;

printf(" %d",n3);

n1=n2;

n2=n3;

}

return 0;

}

OUTPUT:

Enter the number of elements:5

0 1 1 2 3

--------------------------------

Process exited after 2.946 seconds with return value 0

Press any key to continue . . .

### 8.ARRAY INSERT,DELETE AND DISPLAY

#include<stdio.h>

#include<stdlib.h>

#define size 100

int a[size],n=6;

void create()

{

int i;

printf("enter the size");

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

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

}

void insert()

{

int i,p,e;

printf("enter the position & element");

scanf("%d%d",&p,&e);

for(i=n;i>=p;i--)

a[i]=a[i-1];

a[p]=e;

}

void del()

{

int i,p;

printf("enter the position");

scanf("%d",&p);

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

a[i]=a[i+1];

}

void display()

{

int i,ch;

do{

printf("\nMenu\n1.After Create\n2.After Insertion\n3.After Deletion");

printf("Enter your choice");

scanf("%d",&ch);

switch(ch)

{

case 1:

printf("the elements are\n");

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

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

break;

case 2:

printf("the elements are\n");

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

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

break;

case 3:

printf("the elements are\n");

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

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

break;

default: printf("Enter 1,2,3");

}

}while(ch>0 && ch<=3);

}

int main()

{

int choice;

do

{

printf("main menu\n");

printf("\n1.create\n2.insert\n3.delete\n4.display\n5.exit");

printf("enter your choice");

scanf("%d",&choice);

switch (choice)

{

case 1:

create();

break;

case 2:

insert();

break;

case 3:

del();

break;

case 4:

display();

break;

case 5:

exit(0);

default:

printf("enter numbers from 1 to 5");

}

}while (choice>0 && choice<=5);

return 0;

}

9.ODD OR EVEN IN A ARRAY

#include <stdio.h>

void main()

{

int i, n;

printf("Enter number of elements in the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter %d elements in the array: ",n);

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

{

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

}

printf("Even numbers in the array are: ");

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

{

if(arr[i]&1==1);

else

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

}

printf("\nOdd numbers in the array are: ");

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

{

if(arr[i]&1==1)

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

}

}