**Program-1:**

#include<stdio.h>

#include<stdlib.h>

void insert(int n,int arr[],int index,int num){

int i;

for(i=n-1;i>=(index-1);i--){

arr[i+1]=arr[i];

}

arr[index-1]=num;

n++;

printf("Array after Insertion:\n");

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

printf("Array[%d]: %d\n",i,arr[i]);

}

}

void del(int n,int arr[],int index){

int i;

for(i=index-1;i<=n;i++){

arr[i]=arr[i+1];

}

printf("Array after Deletion:\n");

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

printf("Array[%d]: %d\n",i,arr[i]);

}

}

int main(){

int n,i,ch,index,num;

printf("Enter No. of Terms: ");

scanf("%d",&n);

int arr[n];

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

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

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

}

while(1){

printf("-----------------\n");

printf("1. Insertion\n2. Deletion\n3. Exit\nEnter your Choice: ");

scanf("%d",&ch);

switch(ch){

case 1:printf("Enter Index: ");

scanf("%d",&index);

printf("Enter Number: ");

scanf("%d",&num);

insert(n,arr,index,num);

break;

case 2:printf("Enter Index: ");

scanf("%d",&index);

del(n,arr,index);

break;

case 3:printf("EXITING...");

exit(0);

default:printf("Invalid Input!");

break;

}

}

return 0;

}

**Output:**

Enter No. of Terms: 5

Array[0]: 10

Array[1]: 20

Array[2]: 30

Array[3]: 40

Array[4]: 50

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

1. Insertion

2. Deletion

3. Exit

Enter your Choice: 1

Enter Index: 6

Enter Number: 60

Array after Insertion:

Array[0]: 10

Array[1]: 20

Array[2]: 30

Array[3]: 40

Array[4]: 50

Array[5]: 60

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

1. Insertion

2. Deletion

3. Exit

Enter your Choice: 2

Enter Index: 3

Array after Deletion:

Array[0]: 10

Array[1]: 20

Array[2]: 40

Array[3]: 50

Array[4]: 60

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

1. Insertion

2. Deletion

3. Exit

Enter your Choice: 3

EXITING...

**Program-2:**

#include <stdio.h>

#include <stdlib.h>

#define SIZE 4

int top = -1, inp\_array[SIZE];

void push();

void pop();

void show();

int main()

{

int choice;

while (1)

{

printf("\nPerform operations on the stack:");

printf("\n1.Push the element\n2.Pop the element\n3.Show\n4.End");

printf("\n\nEnter the choice: ");

scanf("%d", &choice);

switch (choice)

{

case 1:

push();

break;

case 2:

pop();

break;

case 3:

show();

break;

case 4:

exit(0);

default:

printf("\nInvalid choice!!");

}

}

}

void push()

{

int x;

if (top == SIZE - 1)

{

printf("\nOverflow!!");

}

else

{

printf("\nEnter the element to be added onto the stack: ");

scanf("%d", &x);

top = top + 1;

inp\_array[top] = x;

}

}

void pop()

{

if (top<0)

{

printf("\nUnderflow!!");

}

else

{

int val=inp\_array[top];

printf("\nPopped element: %d", val);

top = top - 1;

}

}

void show()

{

if (top<0)

{

printf("\nUnderflow!!");

}

else

{

printf("\nElements present in the stack: \n");

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

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

}

}

**Output:**

Perform operations on the stack:

1.Push the element

2.Pop the element

3.Show

4.End

Enter the choice: 1

Enter the element to be added onto the stack: 100

Perform operations on the stack:

1.Push the element

2.Pop the element

3.Show

4.End

Enter the choice: 3

Elements present in the stack:

100

Perform operations on the stack:

1.Push the element

2.Pop the element

3.Show

4.End

Enter the choice: 2

Popped element: 100

Perform operations on the stack:

1.Push the element

2.Pop the element

3.Show

4.End

Enter the choice: 4

**Program-2:**

#include<stdio.h>

#include<stdlib.h>

void tower(int n,char beg,char aux,char end){

if(n==1){

printf("Move Disc 1 from Tower %c to Tower %c\n",beg,end);

return;

}

tower(n-1,beg,end,aux);

printf("Move Disc %d from Tower %c to Tower %c\n",n,beg,end);

tower(n-1,aux,beg,end);

}

int fact(int n){

if(n==0)

return 1;

else

return n \* fact(n-1);

}

int fib(int n){

if(n==0 || n==1)

return n;

else

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

}

int gcd(int x, int y){

if(y==0)

return x;

else if(x==0)

return y;

else{

if(x>y){

int temp=x;

x=y;

y=temp;

}

return gcd(x,y%x);

}

}

int main(){

int ch,n,x,y,i;

while(1){

printf("----------------\n");

printf("1. TOWER OF HANOI\n2. FACTORIAL\n3. FIBONACCI\n4. GCD\n5. EXIT\n");

printf("----------------\n");

printf("Enter your Choice: ");

scanf("%d",&ch);

switch(ch){

case 1:printf("Enter No. of Discs: ");

scanf("%d",&n);

tower(n,'A','B','C');

break;

case 2:printf("Enter the Number: ");

scanf("%d",&n);

printf("The Factorial is: %d\n",fact(n));

break;

case 3:printf("Enter the Number of Terms: ");

scanf("%d",&n);

printf("The Fibonacci Series is: ");

if(n==0)

printf("N.A");

else{

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

printf("%d ",fib(i));

}

printf("\n");

break;

case 4:printf("Enter 1st No.: ");

scanf("%d",&x);

printf("Enter 2nd No.: ");

scanf("%d",&y);

printf("The GCD of two no.s is: %d\n",gcd(x,y));

break;

case 5:printf("EXITING...");

exit(0);

default:printf("Invalid Choice!\n");

break;

}

}

return 0;

}

**Output:**

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

1. TOWER OF HANOI

2. FACTORIAL

3. FIBONACCI

4. GCD

5. EXIT

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

Enter your Choice: 1

Enter No. of Discs: 3

Move Disc 1 from Tower A to Tower C

Move Disc 2 from Tower A to Tower B

Move Disc 1 from Tower C to Tower B

Move Disc 3 from Tower A to Tower C

Move Disc 1 from Tower B to Tower A

Move Disc 2 from Tower B to Tower C

Move Disc 1 from Tower A to Tower C

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

1. TOWER OF HANOI

2. FACTORIAL

3. FIBONACCI

4. GCD

5. EXIT

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

Enter your Choice: 2

Enter the Number: 5

The Factorial is: 120

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

1. TOWER OF HANOI

2. FACTORIAL

3. FIBONACCI

4. GCD

5. EXIT

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

Enter your Choice: 3

Enter the Number of Terms: 5

The Fibonacci Series is: 0 1 1 2 3

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

1. TOWER OF HANOI

2. FACTORIAL

3. FIBONACCI

4. GCD

5. EXIT

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

Enter your Choice: 4

Enter 1st No.: 8

Enter 2nd No.: 4

The GCD of two no.s is: 4

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

1. TOWER OF HANOI

2. FACTORIAL

3. FIBONACCI

4. GCD

5. EXIT

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

Enter your Choice: 5

EXITING...

**Program-4:**

#include<stdio.h>

#include<stdlib.h>

# define SIZE 4

void enqueue();

void dequeue();

void show();

int inp\_arr[SIZE];

int Rear = - 1;

int Front = - 1;

main()

{

int ch;

while (1)

{

printf("1.Enqueue Operation\n");

printf("2.Dequeue Operation\n");

printf("3.Display the Queue\n");

printf("4.Exit\n");

printf("Enter your choice of operations : ");

scanf("%d", &ch);

switch (ch)

{

case 1:

enqueue();

break;

case 2:

dequeue();

break;

case 3:

show();

break;

case 4:

exit(0);

default:

printf("Incorrect choice \n");

}

}

}

void enqueue()

{

int insert\_item;

if (Rear == SIZE - 1)

printf("Overflow \n");

else

{

if (Front == - 1)

Front = 0;

printf("Element to be inserted in the Queue: ");

scanf("%d", &insert\_item);

Rear = Rear + 1;

inp\_arr[Rear] = insert\_item;

}

}

void dequeue()

{

if (Front == - 1 || Front > Rear)

{

printf("Underflow \n");

return ;

}

else

{

printf("Element deleted from the Queue: %d\n", inp\_arr[Front]);

Front = Front + 1;

}

}

void show(){

int i;

if (Front == - 1)

printf("Empty Queue \n");

else

{

printf("Queue: \n");

for(i = Front; i <= Rear; i++)

printf("%d ", inp\_arr[i]);

printf("\n");

}

}

**Output:**

1.Enqueue Operation

2.Dequeue Operation

3.Display the Queue

4.Exit

Enter your choice of operations : 1

Element to be inserted in the Queue: 100

1.Enqueue Operation

2.Dequeue Operation

3.Display the Queue

4.Exit

Enter your choice of operations : 3

Queue:

100

1.Enqueue Operation

2.Dequeue Operation

3.Display the Queue

4.Exit

Enter your choice of operations : 2

Element deleted from the Queue: 100

1.Enqueue Operation

2.Dequeue Operation

3.Display the Queue

4.Exit

Enter your choice of operations : 4