

Program - 1

Ques) Write a program to implement Insertion & Deletion in an array.

```
#include <stdio.h>
int main()
{
    int arr[10] = {1, 5, 78, 56, 12, 44};
    int K, n, j, i, N;
    // n = total no. of elements in array
    // k = index no.
```

Traversing

```
n = 6;
j = n;
```

```
printf("array before insertion:\n");
for (i = 0; i < n; i++)
{
    printf("%d", arr[i]);
```

// Insertion

```
printf("Enter the index for insertion = ");
scanf("%d", &k);
```

```
while (j >= k)
{
```

```
    arr[j + 1] = arr[j];
```

```
    j = j - 1;
```

```
}
```

```
printf("enter the no. to be inserted = ");
scanf("%d", &arr[k]);
```

```
n = n + 1;
```

```
Printf ("array after insertion:\n");
for (i = 0; i < n; i++)
    printf ("%d", arr[i]);
```

// Deletion

```
Printf ("Enter the index to be deleted :");
```

```
Scanf ("%d", &k);
```

```
n = arr[k];
```

```
j = k;
```

```
while (j < n) {
```

```
arr[j] = arr[j+1];
```

```
j = j+1;
```

```
n = n-1;
```

```
Printf ("array after deletion:\n");
```

```
for (i = 0; i < n; i++)
    printf ("%d", arr[i]);
```

```
}
```

```
} return 0;
```

Out Put

array before Insertion:

1 5 78 56 12 44

Enter the index for insertion = 4

Enter the no. to be inserted = 33

array after insertion:

1 5 78 56 33 12 44

Enter the index to be deleted = 1

array after deletion:

78 56 33 12 44

BB

Program - 2

WAP to perform push and pop operations in stack

```
#include <StdIo.h>
#include <StdLib.h>
#define Max 10
int Stack[Max];
int top = -1;
void Push (int value) {
    if (top == Max-1) {
        printf ("Stack Overflow");
        return;
    }
    Stack [++top] = value;
}
int Pop () {
    if (top == -1) {
        printf ("Error Stack Underflow");
        return -1;
    }
    return Stack [top--];
}
int main () {
    int choice, value;
    while () {
        printf ("1) Push\n2) Pop\n3) Exit\n");
        printf ("Enter your choice");
        scanf ("%d", &choice);
        switch (choice) {
```

Case I :

```
    printf ("Enter value to be Push");
    scanf ("%d", &value);
    Push (value);
    break;
```

Case II:

```
    value = pop();
    if (value != -1) {
        cout << "Popped value: " << value;
    }
    break;
```

Case III:

```
    exit(0);
default:
    cout << "Invalid choice(" << i << ")";
}
return 0;
```

Out Put:

- 1) PUSH
- 2) POP
- 3) EXIT

Enter your choice 1

Enter value to be pushed 15

- 1) PUSH
- 2) POP
- 3) EXIT

Enter your choice 1

Enter value to be pushed 45

- 1) PUSH
- 2) POP
- 3) EXIT

Enter your choice 1

Enter value to be pushed 25

80

Program - 3

WAP to implement Insertion and deletion in a queue.

```
#include <Stdio.h>
#define Size 100
Void insert();
Void delete();
Void Show();
Int arr [Size];
Int Rear = -1;
Int Front = -1;
Main()
{
    Int ch;
    While(1)
    {
        printf("1) Insert Operation\n");
        printf("2) Delete Operation\n");
        printf("3) Display Queue\n");
        printf("4) Exit\n");
        printf("Enter your choice of operation:");
        Scanf("%d", &ch);
        Switch(ch)
        {
            Case I:
                insert();
                break;
            Case II:
                delete();
                break;
            Case III:
                Show();
                break;
            Case IV:
                exit(0);
            Default:
        }
    }
}
```

```

printf (" In CreateCircularQ()");

}

void Insert()
{
    int Insert_item;
    printf (" Rear = Size - 1 ");
    printf (" " "Overflow" );
    else
    {
        if (front == -1)
            front = 0;
        printf (" Element to be inserted in Queue \n" );
        Scanf ("%d", &Insert_item);
        Rear = Rear + 1;
        arr[Rear] = Insert_item;
    }
}

void delete()
{
    if (front == -1 || front > Rear)
    {
        printf (" Underflow \n" );
        return;
    }
    else
    {
        printf (" Element deleted from queue \n" );
        front = front + 1;
    }
}

void Show()
{
    if (front == -1)
        printf (" Empty Queue \n" );
    else
    {

```

```
printf("Queue: (%d);\n", q.front);
for (int i = front; i <= rear; i++)
    printf("%d ", arr[i]);
printf("\n");}
```

Output:

- (i) Insert
- (ii) Deletion
- (iii) Display
- (iv) Exit

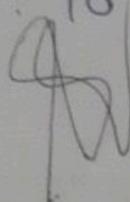
Enter your choice of operations: 1

Element to be inserted in Queue: 10

- (i) Insert
- (ii) Deletion
- (iii) Display
- (iv) Exit

Enter your choice of operations: 3

Element to be inserted in Queue: 10 20



- (i) Insert
- (ii) Deletion
- (iii) Display
- (iv) Exit

Enter choice of operations: 2

Element to be deleted from Queue: 10