Name: Atharva Salitri **Div: CSAI-B** **Batch: 2** **Roll No.: 37**

Queue implementation using Array

```
#include <stdio.h>
#include<stdlib.h>
#define MAX 50
int queue[MAX];
int front = -1;
int rear = -1;
void enqueue(){
   int x;
   if(rear == MAX -1 )
      printf("Overflow\n");
   else{
      if(front == -1)
         front = 0;
   }
   printf("Insert the element in queue: ");
   scanf("%d", &x);
   rear = rear + 1;
   queue[rear] = x;
}
void dequeue (){
   if (front == -1 || front > rear)
      printf("Underflow\n");
   else{
      printf("Element deleted is: %d\n",
queue[front]);
      front ++;
   }
}
void display(){
   if(front == -1)
      printf("Empty\n");
   else{
      printf("Elements: ");
      for(int i = front; i<=rear; i++)
         printf("%d ", queue[i]);
      printf("\n");
   }
}
```

```
int main() {
   int choice;
   while (1) {
      printf("\n1. Insert element to queue\n");
      printf("2. Delete element from queue\n");
      printf("3. Display all elements of queue\n");
      printf("4. Quit\n");
      printf("Enter your choice: ");
      scanf("%d", &choice);

      switch (choice) {
         case 1:
            enqueue();
            break;
         case 2:
            dequeue();
            break;
         case 3:
            display();
            break;
         case 4:
            exit(0);
         default:
            printf("Wrong choice\n");
      }
   }
   return 0;
}
```

```
1. Insert element to queue
2. Delete element from queue
3. Display all elements of queue
4. Quit
Enter your choice: 1
Insert the element in queue: 10

1. Insert element to queue
2. Delete element from queue
3. Display all elements of queue
4. Quit
Enter your choice: 1
Insert the element in queue: 20

1. Insert element to queue
2. Delete element from queue
3. Display all elements of queue
4. Quit
Enter your choice: 2
Element deleted is: 10
```

Name: Atharva Salitri **Div: CSAI-B** **Batch: 2** **Roll No.: 37**

Queue Implementation using Array

```
#include <stdio.h>
#include<stdlib.h>
typedef struct node{
   int data;
   struct node* next;
} Node;
Node* front = NULL;
Node* rear = NULL;
void enqueue(int x){
   Node* temp = (Node*)malloc(sizeof(Node));
   temp->data = x;
   temp->next = NULL;
   if(rear == NULL)
      front = rear = temp;
   else{
      rear->next = temp;
      rear=temp;
   }
}
void dequeue(){
   if(front == NULL)
      printf("Queue Empty\n");
   else{
      Node* temp = front;
      printf("Deleted Element is: %d\n", front->data);
      front = front -> next;
      if(front == NULL){
         rear == NULL;
      }
      free(temp);
   }
}
void display(){
   if(front == NULL)
      printf("Queue Empty");
   else{
      printf("Elements: ");
      Node* temp = front;
      while(temp!=NULL){
         printf("%d ", temp->data);
         temp = temp->next;
      }
      printf("\n"); } }
int main() {
   int choice, x;
   while (1) {
      printf("\n1. Insert element to queue\n");
      printf("2. Delete element from queue\n");
      printf("3. Display all elements of queue\n");
      printf("4. Quit\n");
      printf("Enter your choice: ");
      scanf("%d", &choice);
      switch (choice) {
         case 1:
            printf("Insert the element in queue: ");
            scanf("%d", &x);
            enqueue(x);  break;
         case 2:
            dequeue();   break;
         case 3:
            display();   break;
         case 4:
            exit(0);
         default:
            printf("Wrong choice\n");
      } }
   return 0;
}
```

```
1. Insert element to queue
2. Delete element from queue
3. Display all elements of queue
4. Quit
Enter your choice: 1
Insert the element in queue: 15

1. Insert element to queue
2. Delete element from queue
3. Display all elements of queue
4. Quit
Enter your choice: 1
Insert the element in queue: 30

1. Insert element to queue
2. Delete element from queue
3. Display all elements of queue
4. Quit
Enter your choice: 2
Deleted Element is: 15
```