PROBLEM: 1

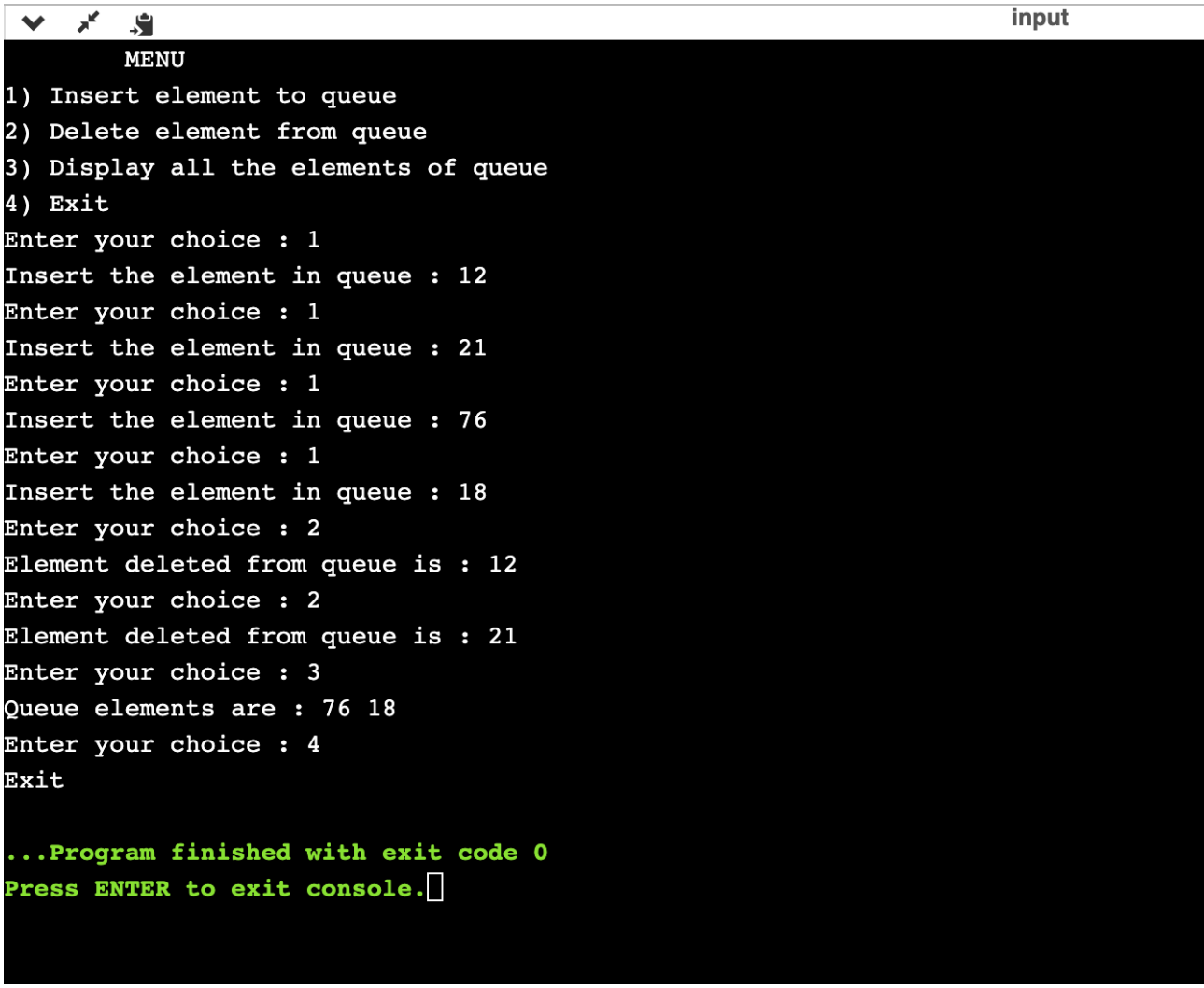
AIM: To write a program to perform insertion and deletion in simple queue.

Source code:

main.c

```c
#include<stdio.h>
int queue[100], n = 100, front = - 1, rear = - 1;
void Insert() {
   int val;
   if (rear == n - 1)
   printf("Queue Overflow");
   else {
      if (front == - 1)
      front = 0;
      printf("Insert the element in queue : ");
      scanf("%d",&val);
      rear++;
      queue[rear] = val;
   }
}
void Delete() {
   if (front == - 1 || front > rear) {
      printf("Queue Underflow ");
      return ;
   } else {
      printf("Element deleted from queue is : %d \n", queue[front]);
      front++;;
   }
}
void Display() {
   if (front == - 1)
   printf("Queue is empty");
   else {
      printf("Queue elements are : ");
      for (int i = front; i <= rear; i++)
      printf("%d ",queue[i]);
      printf("\n");
   }
}
int main() {
   int ch;
   printf("\tMENU\n");
   printf("1) Insert element to queue\n");
   printf("2) Delete element from queue\n");
   printf("3) Display all the elements of queue\n");
   printf("4) Exit\n");
   do {
      printf("Enter your choice : ");
      scanf("%d",&ch);
      switch (ch) {
         case 1: Insert();
         break;
         case 2: Delete();
         break;
         case 3: Display();
         break;
         case 4: printf("Exit");
         break;
         default: printf("Invalid choice");
      }
   } while(ch!=4);
   return 0;
}
```

OUTPUT:


input
MENU
1) Insert element to queue
2) Delete element from queue
3) Display all the elements of queue
4) Exit
Enter your choice : 1
Insert the element in queue : 12
Enter your choice : 1
Insert the element in queue : 21
Enter your choice : 1
Insert the element in queue : 76
Enter your choice : 1
Insert the element in queue : 18
Enter your choice : 2
Element deleted from queue is : 12
Enter your choice : 2
Element deleted from queue is : 21
Enter your choice : 3
Queue elements are : 76 18
Enter your choice : 4
Exit
...Program finished with exit code 0
Press ENTER to exit console.

PROBLEM: 2

AIM: To Reversing the first K elements of a Queue

Input : Q = [10, 20, 30, 40, 50, 60,
70, 80, 90, 100]
k = 5

Source code:

Main.java | Reverse_k_element_queue.java

```java
package codeForces;

import java.util.LinkedList;
import java.util.Queue;
import java.util.Stack;

public class Reverse_k_element_queue {

    static Queue<Integer> queue;

    // Function to reverse the first
    // K elements of the Queue
    static void reverseQueueFirstKElements(int k)
    {
        if (queue.isEmpty() == true
                || k > queue.size())
            return;
        if (k <= 0)
            return;

        Stack<Integer> stack = new Stack<Integer>();

        // Push the first K elements into a Stack
        for (int i = 0; i < k; i++) {
            stack.push(queue.peek());
            queue.remove();
        }

        // Enqueue the contents of stack
        // at the back of the queue
        while (!stack.empty()) {
            queue.add(stack.peek());
            stack.pop();
        }

        // Remove the remaining elements and enqueue
        // them at the end of the Queue
        for (int i = 0; i < queue.size() - k; i++) {
            queue.add(queue.peek());
            queue.remove();
        }
    }

    // Utility Function to print the Queue
    static void Print()
    {
        while (!queue.isEmpty()) {
            System.out.print(queue.peek() + " ");
            queue.remove();
        }
    }

```

OUTPUT:


Run:
Reverse_k_element_queue
/Library/Java/JavaVirtualMachines/jdk-11.0.8.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/
50 40 30 20 10 60 70 80 90 100
Process finished with exit code 0