Name-Jeevan R

Section-I

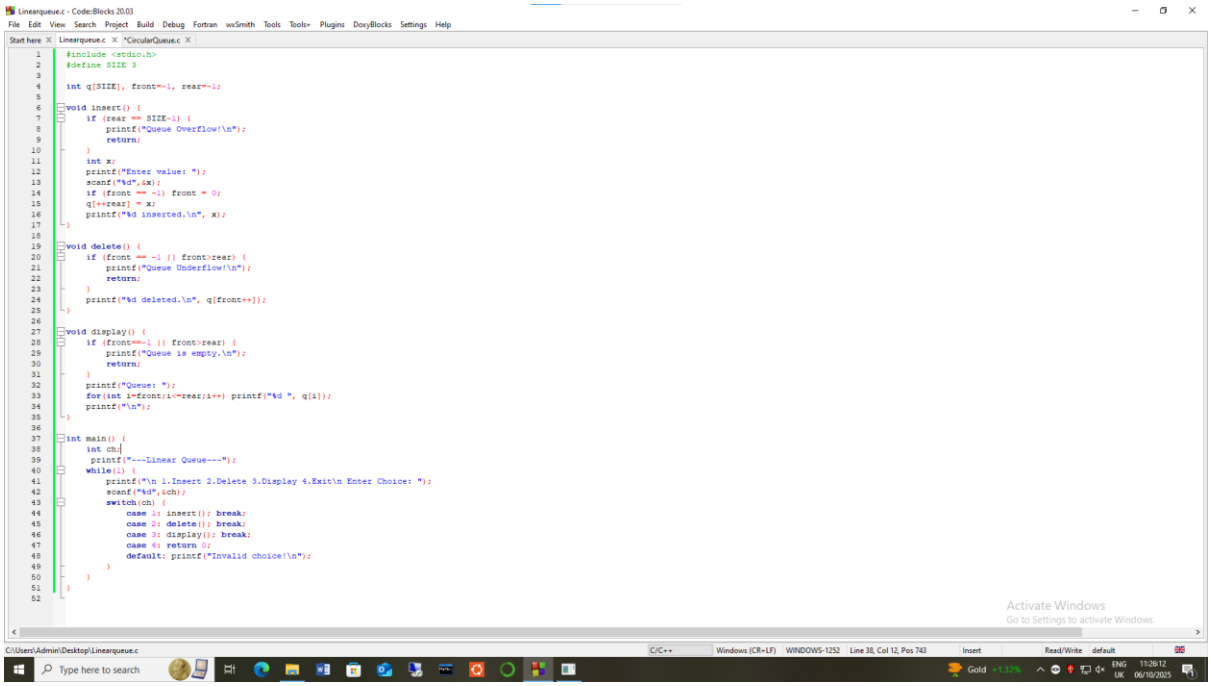
DSA: Lab Program-3

a) WAP to simulate the working of Linear Queue using an array with the following operations: Insert, Delete and Display, also should print appropriate message for queue empty and overflow conditions.

```
#include <stdio.h>
#define SIZE 3

int q[SIZE], front=-1, rear=-1;

void insert() {
   if (rear == SIZE-1) {
      printf("Queue Overflow!\n");
      return;
   }
   int x;
   printf("Enter value: ");
   scanf("%d",&x);
   if (front == -1) front = 0;
   q[++rear] = x;
   printf("%d inserted.\n", x);
}

void delete() {
   if (front == -1 || front>rear) {
      printf("Queue Underflow!\n");
      return;
   }
```

```
    printf("%d deleted.\n", q[front++]);
}

void display() {
    if (front==-1 || front>rear) {
        printf("Queue is empty.\n");
        return;
    }
    printf("Queue: ");
    for(int i=front;i<=rear;i++) printf("%d ", q[i]);
    printf("\n");
}

int main() {
    int ch;
     printf("---Linear Queue---");
    while(1) {
        printf("\n 1.Insert 2.Delete 3.Display 4.Exit\n Enter Choice: ");
        scanf("%d",&ch);
        switch(ch) {
            case 1: insert(); break;
            case 2: delete(); break;
            case 3: display(); break;
            case 4: return 0;
            default: printf("Invalid choice!\n");
        }
    }
}
```

Code and Expected Output:



Linearqueue.c - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Start here | Linearqueue.c | *CircularQueue.c

```c
#include <stdio.h>
#define SIZE 3

int q[SIZE], front=-1, rear=-1;

void insert() {
    if (rear == SIZE-1) {
        printf("Queue Overflow!\n");
        return;
    }
    int x;
    printf("Enter value: ");
    scanf("%d",&x);
    if (front == -1) front = 0;
    q[++rear] = x;
    printf("%d inserted.\n", x);
}

void delete() {
    if (front == -1 || front>rear) {
        printf("Queue Underflow!\n");
        return;
    }
    printf("%d deleted.\n", q[front++]);
}

void display() {
    if (front==-1 || front>rear) {
        printf("Queue is empty.\n");
        return;
    }
    printf("Queue: ");
    for(int i=front;i<=rear;i++) printf("%d ", q[i]);
    printf("\n");
}

int main() {
    int ch;
     printf("---Linear Queue---");
    while(1) {
        printf("\n 1.Insert 2.Delete 3.Display 4.Exit\n Enter Choice: ");
        scanf("%d",&ch);
        switch(ch) {
            case 1: insert(); break;
            case 2: delete(); break;
            case 3: display(); break;
            case 4: return 0;
            default: printf("Invalid choice!\n");
        }
    }
}
```

C:\Users\Admin\Desktop\Linearqueue.exe

```
---Linear Queue---
 1.Insert 2.Delete 3.Display 4.Exit
 Enter Choice: 3
Queue is empty.

 1.Insert 2.Delete 3.Display 4.Exit
 Enter Choice: 1
Enter value: 10
10 inserted.

 1.Insert 2.Delete 3.Display 4.Exit
 Enter Choice: 1
Enter value: 20
20 inserted.

 1.Insert 2.Delete 3.Display 4.Exit
 Enter Choice: 1
Enter value: 30
30 inserted.

 1.Insert 2.Delete 3.Display 4.Exit
 Enter Choice: 1
Queue Overflow!

 1.Insert 2.Delete 3.Display 4.Exit
 Enter Choice: 3
Queue: 10 20 30

 1.Insert 2.Delete 3.Display 4.Exit
 Enter Choice: 2
10 deleted.

 1.Insert 2.Delete 3.Display 4.Exit
 Enter Choice: 3
Queue: 20 30

 1.Insert 2.Delete 3.Display 4.Exit
 Enter Choice: 2
20 deleted.

 1.Insert 2.Delete 3.Display 4.Exit
 Enter Choice: 2
30 deleted.

 1.Insert 2.Delete 3.Display 4.Exit
 Enter Choice: 2
Queue Underflow!

 1.Insert 2.Delete 3.Display 4.Exit
 Enter Choice: 3
Queue is empty.

 1.Insert 2.Delete 3.Display 4.Exit
 Enter Choice: 1
Queue Overflow!

 1.Insert 2.Delete 3.Display 4.Exit
 Enter Choice: 4

Process returned 0 (0x0)   execution time : 64.044 s
Press any key to continue.
```