

Laboratory 6

1. Questions

1. Implement the INSERT, DELETE and PRINT operations on queue.
2. Implement a priority queue using suitable application.

2. Algorithm

1. Algorithm to Implement the INSERT, DELETE and PRINT operations on queue.
 1. Ask the user for the operation like insert, delete, display and exit.
 2. According to the option entered, access its respective function using switch statement. Use the variables front and rear to represent the first and last element of the queue.
 3. In the function insert(), firstly check if the queue is full. If it is, then print the output as “Queue Overflow”. Otherwise take the number to be inserted as input and store it in the variable add_item. Copy the variable add_item to the array queue_array[] and increment the variable rear by 1.
 4. In the function delete(), firstly check if the queue is empty. If it is, then print the output as “Queue Underflow”. Otherwise print the first element of the array queue_array[] and decrement the variable front by 1.
 5. In the function display(), using for loop print all the elements of the array starting from front to rear.
 6. Exit.
2. Algorithm to Implement a priority queue using suitable application.

Let head be a pointer that stores the address of first node

step 1: Read the value and its priority

step 2: create a new node using malloc function

step 3: Assign the value read to the data field of newnode (newnode -> data =value)
and newnode -> pri = priority

step 4: store the address of node whose priority is > newnode priority in pointer c

step 5: store the address of node before c in pointer p

step 5: newnode -> next = p->next

step 6: p -> next = newnode

3. Program

```
1  /*program done by Deepak R
2  * C Program to Implement a Queue using an Array
3  */
4  #include <stdio.h>
5  #define MAX 50
6  void insert();
7  void delete();
8  void display();
9  int queue_array[MAX];
10 int rear = - 1;
11 int front = - 1;
12 main()
13 {
14     int choice;
15     while (1)
16     {
17         printf("1.Insert element to queue \n");
18         printf("2.Delete element from queue \n");
19         printf("3.Display all elements of queue \n");
20         printf("4.Quit \n");
21         printf("Enter your choice : ");
22         scanf("%d", &choice);
23         switch (choice)
24         {
25             case 1:
26                 insert();
27                 break;
28             case 2:
29                 delete();
30                 break;
31             case 3:
32                 display();
```

```

31             case 3:
32                 display();
33                 break;
34             case 4:
35                 exit(1);
36             default:
37                 printf("Wrong choice \n");
38         } /* End of switch */
39     } /* End of while */
40 } /* End of main() */
41
42 void insert()
43 {
44     int add_item;
45     if (rear == MAX - 1)
46         printf("Queue Overflow \n");
47     else
48     {
49         if (front == - 1)
50             /*If queue is initially empty */
51             front = 0;
52         printf("Inset the element in queue : ");
53         scanf("%d", &add_item);
54         rear = rear + 1;
55         queue_array[rear] = add_item;
56     }
57 } /* End of insert() */
58
59 void delete()
60 {
61     if (front == - 1 || front > rear)
62     {
```

```
61     if (front == -1 || front > rear)
62     {
63         printf("Queue Underflow \n");
64         return ;
65     }
66     else
67     {
68         printf("Element deleted from queue is : %d\n", queue_array[front]);
69         front = front + 1;
70     }
71 } /* End of delete() */
72
73 void display()
74 {
75     int i;
76     if (front == - 1)
77         printf("Queue is empty \n");
78     else
79     {
80         printf("Queue is : \n");
81         for (i = front; i <= rear; i++)
82             printf("%d ", queue_array[i]);
83         printf("\n");
84     }
85 } /* End of display() */
```

Fig 1 program to Implement the INSERT, DELETE and PRINT operations on queue.

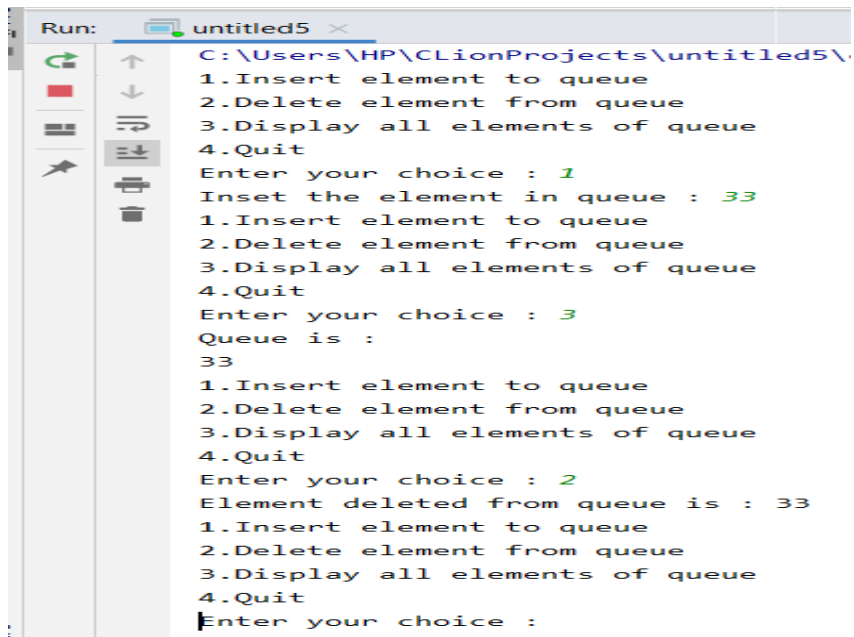
```

CMakeLists.txt × main.c ×
1  #include<stdio.h>
2  #include <stdlib.h>
3  struct node
4  {
5      int data,pri;
6      struct node *next;
7  };
8  struct node *head=NULL,*c,*p;
9  void create();
10 void display();
11 main()
12 {
13     int n,i;
14     printf("enter the number of elements");
15     scanf("%d",&n);
16     for(i=1;i<=n;i++)
17         create();
18     display();
19 }
20 void create()
21 {
22     int v,priority;
23     printf("enter value and priority\n");
24     scanf("%d%d",&v,&priority);
25     struct node *newnode = (struct node *)malloc(sizeof(struct node));
26     newnode->data =v;
27     newnode->pri=priority;
28     newnode->next = NULL;
29     if(head == NULL)
30         head = newnode;
31     else if( newnode->pri < head->pri)
32     {
33         newnode->next=head;
34         head=newnode;
35     }
36     else
37     {
38         c=head;
39         while(newnode->pri >=c->pri && c->next != NULL)
40         {
41             p=c;
42             c=c->next;
43         }
44         if(c->next == NULL && newnode->pri >= c->pri)
45         {
46             c->next = newnode;
47         }
48         else
49         {
50             p->next=newnode;
51             newnode->next=c;
52         }
53     }
54 }
55 void display()
56 {
57     if(head == NULL)
58         printf("list is empty");
59     else
60     {
61         c=head;
62         while(c != NULL)
63         {
64             printf("%d  %d->",c->data,c->pri);
65             c=c->next;
66         }
67     }
68 }
69
70

```

Fig 2 program to Implement a priority queue using suitable application.

4. Presentation of Results

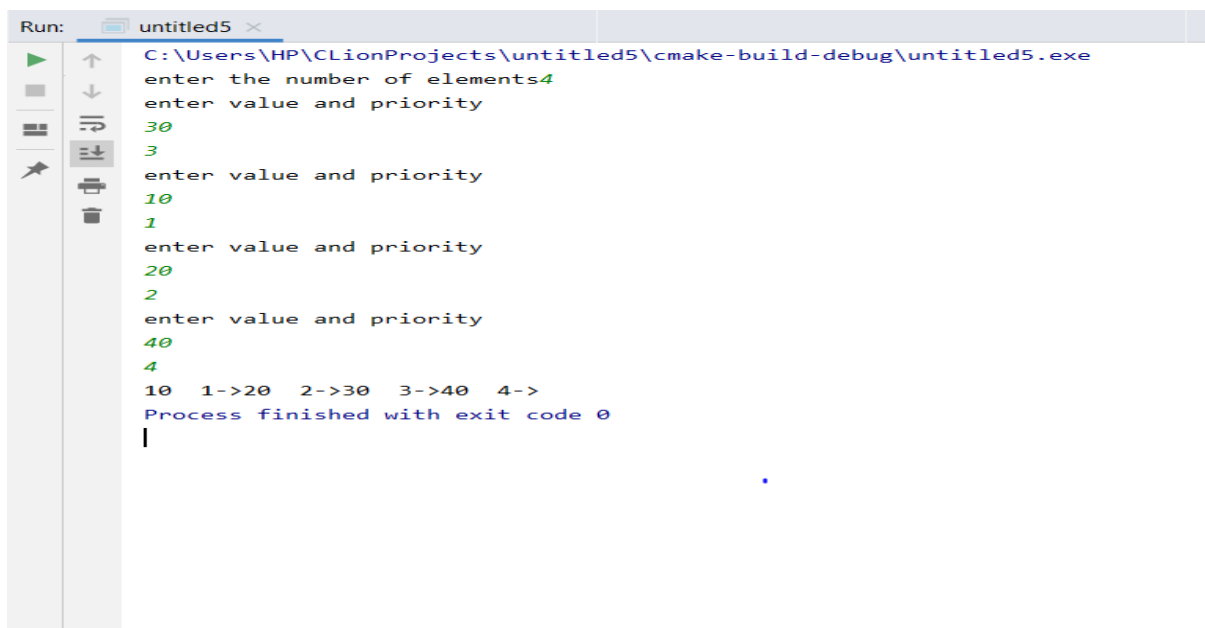


```

Run: untitled5 x
C:\Users\HP\CLionProjects\untitled5\
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 1
Inset the element in queue : 33
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 3
Queue is :
33
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 2
Element deleted from queue is : 33
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice :

```

Fig 3 Result of program to Implement the INSERT, DELETE and PRINT operations on queue.



```

Run: untitled5 x
C:\Users\HP\CLionProjects\untitled5\cmake-build-debug\untitled5.exe
enter the number of elements4
enter value and priority
30
3
enter value and priority
10
1
enter value and priority
20
2
enter value and priority
40
4
10 1->20 2->30 3->40 4->
Process finished with exit code 0
|

```

Fig 4 Result of program to Implement a priority queue using suitable application.

5. Conclusions

In this lab we learnt how to Implement the INSERT, DELETE and PRINT operations on queue.and to Implement a priority queue using suitable application.