

Name : Bhushan Sharad Tejankar

Roll no. : I30

reg\_no. : 2020BIT030

```
1  #include <iostream>
2  using namespace std;
3  struct Node {
4      int data;
5      struct Node *next;
6  };
7  struct Node* top = NULL;
8  void push(int val) {
9      struct Node* newnode = (struct Node*) malloc(sizeof(struct Node));
10     newnode->data = val;
11     newnode->next = top;
12     top = newnode;
13 }
14 void pop() {
15     if(top==NULL)
16         cout<<"Stack Underflow"<<endl;
17     else {
18         cout<<"The popped element is "<< top->data <<endl;
19         top = top->next;
20     }
21 }
22 void display() {
23     struct Node* ptr;
24     if(top==NULL)
25         cout<<"stack is empty";
26     else {
```

```
27     ptr = top;
28     cout<<"Stack elements are: ";
29     while (ptr != NULL) {
30         cout<< ptr->data <<" ";
31         ptr = ptr->next;
32     }
33 }
34 cout<<endl;
35 }
36 int main() {
37     int ch, val;
38     cout<<"1) Push in stack"<<endl;
39     cout<<"2) Pop from stack"<<endl;
40     cout<<"3) Display stack"<<endl;
41     cout<<"4) Exit"<<endl;
42     do {
43         cout<<"Enter choice: "<<endl;
44         cin>>ch;
45         switch(ch) {
46             case 1: {
47                 cout<<"Enter value to be pushed:"<<endl;
48                 cin>>val;
49                 push(val);
50                 break;
51             }
52             case 2: {
```

```
53         pop();
54         break;
55     }
56     case 3: {
57         display();
58         break;
59     }
60     case 4: {
61         cout<<"Exit"<<endl;
62         break;
63     }
64     default: {
65         cout<<"Invalid Choice"<<endl;
66     }
67 }
68 }while(ch!=4);
69 return 0;
70 }
```

- 1) Push in stack
- 2) Pop from stack
- 3) Display stack
- 4) Exit

Enter choice:

1

Enter value to be pushed:

7

Enter choice:

3

Stack elements are: 7

Enter choice:



```
1 // 2. Queue using Linked List
2 #include <iostream>
3 using namespace std;
4 struct node {
5     int data;
6     struct node *next;
7 };
8 struct node* front = NULL;
9 struct node* rear = NULL;
10 struct node* temp;
11 void Insert() {
12     int val;
13     cout<<"Insert the element in queue : "<<endl;
14     cin>>val;
15     if (rear == NULL) {
16         rear = (struct node *)malloc(sizeof(struct node));
17         rear->next = NULL;
18         rear->data = val;
19         front = rear;
20     } else {
21         temp=(struct node *)malloc(sizeof(struct node));
22         rear->next = temp;
23         temp->data = val;
24         temp->next = NULL;
25         rear = temp;
26     }
```

```
27     }
28     void Delete() {
29         temp = front;
30         if (front == NULL) {
31             cout<<"Underflow"<<endl;
32             return;
33         }
34         else
35             if (temp->next != NULL) {
36                 temp = temp->next;
37                 cout<<"Element deleted from queue is : "<<front->data<<endl;
38                 free(front);
39                 front = temp;
40             } else {
41                 cout<<"Element deleted from queue is : "<<front->data<<endl;
42                 free(front);
43                 front = NULL;
44                 rear = NULL;
45             }
46     }
47     void Display() {
48         temp = front;
49         if ((front == NULL) && (rear == NULL)) {
50             cout<<"Queue is empty"<<endl;
51             return;
52         }
```



```
53     cout<<"Queue elements are: ";
54     while (temp != NULL) {
55         cout<<temp->data<<" ";
56         temp = temp->next;
57     }
58     cout<<endl;
59 }
60 int main() {
61     int ch;
62     cout<<"1) Insert element to queue"<<endl;
63     cout<<"2) Delete element from queue"<<endl;
64     cout<<"3) Display all the elements of queue"<<endl;
65     cout<<"4) Exit"<<endl;
66     do {
67         cout<<"Enter your choice : "<<endl;
68         cin>>ch;
69         switch (ch) {
70             case 1: Insert();
71                 break;
72             case 2: Delete();
73                 break;
74             case 3: Display();
75                 break;
76             case 4: cout<<"Exit"<<endl;
77                 break;
78             default: cout<<"Invalid choice"<<endl;
```



```
79         }  
80     } while(ch!=4);  
81     return 0;  
82 }
```

- 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 :

7

Enter your choice :

2

Element deleted from queue is : 7

Enter your choice :

3

Queue is empty

Enter your choice :

```
1 // 3. Doubly Linked List
2 #include <iostream>
3 using namespace std;
4 struct Node {
5     int data;
6     struct Node *prev;
7     struct Node *next;
8 };
9 struct Node* head = NULL;
10 void insert(int newdata) {
11     struct Node* newnode = (struct Node*) malloc(sizeof(struct Node));
12     newnode->data = newdata;
13     newnode->prev = NULL;
14     newnode->next = head;
15     if(head != NULL)
16         head->prev = newnode ;
17     head = newnode;
18 }
19 void display() {
20     struct Node* ptr;
21     ptr = head;
22     while(ptr != NULL) {
23         cout<< ptr->data <<" ";
24         ptr = ptr->next;
25     }
26 }
```

```
27  int main() {  
28      insert(3);  
29      insert(1);  
30      insert(7);  
31      insert(2);  
32      insert(9);  
33      cout<<"The doubly linked list is: ";  
34      display();  
35      return 0;  
36  }
```

The doubly linked list is: 9 2 7 1 3

○ PS C:\Users\91830\OneDrive\Desktop>

```
1 // 4,5. Enqueue and Dequeue
2 #include<iostream>
3 #include<conio.h>
4 #include<stdlib.h>
5 #define SIZE 5
6 using namespace std;
7 int q[SIZE],front=0,rear=0;
8 int main()
9 {
10 int ch;
11 void enqueue();
12 void dequeue();
13 void display();
14
15 while(1)
16 {
17 cout<<"\n 1. add element";
18 cout<<"\n 2. remove element";
19 cout<<"\n 3.display";
20 cout<<"\n 4.exit";
21 cout<<"\n enter your choice:";
22 cin>>ch;
23
24 switch(ch)
25 {
26 case 1:
```



```
27     enqueue();
28     break;
29     case 2:
30     dequeue();
31     break;
32     case 3:
33     display();
34     break;
35     case 4:
36     exit(0);
37     default:
38     cout<<"\n invalid choice";
39     }
40 }
41 }
42 void enqueue()
43 {
44     int no;
45     if (rear==SIZE && front==0)
46     cout<<"queue is full";
47     else
48     {
49     cout<<"enter the num:";
50     cin>>no;
51     q[rear]=no;
52 }
```

```
53     rear++;
54 }
55 void dequeue()
56 {
57     int no,i;
58     if (front==rear)
59         cout<<"queue is empty";
60     else
61     {
62         no=q[front];
63         front++;
64         cout<<"\n"<<no<<" -removed from the queue\n";
65     }
66 }
67 void display()
68 {
69     int i,temp=front;
70     if (front==rear)
71         cout<<"the queue is empty";
72     else
73     {
74         cout<<"\n element in the queue:";
75         for(i=temp;i<rear;i++)
76         {
77             cout<<q[i]<<" ";
78         }
79     }
80 }
81
82
```

```
1. add element
2. remove element
3.display
4.exit
enter your choice:1
enter the num:7
```

```
1. add element
2. remove element
3.display
4.exit
enter your choice:3
```

```
element in the queue:7
1. add element
2. remove element
3.display
4.exit
enter your choice:
```

THANK  
You!