

Gagan Jain

Roll no- 15

IT

CODE

```
#include <stdio.h>
#include <conio.h>

#define MAX 10

int deque[MAX];

int left = -1, right = -1;

void input_deque(void);
void output_deque(void);
void insert_left(void);
void insert_right(void);
void delete_left(void);
void delete_right(void);
void display(void);

int main()
{
    int option;

    // clrscr();

    printf("\n *****MAIN MENU*****");
    printf("\n 1.Input restricted deque");
    printf("\n 2.Output restricted deque");
    printf("Enter your option : ");

    scanf("%d",&option);

    switch(option)
    {
```

```

    case 1:
        input_deque();
        break;
    case 2:
        output_deque();
        break;
    }
    return 0;
}

void input_deque()
{
    int option;
    do
    {
        printf("\n INPUT RESTRICTED
        DEQUE"); printf("\n 1.Insert at right");
        printf("\n 2.Delete from left");
        printf("\n 3.Delete from
        right"); printf("\n 4.Display");
        printf("\n 5.Quit");
        printf("\n Enter your option :
        "); scanf("%d",&option);

        switch(option)
        {
            case 1:
                insert_right();
                break;

            case 2:
                delete_left();

                break;

            case 3:
                delete_right();

```

```

        break;

        case 4:

            display();

            break;

        }

        }while(option!=5);

    }

void output_deque()
{
    int option;

    do
    {
        printf("OUTPUT RESTRICTED
        DEQUE"); printf("\n 1.Insert at right");

        printf("\n 2.Insert at left");

        printf("\n 3.Delete from left");

        printf("\n 4.Display");

        printf("\n 5.Quit");

        printf("\n Enter your option :
        "); scanf("%d",&option);

        switch(option)
        {
            case 1:

                insert_right();

                break;

            case 2:

                insert_left();

                break;

            case 3:

                delete_left();

                break;

```

```

case 4:
display();
break;
}
}while(option!=5);
}
void insert_right()
{
int val;
printf("\n Enter the value to be
added:"); scanf("%d", &val);
if((left == 0 && right == MAX-1) || (left == right+1))
{
printf("\n OVERFLOW");
return;
}
if (left == -1) /* if queue is initially empty
*/ {
left = 0;
right = 0;
}
else
{
if(right == MAX-1) /*right is at last position of queue
*/ right = 0;
else
right = right+1;
}
deque[right] = val ;
}
void insert_left()

```

```

{
int val;

printf("\n Enter the value to be added:");

scanf("%d", &val);

if((left == 0 && right == MAX-1) || (left == right+1))
{
printf("\n Overflow");

return;
}

```

```

if (left == -1)/*If queue is initially empty*/

```

```

left = 0;

right = 0;

}

else

{

if(left == 0)

left=MAX-1;

else

left=left-1;

}

deque[left] = val;

}

void delete_left()

{

if (left == -1)

{

printf("\n UNDERFLOW");

return ;

}

printf("\n The deleted element is : %d", deque[left]);

if(left == right) /*Queue has only one element */ {

```

```

left = -1;
right = -1;
}
else
{
if(left == MAX-1)
left = 0;
else
left = left+1;
}
}

void delete_right()
{
if (left == -1)
{
printf("\n UNDERFLOW");
return ;
}

printf("\n The element deleted is : %d", deque[right]);
if(left == right) /*queue has only one element*/ {
left = -1;
right = -1;
}
else
{
if(right == 0)
right=MAX-1;
else
right=right-1;
}
}

void display()
{
int front = left, rear = right;

```

```
if(front == -1)
{
printf("\n QUEUE IS EMPTY");
return;
}
printf("\n The elements of the queue are : ");
```

```
if(front <= rear )
{
while(front <= rear)
{
printf("%d",deque[front]);
front++;
}
}
else
{
while(front <= MAX-1)
{
printf("%d", deque[front]);
front++;
}
front = 0;
while(front <= rear)
{
printf("%d",deque[front]);
front++;
}
}
printf("\n");
}
```

SCREENSHOTS

```

101     }
102     }while(option!=5);
103     }
104     void insert_right()
105     {
106     int val;
107     printf("\n Enter the value to be added:");
108     scanf("%d", &val);
109     if((left == 0 && right == MAX) || (left == right+1))
110     {
111     printf("\n Overflow");
112     return;
113     }
114     if (left == -1) /* If queue is initially empty */
115     {
116     left = 0;
117     right = 0;
118     }
119     else
120     {
121     if(right == MAX-1) /*right is at last position of queue */
122     right = 0;
123     else
124     right = right+1;
125     }
126     deque[right] = val ;
127     }
128     void insert_left()
129     {
130     int val;
131     printf("\n Enter the value to be added:");
132     scanf("%d", &val);

```

```

133     }
134     }
135     }
136     void output_deque()
137     {
138     int option;
139     do
140     {
141     printf("OUTPUT RESTRICTED DEQUE");
142     printf("\n 1.Insert at right");
143     printf("\n 2.Insert at left");
144     printf("\n 3.Delete from left");
145     printf("\n 4.Display");
146     printf("\n 5.Quit");
147     printf("\n Enter your option : ");
148     scanf("%d",&option);
149     switch(option)
150     {
151     case 1:
152     insert_right();
153     break;
154     case 2:
155     insert_left();
156     break;
157     case 3:
158     delete_left();
159     break;
160     case 4:
161     display();
162     break;
163     }
164     }while(option!=5);
165     }

```

```

166     }
167     }
168     }
169     void delete_left()
170     {
171     int val;
172     printf("\n Enter the value to be added:");
173     scanf("%d", &val);
174     if((left == 0 && right == MAX) || (left == right+1))
175     {
176     printf("\n Overflow");
177     return;
178     }
179     if (left == -1)/*If queue is initially empty*/
180     {
181     left = 0;
182     right = 0;
183     }
184     else
185     {
186     if(left == 0)
187     left=MAX-1;
188     else
189     left=left-1;
190     }
191     deque[left] = val;
192     }
193     void delete_left()
194     {
195     if (left == -1)
196     {
197     printf("\n Underflow");
198     return ;
199     }
200     printf("\n The deleted element is : %d", deque[left]);
201     if(left == MAX-1) //When the only one element is

```



```

}
}
printf("The deleted element is : %d", deque[left]);
if(left == right) /*Queue has only one element */
{
left = -1;
right = -1;
}
else
{
if(left == MAX-1)
left = 0;
else
left = left+1;
}
}
void delete_right()
{
if (left == -1)
{
printf("UNDERFLOW");
return ;
}
printf("The element deleted is : %d", deque[right]);
if(left == right) /*queue has only one element*/
{
left = -1;
right = -1;
}
else
{
if(right == 0)

```

OUTPUT

```

Microsoft PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.
****Queue deque****
1.Input restricted deque
2.Output restricted deque enter your option : 1

INPUT RESTRICTED DEQUE
1.Insert at right
2.Delete from left
3.Delete from right
4.Display
5.Quit
Enter your option : 1

Enter the value to be added at

INPUT RESTRICTED DEQUE
1.Insert at right
2.Delete from left
3.Delete from right
4.Display
5.Quit
Enter your option : 4

The elements of the queue are : 45

INPUT RESTRICTED DEQUE
1.Insert at right
2.Delete from left
3.Delete from right
4.Display
5.Quit
Enter your option : 5
PS C:\Users\vaani\Desktop> c:\programming\

```