

08/01/2024

### Prog - 03.

03. Write a program to stimulate the working of the queue of integers. Using array provide the following operations: insert, delete, display. The program should print appropriate message for overflow and underflow condition.

=> #include <stdio.h>

#include <conio.h>

#define MAX 10

int queue[MAX];

int front = -1, rear = -1;

void insert(void);

int delete\_element(void);

int peek(void);

void display(void);

int main()

{ int option, val;

do

{

printf ("In In \*\* Main Menu \* ");

printf ("In 1. Insert an element");

printf ("In 3. peek");

printf ("In 4. Display the Queue");

printf ("In 5. Exit");

printf ("In Enter your Option:");

scanf ("%d", &option);

switch (option)

{

Case 1 :

Insert();

break;

08/01/2024

case 2 :

```
val = delete_element();
if (val != -1)
printf ("In The number deleted is : %d", val);
break;
```

case 3 :

```
val = peek();
if (val == -1)
printf ("In The first value in queue is : %d", val);
break;
```

case 4 :

```
display();
break;
```

}

```
while (option != 5);
getch();
return 0;
}
```

void insert ()

```
{
int num;
printf ("In Enter the number to be inserted in the queue:");
scanf ("%d", &num);
if (rear == MAX-1)
printf ("In overflow");
else if (front == -1 && rear == -1)
front = rear = 0;
else
rear++;
queue [rear] = num;
}
```

8/01/2024

```
int delete_element()
{
    int val;
    if (front == -1 || front > rear)
    {
        printf("In UNDERFLOW");
        return -1;
    }
    else
    {
        val = queue[front];
        front++;
        if (front > rear)
            front = rear = -1;
        return val;
    }
}

int peek()
{
    if (front == -1 || front > rear)
        printf("In QUEUE is EMPTY");
    return -1;
}

else
{
    return queue[front];
}

void display()
{
    int i = front + 1;
    for (i = front + 1; i <= rear; i++)
        printf("%d ", queue[i]);
}
```

08/01/2024

```
< value> = 6 previous set statement of menu is still there  
int i; friend setting global, it makes local to print  
printf("In"); queue storage long time because of  
& (front == -1 || front > rear)  
printf("In QUEUE IS EMPTY");  
else  
{  
    for (i = front; i <= rear; i++)  
        printf("It %d. d", queue[i]);  
}  
y  
3
```

### Output

- \*\* MAIN MENU \*\*
- 1. Insert an element
- 2. Delete an element
- 3. peek
- 4. Display the Queue
- 5. Exit

Enter Your Option : 4

Enter the number to be inserted in the queue : 20

```
****MAIN MENU****
1.Insert an element
2.Delete an element
3.Peek
4.Display the queue
5.Exit
Enter your option:1
```

```
Enter the number to be inserted in the queue:10
```

```
****MAIN MENU****
1.Insert an element
2.Delete an element
3.Peek
4.Display the queue
5.Exit
Enter your option:1
```

```
Enter the number to be inserted in the queue:20
```

```
****MAIN MENU****
1.Insert an element
2.Delete an element
3.Peek
4.Display the queue
5.Exit
Enter your option:1
```

```
Enter the number to be inserted in the queue:30
```

```
****MAIN MENU****
1.Insert an element
2.Delete an element
3.Peek
4.Display the queue
5.Exit
Enter your option:1
```

```
Enter the number to be inserted in the queue:25
```

```
OVERFLOW
```

```
****MAIN MENU****
1.Insert an element
2.Delete an element
3.Peek
4.Display the queue
5.Exit
Enter your option:2
```

```
The number deleted is :10
```

```
****MAIN MENU****
1.Insert an element
2.Delete an element
3.Peek
4.Display the queue
5.Exit
Enter your option:3
```

```
The first value in queue is:20
```

```
****MAIN MENU****
1.Insert an element
2.Delete an element
3.Peek
4.Display the queue
5.Exit
```

08/01/2024

### Prog-04.

04. Write a program to simulate the working of a circular queue using an array. provide the following operation; insert, delete & display. The program should print appropriate message for queue empty and queue overflow condition.

=>

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

```
# include <ratio.h>
```

```
# include <process.h>
```

```
# define SIZE 3
```

```
int item, front = 0, rear = -1, q[10], count = 0;
```

```
void insert rear()
```

```
{
```

```
if (count == QUE-SIZE)
```

```
{
```

```
printf ("Queue overflow\n");
```

```
return;
```

```
}
```

```
rear = (rear + 1) % QUE-SIZE;
```

```
q[rear] = item;
```

```
count++;
```

```
}
```

```
int delete front()
```

```
{
```

```
if (count == 0) return -1;
```

~~item = q[front];~~~~front = (front + 1) % QUE-SIZE;~~~~count = count - 1;~~~~return item;~~

```
}
```

```
void display()
```

```
{
```

03/01/2024

```
int r, f;
if (count == 0)
{
    printf ("Queue is empty\n");
    return;
}
f = front;
printf ("contents of Queue\n");
for (i=1; i<=count; i++)
{
    printf ("%d\n", q[i]);
    j = (j+1) % QUEUE_SIZE;
}
```

```
void main()
```

```
{
```

```
int choice;
```

```
clrscr();
```

```
for (;;)
```

```
{
```

```
printf ("1: insert rear\n2: delete front\n3: display\n4: exit\n");
```

```
printf ("enter the choice\n");
```

```
scanf ("%d", &choice);
```

```
switch (choice)
```

```
{
```

~~case 1: printf ("enter the item to be inserted\n");~~

```
scanf ("%d", &item);
```

```
insertrear ();
```

```
break;
```

03/01/2024

Case 2: if (m = deletefront());  
if (item == -1)  
print ("Character queue is empty in");

else

print ("item deleted = " + dln, item);  
break;

Case 3: display (Q);

break;

default: exit (0);

)

### Output

Enter

1. Insert rear
2. delete front
3. display
4. Exit 2

Underflow

Enter

1. Insert rear
2. delete front
3. display
4. Exit 1

Enter value:

3

Value inserted

08/01/2024

Enter

1. Insert rear
2. delete front

3. display

4. Exit 1

Enter value 6

value inserted

Enter

1. Insert rear

2. delete front

3. display

4. Exit 1

Enter value 9

value inserted

Enter

1. Insert rear

2. delete front

3. display

4. Exit 1

Enter value 12

OVERFLOW

Enter

1. Insert rear

2. delete front

3. display

4. Exit 1

3

6

9.

Enter

1. Insert rear

2. delete front

3. display

4. Exit 2

9

```
1:insertrear  
2:deletefront  
3:display  
4:exit  
enter the choice  
1  
enter the item to be inserted  
10

1:insertrear  
2:deletefront  
3:display  
4:exit  
enter the choice  
1  
enter the item to be inserted  
20

1:insertrear  
2:deletefront  
3:display  
4:exit  
enter the choice  
2  
item deleted =10

1:insertrear  
2:deletefront  
3:display  
4:exit  
enter the choice  
3  
Contents of queue  
20

1:insertrear  
2:deletefront  
3:display  
4:exit  
enter the choice  
4
```

Process returned 0 (0x0) execution time : 22.206 s  
Press any key to continue.