

11-1-24

## Week - 3

classmate

Date \_\_\_\_\_  
Page \_\_\_\_\_

(3b) WAP to simulate the working of a circular queue of integers using an array.

#include &lt;stdio.h&gt;

#define ~~MA~~ SIZE 10

int items[SIZE];

int front = -1, rear = -1;

int isFull()

{ if ((front == rear + 1) || (front == 0 &amp;&amp; rear == SIZE - 1))

return 1;

return 0;

}

int isEmpty()

{ if (front == -1)

return 1;

return 0;

}

void enqueue(int element)

{ if (isFull())

printf("Queue is full");

{ else {

if (front == -1)

front = 0;

rear = (rear + 1) % SIZE;

items[rear] = element;

printf("Inserted %d", element);

}

}

int dequeue()

{ int element;

if (isEmpty())

printf("Queue is empty");

return -1;

```

else
{
    element = items[front];
    if (front == rear)
        front = -1;
        rear = -1;
}
else {
    front = (front + 1) % SIZE;
}
printf("Deleted element %d", element);
return (element);
}
}

```

```

void display()
{
    int i;
    if (isEmpty())
        printf("Empty queue");
    else
        printf("Front = %d", front);
        printf("Items ");
        for (i = front; i != rear; i = (i + 1) % SIZE)
            printf("%d", items[i]);
        printf("%d", items[i]);
        printf("Rear = %d", rear);
}

```

```

void main()
{
    int option, val;
    do {
        printf("1. Insert\n");
        printf("2. Delete\n");
        printf("3. Display\n");
        printf("4. Exit\n");
    }
}

```



```

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

```

```

    case 1: insert enqueue();
            break;

```

```

    case 2: val = dequeue();
            if(val != -1)
                printf("the number deleted is: %d", val);
            break;

```

```

    case 3: display();
            break;

```

```

}
while(option != 4);

```

```

}

```

Output:

1. Insert

2. Delete

3. Display

4. Exit

Enter your option: 1

enter the element: 1

Inserted → 1

1. Insert

2. Delete

3. Display

4. Exit

Enter your option: 1

enter the element: 2

Inserted → 2

1. Insert

2. Delete

3. Display

4. Exit

Enter your option: 2

Deleted element → 1

the number deleted is: 1

1. Insert

2. Delete

3. Display

4. Exit

Enter your option: 3

Front → 1

Back → 2

Rear → 1

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

(Q) WAP to show implementation of insert and display using singly linked list.

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

```
#include <stdlib.h>
```

```
struct Node
```

```
{
```

```
    int data;
```

```
    struct Node *next;
```

```
};
```

```
void insert (struct Node **head, int data)
```

```
{
```

```
    struct Node *newnode = (struct Node *) malloc (sizeof (struct Node));
```

```
    newnode →
```

```
    data = data;
```

```
    newnode →
```

```
    next = *head;
```

```
    *head = newnode;
```

```
}
```

```
void display (struct Node *node)
```

```
{
```

```
    printf ("Linked list = ");
```

```
    while (node != NULL)
```

```
    {
```

```
        printf ("%d", node → data);
```

```
        node = node → next;
```

```
    }
```

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

```
}
```



```
void main ()  
{  
    struct Node *head = NULL;  
    insert(&head, 100);  
    insert(&head, 80);  
    insert(&head, 60);  
    insert(&head, 40);  
    insert(&head, 20);  
    display(head);  
}
```

Output:

Linked list : 20 40 60 80 100

S.P.  
11/11/24