

08/01/2024

WEEK - 4

1) Write a program to simulate the working of the queue of integers using an array. Provide the following operations:- Insert, delete, display. The program should print appropriate message for overflow & underflow condition.

```
pr    z/p    #include <stdio.h>
          #include <stdlib.h>
          #define MAX 5
          int queue_array[MAX];
          int rear = -1;
          int front = -1;
          void display();
          void insert();
          void delete();
          void main() {
              int is choice;
              if (front == -1) {
              while (1) {
                  printf("1. Insert\n 2. Delete\n 3. Display\n 4. Exit\n");
                  printf("Enter your choice: ");
                  scanf("%d", &choice);
```

switch(choice)

```
{
    case 1: insert();
            break;
    case 2: delete();
            break;
    case 3: display();
            break;
    case 4: exit(0);
    default: printf("Invalid choice");
}
```

void insert()

```
{
    int add_item;
    if (rear == max-1)
        printf("Que overflow\n");
    else {
        if (front == -1)
            front = 0;
        printf("Insert the element in queue: ");
        scanf("%d", &add_item);
        rear = rear + 1;
        queue_array[rear] = add_item;
    }
}
```

void delete()

```
{
    if (front == -1 || front > rear)
    {
        printf("queue underflow\n");
        return;
    }
    else {
        printf("Deleted element is: %d\n",
               queue_array[front]);
        front = front + 1;
    }
}
```

void display()

```
{
    int i;
    if (front == -1)
        printf("Queue is empty");
    else {
        printf("queue is: ");
        for (i = front; i <= rear; i++)
            printf("%d ", queue_array[i]);
        printf("\n");
    }
}
```

- 2) Write a program to simulate working of a circular queue using an array. provide the following operations: insert, delete, display. The program should print appropriate message for queue empty & queue over flow conditions.

Pt

1/1

int: ~~arr~~ eq[Size], rear = -1, front = -1;

Size is size of array

int isfull()

```
{
    if ((front == rear + 1) || (front == 0 & rear == Size - 1))
        return 1;
    return 0;
}
```

int empty()

```
{
    if (front == -1)
        return 1;
    return 0;
}
```

void insert (int a)

```
{
    if (isfull())
        printf("Over flow\n");
}
```



```

else {
    if (front == -1)
        front = 0;
    rear = (rear + 1) % size;
    rear cq[rear] = a;
    printf("%d is inserted", a);
}

```

```

}
}

```

```

int delete()

```

```

{
    int value;
    if (empty())
    {
        printf("Underflow\n");
        return -1;
    }
    else
    {
        value = cq[front];
        if (front == rear)
        {
            front = -1;
            rear = -1;
        }
        else
        {
            front = (front + 1) % size;
        }
        return value;
    }
}

```

```

void display()

```

```

{
    if (empty())
    {
        printf("Empty - Underflow\n");
    }
}

```

```

else { printf("In front: %d\n", front);

```

```

    for (i = front; i < rear; i = (i + 1) % size)
        printf("It %d\n", cq[i]);
}

```

```

printf("%d\n", cq[i]);
}

```

```

}

```

Ans  
8/11/24

1. Insert
2. Delete
3. Display
4. Exit

Enter choice: 2

Queue underflow

- 2) 1. Insert 2. Delete 3. display 4. exit

Enter choice:

1 elements to be inserted:

2

2

3

4

5

Enter choice: 3

2 2 3 4 5

Enter choice: 2

2 is deleted

Enter choice: 1

element to be inserted: 6

Enter choice: 3

2 3 4 5 6