08/01/2024

Queues

1. Linear Queue:

```
#include<stdio.h>
#define queue[N];
  int front=-1, rear=-1;

void enqueue(int x)
{
   if(rear==N-1)
   {
    Printf("Overflow");
   }
  else{
      rear++;
       queue[rear]=x;
  }  }
void deque()
{  if(front==-1 && rear==-1)
   {  printf("Underflow");
   }
  else{
      printf("%d", queue[front]);
      front++;
   }
}
void display()
{  if(front==-1 && rear==-1)
   {  Printf("Underflow");
   }
```

```
else
{ printf("Queue Contains");
  for(int i=front; i<rear+1; i++)
  { printf("%d", queue[i]);
  }
}
}
int main()
{
 enqueue(10);
 enqueue(20);
 enqueue(30);
 display Queue();
 int deqed = dequeue();
Printf("Deqed item: %d", deqed);
 display Queue();
 return 0;
}
```

Output:

Queue contains.

10
20
30

Queue contains

20
30

08/01/2024

2. Circular Queues:

```
#include<stdio.h>
#define N=5
 int queue[5];
 int front=-1, rear=-1;

void enqueue(int x)
{
 for(front==-1 && rear==-1)
 {
  front=rear=0;
   queue[rear]=x;
 }
else if((rear+1)%N==front)
   {
     printf("Queue is full");
   }
else{
    rear=(rear+1)%N;
    queue[rear]=x;
  }
} void dequeue()
 {
   if(front==-1 && rear==-1)
   { printf("underflow");
   }
  else if(front==rear)
  { front=rear=-1;
  }
 else
   { printf("%d", q[front]);
    front=(front+1) %N;
}}
```

```
void display()
{ if(front==-1 && rear==-1)
  { printf("Underflow");
  }
  else{
    printf("Queue Contains");
      int i=front;
    while(i!=rear)
    { Printf("%d",queue[i]);
        i=(i+1)%N;
    }
    printf("%d",queue[rear]);
  }
int main()
{  enqueue(10);
   enqueue(20);
   enqueue(30);
   display Queue();
   int dequed=dequeue();
   printf("Dequed item: %d",dequed);
   display Queue();
   enqueue(40);
   enqueue(50);
   display Queue;
   return 0;
}
```

Output:

Queue contains.
10
20
30
Queue contains
20
30

Queue Contains
20
30
40
50