8/01/2024

Linear Queue:

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

void enqueue(char a)
{
    if(rear==size-1)
        printf("Overflow");
    else
    {
        if(front==-1)
            front=0;
        queue[++rear]=a;
    }
}

char dequeue()
{
    if(front==-1 || front>rear)
        printf("underflow");
    else
    {
        char s=queue[front];
        if(front==rear)
            front=rear=-1;
        else
        {
            front++;
        }
        return s;
    }
}

void display()
{
    if(front==-1 || front>rear)
        printf("underflow");
    else
    {
        for(i=rear; i>=0; i--)
        {
```

```
        printf("%d", queue[i]);
    }
}
```

Output:

```
------ Menu ------
1. enqueue
2. dequeue
3. display
4. exit
enter your choice: 1
enter a value: 32
enter your choice: 1
enter a value: 45
enter your choice: 32
45
enter your choice: 2
enter your choice: 3
45
enter your choice: 4
it's the queue.
```

Circular queue

```
#include<stdio.h>
#include<stdlib.h>
# define size 5

int items[size], rear=-1, front=-1;

int isfull()
{
    if ((front==rear-1)||(front==0 && 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("%d is inserted", element);
    }
}

int dequeue()
{
    int value;
    if(isempty())
        printf("queue is empty") return -1;
```

Sw 8/11/24

```
{
    if(front == -1)
        front = 0;
    rear
    value = item[front];
    if (front == rear)
    {
        front = -1;
        rear = -1;
    }
    else
    {
        front = (front-1)%size;
    }
    return value;
}

void display()
{
    int i;
    if(isempty())
        printf("queue is empty");
    else
    {
        printf("front position=%d\n", front);
        for (i = front; i != rear; i = (i+1)%size)
        {
            printf("%d ", item[i]);
        }
        printf("%d", items[i]);
    }
}
```

output:

--- Menu ---

enter 1. enqueue
2. dequeue
3. display
4. exit

enter your choice: 1

enter your choice : 3

3 1

enter your choice : 2

enter your choice : 4.

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

void enqueue()
{
    int a;
    if (rear==size-1)
    {
        printf("overflow\n");
    }
    else
    {
        if(front==-1)
            front=0;
        printf("enter a element:");
        scanf("%d",&a);
        queue[++rear]=a;
    }
}

int dequeue()
{
    if(front==-1||front>rear)
    {
        printf("underflow\n");
    }
    else
    {
        int s = queue[front];
        ++front;
        if (front==rear)
        {
            front=-1;
            rear=-1;
        }
```

```c
        }
        return s;
    }
}

void display()
{
    if(front==-1)
    {
        printf("overflow\n");
    }
    else
    {
        for (int i=front;i<=rear;i++)
        {
            printf("%d\n",queue[i]);
        }
    }
}

int main()
{
    int choice;

    while(1)
    {
        printf("----------MENU----------\n");
        printf("1.enqueue\n 2.dequeue\n 3.display\n 4.exit\n");
        scanf("%d",&choice);

        switch(choice)
        {
            case 1:enqueue();
                   break;
            case 2:dequeue();
                   break;
            case 3:display();
                   break;
```

```c
int main()
{
    int choice;

    while(1)
    {
        printf("----------MENU----------\n");
        printf("1.enqueue\n 2.dequeue\n 3.display\n 4.exit\n");
        scanf("%d",&choice);

        switch(choice)
        {
            case 1:enqueue();
                   break;
            case 2:dequeue();
                   break;
            case 3:display();
                   break;
            case 4: exit(0);
                   break;
            default:printf("wrong input");
        }
    }
}
```

```
----------MENU----------
1.enqueue
 2.dequeue
 3.display
 4.exit
1
enter a element:65
----------MENU----------
1.enqueue
 2.dequeue
 3.display
 4.exit
1
enter a element:98
----------MENU----------
1.enqueue
 2.dequeue
 3.display
 4.exit
1
enter a element:68
----------MENU----------
1.enqueue
 2.dequeue
 3.display
 4.exit
3
65
98
68
----------MENU----------
1.enqueue
 2.dequeue
 3.display
 4.exit
2
----------MENU----------
1.enqueue
 2.dequeue
 3.display
 4.exit
3
98
68
----------MENU----------
1.enqueue
 2.dequeue
 3.display
 4.exit
```

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

void enqueue()
{
    int a;
    if (front==rear+1||front==0 && rear==size-1)
    {
        printf("overflow\n");
    }
    else
    {
        if(front==-1)
            front=0;
        printf("enter a element:");
        scanf("%d",&a);
        rear=(rear+1)%size;
        queue[rear]=a;
    }
}

int dequeue()
{
    if(front==-1)
    {
        printf("underflow\n");
    }
    else
    {

        int s = queue[front];
        if (front==rear)
        {
            front=-1;
            rear=-1;
```

```
int dequeue()
{
    if(front==-1)
    {
        printf("underflow\n");
    }
    else
    {

        int s = queue[front];
        if (front==rear)
        {
            front=-1;
            rear=-1;
        }
        else{
            front=(front+1)%size;
        }
        printf("deleted element:%d",s);
        return s;
    }
}
void display()
{
    int i;
    if(front==-1)
    {
        printf("overflow\n");
    }
    else
    {
        for (i=front;i!=rear;i=(i+1)%size)
        {
            printf("%d\n",queue[i]);
        }
        printf("%d\n",queue[i]);
    }
```

```c
int main()
{
    int choice;

    while(1)
    {
        printf("----------MENU---------\n");
        printf("1.enqueue\n 2.dequeue\n 3.display\n 4.exit\n");
        scanf("%d",&choice);

        switch(choice)
        {
            case 1:enqueue();
                   break;
            case 2:dequeue();
                   break;
            case 3:display();
                   break;
            case 4: exit(0);
                   break;
            default:printf("wrong input");
        }
    }
}
```

```
----------MENU----------
1.enqueue
 2.dequeue
 3.display
 4.exit
1
enter a element:32
----------MENU----------
1.enqueue
 2.dequeue
 3.display
 4.exit
1
enter a element:65
----------MENU----------
1.enqueue
 2.dequeue
 3.display
 4.exit
3
32
65
----------MENU----------
1.enqueue
 2.dequeue
 3.display
 4.exit
1
enter a element:98
----------MENU----------
1.enqueue
 2.dequeue
 3.display
 4.exit
1
enter a element:74
----------MENU----------
1.enqueue
 2.dequeue
 3.display
 4.exit
```

```
 3.display
 4.exit
1
enter a element:54
----------MENU---------
1.enqueue
 2.dequeue
 3.display
 4.exit
1
overflow
----------MENU---------
1.enqueue
 2.dequeue
 3.display
 4.exit
88
wrong input----------MENU---------
1.enqueue
 2.dequeue
 3.display
 4.exit
1
overflow
----------MENU---------
1.enqueue
 2.dequeue
 3.display
 4.exit
2
deleted element:32----------MENU---------
1.enqueue
 2.dequeue
 3.display
 4.exit
1
enter a element:88
----------MENU---------
1.enqueue
 2.dequeue
 3.display
```

```
enter a element:88
-----------MENU-----------
1.enqueue
 2.dequeue
 3.display
 4.exit
3
65
98
74
54
88
-----------MENU-----------
1.enqueue
 2.dequeue
 3.display
 4.exit
```