08/01/2024

7] Queue

```
#include <stdio.h>
int max=10;
int q[max], front=-1, rear=-1;
Void insert (int);
int delete ();
Void display ();
Void insert (int num)
{
if (rear == Max-1)
    Printf("\n overflow");
else if (front ==-1 && rear ==-1)
{
front=0;
rear=0;
}
else
rear++
q[rear]=num;
}
int delete
{
int val;
if (front ==-1 || front > rear)
{
Printf ("\n underflow");
return:-1;
}
else{
val = q[front++];
if (front >rear){
rear =-1; front=-1;}
return val; }
}
```

```
void display()
{
 int i;
 Printf("\n");
 if (front == -1 || front > rear)
      Printf("\n Queue is empty");
 else{
 for (i= front ; i<= rear; i++)
   Printf("\t %d", q[i]);
}
}

Void main()
{
Printf("\n 1. Insert \n 2. Delete \n 3. Display \n 4. Exit");
Printf("\n Enter your choice");
int ch; int val;
 Scanf("%d", &ch);
 while (ch!=)
{
case 1: Printf("\n Enter Value to be inserted:");
        Scanf("%d", &val);
        insert (val);
Case 2 : Printf("\n the queue is:");
        displa.
Case 2: val = delete ();
        if (val != -1);
        Printf("\n Deleted number is %d", val);
        break;
Case 3 : Printf("\n The queue is:");
        display ();
        break;
```

```
default : Printf("\n Wrong Option");
}
}
```

Out Put:

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

1. Enter your choice : 1

Enter value to be inserted : 10

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

Enter Choice : 2

Enter value to be inserted : 12

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

Enter your choice 3

10

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

Enter your choice : 2

Deleted number is 10

3] Circular Queue

```
#include <stdio.h>
#include <Process.h>
#define QUE_SIZE 3
int item, front=0, rear=-1, q[QUE_SIZE], 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 q()
{
int i,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]);
i=(i+1)%QUE_SIZE;
}
}
void main()
{ int choice;
  for(;;)
{
Printf("\n 1: insert rear \n 2: delete front \n
          3: display \n 4. 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);
         ~~Insert~~
         insert rear();
         break;
Case 2 : item = delete front();
         if(item == -1)
         Printf("queue is empty \n");
         else
         Printf("item deleted = %d\n",item);
         break;
Case 3 : display Q();
         break;
   default: exit();
  }
  }
  getch();
}
```

Output

1. insert 2. delete 3. display 4. exit

1

enter the item to be inserted 23

1. insert 2. delete 3 display 4. exit

1

enter the item to be inserted 34

1. insert 2. delete 3. display 4. exit

1

enter the item to be inserted 45

1. insert 2. delete 3. display 4. exit

3

Content of queue 23 34 45

8/01/24