11/1/24 weak-4

3b) circular queue

```
#include<stdio.h>
#include<stdlib.h>
# define size 50.
int q[size];
int rear=-1;
int front=-1;
int ISFull()
 {
   if(front==(rear+1)%size)
   {
     return0;
   }
   else
   {
     return-1;
   }
 }
int IsEmpty()
 {
   if(front==-1 && rear==-1)
   {
     return0;
   }
   else
   {
     return-1;
   }
 }
```

```
void Enqueue(int x)
{
   int item;
   if(IsFull()==0)
   {
     printf("queue Overflow \n");
     return;
   }
   else
  {
   if (IsEmpty()==0)
   {
     front=0;
     rear=0;
   }
   else
   {
    rear= (rear+1)%size;
   }
   Q[rear]=x;
  }
}
int Dequeue()
{
   int x;
   if(IsEmpty()==0)
   {
     printf("queue underflow \n");
   }
   else
   {
     if(front==rear)
    {
     x=Q[front];
     front=-1;
     rear=-1;
    }
    else
    {
     x=Q[front];
     front=(front+1)%size;
    }
    return x;
    }
  }
  void Display()
  {
    int i;
    if(IsEmpty()==0)
    {
      printf("Queue is empty \n");
    }
    else
    {
      printf("Queue elements: \n");
      for(i=front; i!=rear; i=(i+1)% size)
      {
         printf("%d \n", Q[i]);
      }
      printf("%d \n", Q[i]);
    }
  }
```

```
Void main()
{
 int choice,x,b;
 while(1)
 {
  printf("1.Enqueue, 2.Dequeue ,3.Display.
          4.Exit \n");
  printf("Enter your choice:\n");
  scanf("%d",&choice);
  switch(choice)
  {
    case 1:
      printf("Enter the number to be Inserted
              into the queue \n");
      scanf("%d",&x);
      Enqueue(x);
      break;
    case 2:
      b=Dequeue();
      printf("%d was removed from the queue\n",
                        b);

      break;
    case 3:
       Display();
       break;
    case 4:
       Exit(1);
    default:
       printf("Invalid input \n");
  }
 }
}
```

11/1/24

Output:

```
1.Enqueue, 2.Dequeue, 3.Display, 4.Exit
Enter your choice: 1
Enter the number to be inserted into the
 queue: 23.
1.Enqueue, 2, Dequeue, 3, Display, 4.Exit
Enter your choice: 1
Enter the number to be inserted into  the
 queue: 24.
1.Enqueue, 2.Dequeue, 3.Display, 4.Exit
Enter your choice: 3.
Queue element: 23, 24.
1.Enqueue, 2.Dequeue, 3.Display, 4.Exit
Enter your choice: 4.
```

singly linked list.

```
#include<stdio.h>
#include<stdlib.h>
Struct node
 {
   int data;
   Struct node * next;
 };
void display();
void insert_begin();
void insert_end();
void insert_pos();
struct node *head=NULL;
void display()
{
  printf("Elements are :\n");
  Struct node *ptr;
  if(head==NULL)
  {
     printf("List is empty \n");
     return;
  }
else
{
  ptr=head;
  while(ptr!=NULL)
  {
     printf("%d \n", ptr->data);
     ptr=ptr->next;
  }
}
```

```
}
void insert_begin()
{
  struct node *temp;
  temp=(struct node *) malloc(sizeof(struct node));
  printf("Enter the value to be inserted \n");
  scanf("%d", &temp->data);
  temp->next =NULL;
  if (head = =NULL)
     head= temp;
    else
  {
    temp->next =head;
     head =temp;
  }
}
void insert_end()
 {
  struct node *temp, *ptr;
  temp=(struct node*)malloc(sizeof(struct node));
  printf("Enter the value to be inserted \n");
  scanf("%d", &temp->data);
  temp->next = NULL;
  if (head =NULL)
  {
     head=temp;
  }
 else
  {
    ptr=head;
    while (ptr->next !=NULL)
    {
        ptr=ptr->next;
      }
      ptr->next=temp;
    }
  }
void insert_pos()
{
   int pos,i;
   struct node *temp, *ptr;
      printf("enter the position");
   scanf("%d", &pos);
   temp=(struct node*)malloc(sizeof(struct node));
   printf("enter the value to be inserted \n");
   scanf("%d", &temp->data);
   temp->next=NULL;
   if(pos==0)
   {
      temp->next=head;
      head=temp;
   }
  else
  {
     for(i=0, ptr=head; i<pos-1; i++)
     {
       ptr=ptr->next;
     }
     temp->next =ptr->next;
     ptr->next=temp;
     }
  }
  void main()
  {
```

```
int choice;
while(1)
{
 printf("\n1. to insert at the beginning \n
          2. to insert at the end \n
          3. to insert at the position \n
         4. to display \n 5. exit \n");
 printf("enter your choice: \n");
 scanf("%d", &choice);
 switch(choice)
 {
 case1:
    insert-begin();
    break;
 case2:
    insert_end();
    break;
 case3:
    insert_ pos();
    break;
 case 4:
    display();
    break;
 case 5;
    exit(0);
    break;
 ~~deafu~~
 default:
    printf("Invalid ~~Enter the~~ choice choice \n");
      break;
 }
}}
```

output:

```
1. to insert at the beginning
2. to insert at the end
3. to insert at any position enter
4. to display
5. to exit
Enter your choice: 1
Enter the value to be inserted 23

1. to insert at the beginning
2. to insert at the end
3. to insert at any position enter
4. to display
5. to exit
Enter your choice: 2
Enter the value to be inserted 13

1. to insert at the beginning
2. to insert at the end
3. to insert at any position enter
4. to display
5. to exit
Enter to your choice: 3
Enter the position: 1
Enter the valued to be inserted
25

1. to insert at the beginning
2. to insert at the end
3. to insert at any position enter
4. to display
```

```
5. to exit.
Enter your choice:-4
 23,25,13
```