Queues

Standared queue implementation:

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#define MAX 5
int queue[MAX];
int front=-1,rear=-1;
void insert();
void traverse();
int delete();
void main()
{
        int op,val;
        while(1)
        {
                printf("\n**menu**\n");
                printf("1.insert\n");
                printf("2.delete\n");
                printf("3.display\n");
                printf("4.exit\n");
                printf("enter option\n");
                scanf("%d",&op);
                switch(op)
                {
                        case 1:
                                        insert();
                                        break;
                        case 2:
                                        val=delete();
                                        if(val!=-1)
                                                printf("%d deleted\n",val);
                                        break;
                        case 3:
                                        traverse();
                                        break;
                        case 4:
                                        exit(0);
                                        break;
                }
```

```
	}

}
void insert()
{
	int val;
	printf("enter element\n");
	scanf("%d",&val);
	if(rear==MAX-1)
	{
		printf("queue is full\n");
	}
	else if(front==-1&&rear==-1)
	{
		front=rear=0;
		queue[rear]=val;
	}
	else
	{
		rear++;
		queue[rear]=val;

	}

}
void traverse()
{
	int i;
	if(front==-1||front>rear)
	{
		printf("queue is empty\n");
	}
	else
	{
		for(i=front;i<=rear;i++)
		{
			printf("%d\t",queue[i]);
		}
	}
}
int delete()
{
	int val;
	if(front==-1||front>rear)
	{
		printf("queue is empty\n");
```

```
                return -1;
        }
        else
        {
                val=queue[front];
                front++;
                if(front>rear)
                {
                        front=rear=-1;
                }
        }
        return val;
}
```

Programme to implement linked queue

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *link;
};
struct node *front=NULL;
struct node *rear=NULL;
void insert();
int display(struct node*);
void delete();
void main()
{
    int op;
    while(1)
    {
                printf("\n***menu***\n");
                printf("1.insert\n");
                printf("2.display\n");
                printf("3.delete\n");
                printf("4.exit\n");
                printf("enter option\n");
                scanf("%d",&op);
                switch(op)
                {
                        case 1:
                                        insert();
```

```
                                        break;
                        case 2:
                                        display(front);
                                        break;
                        case 3:
                                        delete();
                                        break;
                        case 4:
                                        exit(0);
                                        break;

                }

        }
}
void insert()
{
        struct node *temp;
        temp=(struct node*)malloc(sizeof(struct node));
        printf("enter node data\n");
        scanf("%d",&temp->data);
        temp->link=NULL;
        if(front==NULL&&rear==NULL)
        {
                front=temp;
                rear=temp;
        }
        rear->link=temp;
        rear=temp;
}
int display(struct node* temp)
{
        printf("queue elements\n");
        while(temp!=NULL)
        {
                printf("%d\t",temp->data);
                temp=temp->link;
        }
}
void delete()
{
        if(front==NULL&&rear==NULL)
        {
                printf("queue underflow\n");
        }
        else
        {
```

```
            printf("popped element:%d\n",front->data);
            front=front->link;
        }
    }
```

Programme to implement circular queue.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#define SIZE 5
int queue[SIZE];
int front=-1;
int rear=-1;
void insert(int);
int delete();
void traverse();
void main()
{
    int op,ele,val;
    while(1)
    {
        printf("***menu***\n");
        printf("1.insert\n");
        printf("2.delete\n");
        printf("3.traverse\n");
        printf("4.exit\n");
        printf("enter option\n");
        scanf("%d",&op);
        switch(op)
        {
            case 1:
                printf("enter value\n");
                scanf("%d",&ele);
                insert(ele);
                break;
            case 2:
                val=delete();
                printf("deleted value:%d\n",val);
                break;
            case 3:
                traverse();
                break;
            case 4:
                exit(0);
            default:
```

```
                                        printf("invalud option\n");
                }
        }

}
void insert(int ele)
{
        if(front==rear+1||(rear==SIZE-1&&front==0))
        {
                printf("queue is full\n");
        }
        else if(front==-1&& rear==-1)
        {
                front=0;
                rear=0;
                queue[rear]=ele;
        }
        else if(rear==SIZE-1&&front!=0)
        {
                rear=0;
                queue[rear]=ele;

        }
        else
        {
                rear++;
                queue[rear]=ele;
        }
}
int delete()
{
        int val;
        if(front==-1&&rear==-1)
        {
                printf("queue is empty\n");
        }
        else if(front==rear)
        {
                val=queue[front];
                front=rear=-1;
        }
        else if(front==SIZE-1)
        {
                val=queue[front];
                front=0;
        }
        else
```

```
        {
                val=queue[front];
                front++;
        }
        return val;
    }
    void traverse()
    {
        int i;
        if(front<rear)
        {
                for(i=front;i<=rear;i++)
                {
                        printf("%d\t",queue[i]);
                }
        }
        else
        {
                for(i=front;i<SIZE;i++)
                {
                        printf("%d\t",queue[i]);
                }
                for(i=0;i<=rear;i++)
                {
                        printf("%d\t",queue[i]);
                }
        }
    }
```

Doubled ended queue implementation.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#define SIZE 5
int queue[SIZE];
int front=-1,rear=-1;
void insert(int);
void deletion();
void display();
int main()
{
    int op,ele;
    while(1)
    {
        printf("\n***menu***\n");
        printf("1.insert\n");
```

```
            printf("2.delete\n");
            printf("3.traverse\n");
            printf("4.exit\n");
            printf("enter option\n");
            scanf("%d",&op);
            switch(op)
            {
                  case 1:
                              printf("enter element\n");
                              scanf("%d",&ele);
                              insert(ele);
                              break;
                  case 2:
                              deletion();
                              break;
                  case 3:
                              display();
                              break;
                  case 4:
                              exit(0);

            }
    }
}
void insert(int ele)
{
    if(front==0&&rear==SIZE-1)
    {
            printf("queue is full\n");
    }
    else if(front==-1&&rear==-1)
    {
            front=0;
            rear=0;
            queue[rear]=ele;
    }
    else if(front==0)
    {
            rear++;
            queue[rear]=ele;
    }
    else if(rear==SIZE-1)
    {
            front--;
            queue[front]=ele;
    }
    else
```

```
    {
            int ch;
            printf("1.leftside insert\n");
            printf("2.rightside insertion\n");
            printf("enter choice\n");
            scanf("%d",&ch);
            if(ch==1)
            {
                    front--;
                    queue[front]=ele;
            }
            else if(ch==2)
            {
                    rear++;
                    queue[rear]=ele;
            }
            else
            {
                    printf("invalid option\n");
            }
    }

}
void deletion()
{
    if(front==-1&&rear==-1)
    {
            printf("queue is empty\n");
    }
    else if(front==rear)
    {
            printf("deleted element:%d\n",queue[rear]);
            front=-1;
            rear=-1;
    }
    else
    {
            int ch;
            printf("1.leftside deletion\n");
            printf("2.rightside deletion\n");
            printf("enter choice\n");
            scanf("%d",&ch);
            if(ch==1)
            {
                    printf("deleted elemnt:%d\n",queue[front]);
                    front++;
            }
```

```
            else if(ch==2)
            {
                  printf("deleted element:%d\n",queue[rear]);
                  rear--;
            }
    }
}
void display()
{
    int i;
    for(i=front;i<=rear;i++)
    {
            printf("%d\t",queue[i]);
    }
}
```