

1. Write a 'C' program to perform following Operations on Array: Create, Insert, Delete, Display.

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int i,n,p,arr[20],temp,ins,num;
    // clrscr();
    // creating an Array
    printf("Enter length of an Array :- ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter Number %d for array :- ",i+1);
        scanf("%d",&arr[i]);
    }
    printf("For Insertion Enter 1 \n For Deletion Enter 2 \n For Display Enter 3 :- ");
    scanf("%d",&num);
    switch(num)
    {
        case 1:
            // inserting operation
            printf("Enter the position where to insert :- ");
            scanf("%d",&p);
            printf("Enter the Element to be Insert :- ");
            scanf("%d",&ins);
            for(i=n-1;i>=p-1;i--)
            {
                temp=arr[i] ;
                arr[i]=arr[i+1];
                arr[i+1]=temp;
            }
            arr[p-1]=ins;
            for(i=0;i<n+1;i++)
                printf("%d",arr[i]);
            break;
        case 2:
            // Deleting operation
            printf("Enter the position where to delete :- ");
            scanf("%d",&p);
            for(i=p-1;i<n;i++)
                arr[i]=arr[i+1];
            for(i=0;i<n-1;i++)
                printf(" %d",arr[i]);
```

```

        break;
    case 3:
        // displaying operation
        for(i=0;i<n;i++)
            printf(" %d",arr[i]);
        break;
    default:
        printf("Enter Valid Number.....");
}
// getch();
return 0;
}

```

- 2. Write a 'C' Program to Search a particular data from the given Array of numbers using: Linear Search Method.**

```

#include<stdio.h>
#include<conio.h>
int main()
{
    int arr[100],n,i;
    int search;
    clrscr();

    printf("Enter the length of array : ");
    scanf("%d",&n);

    printf("\nEnter the elements of array : \n");
    for(i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    printf("\nEnter the element you want to search : ");
    scanf("%d",&search);

    for(i=0;i<n;i++)
    {
        if(arr[i]==search)
        {
            printf("Element found at Index %d and Position %d",i,i+1);
            break;
        }
    }
}

```

```

    }
}
if(i==n)
{
    printf("Element not found");
}
getch();
return 0;
}

```

- 3. Write a 'C' Program to Search a particular data from the given Array of Strings using Linear Search Method.**

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<ctype.h>
int main()
{
    char str[100];
    int n,i;
    char search;
    clrscr();

    printf("Enter a string : ");
    gets(str);

    printf("\n\nEnter the character you want to search : ");
    scanf("%c",&search);

    n=strlen(str);

    for(i=0;i<n;i++)
    {
        if(tolower(str[i])==tolower(search))
        {
            printf("Character found at Index %d and Position %d",i,i+1);
            Break;
        }
    }
    if(i==n)

```

```

{
    printf("Character not found");
}
getch();
return 0;
}

```

- 4. Write a 'C' program to Search a particular data from the given Array of numbers using Binary Search Method.**

```

#include<stdio.h>

//binary search of numbers
int main()
{
    int a[20], start, end, mid, search, n;
    printf("Enter the length of array: ");
    scanf("%d",&n);
    for (int i = 0; i < n; i++)
    {
        printf("Enter element %d: ", i+1);
        scanf("%d",&a[i]);
    }
    printf("Enter the search element: ");
    scanf("%d",&search);
    start=0;
    end=n-1;
    mid= (start+end)/2;
    while (search != a[mid] && start < end)
    {
        if (search > a[mid])
        {
            start = mid + 1;
        }
        else
        {
            end = mid - 1;
        }
        mid = (start + end) / 2;
    }
    if (search == a[mid])
    {
        printf("Element %d found at position %d",search,mid+1);
    }
}

```

```

    }
    else
    {
        printf("Element not found");
    }
    return 0;
}

```

5. Write a 'C' Program to Search a particular data from the given Array of Strings using Binary Search Method.

```

#include<stdio.h>
#include<string.h>
#include<ctype.h>
//binary search of string
int main()
{
    int start, end, mid, n;
    char a[20], search;
    printf("Enter the string: ");
    gets(a);
    n = strlen(a);
    printf("Enter the search element: ");
    scanf("%c",&search);
    start=0;
    end=n-1;
    mid= (start+end)/2;
    while (tolower(search) != tolower(a[mid]) && start < end)
    {
        if (tolower(search) > tolower(a[mid]))
        {
            start = mid + 1;
        }
        else
        {
            end = mid - 1;
        }
        mid = (start + end) / 2;
    }
    if (tolower(search) == tolower(a[mid]))
    {
        printf("Element %c found at position %d",search,mid+1);
    }
    else
    {

```

```

        printf("Element not found");
    }
    return 0;
}

```

6. Write a 'C' Program to Sort an Array of numbers using Bubble Sort Method.

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{

    int arr[100],n,i,temp,k,j;
    clrscr();

    printf("Enter the length of array : ");
    scanf("%d",&n);

    printf("Enter the elements of array :\n");
    for(i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    for(i=0;i<n-1;i++)
    {
        for(j=0;j<n;j++)
        {
            if(arr[j]>arr[j+1])
            {
                temp=arr[j];
                arr[j]=arr[j+1];
                arr[j+1]=temp;
            }
        }
        printf("\nPass %d : ",i+1);
        for(k=0;k<n;k++)
        {
            printf("\t%d",arr[k]);
        }
    }
}

```

```

printf("\n\nFinal Sorted Array : ");
for(i=0;i<n;i++)
{
    printf("\t%d",arr[i]);
}
getch();
return 0;
}

```

7. Write a ‘C’ Program to Sort an Array of Strings using Bubble Sort Method.

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<ctype.h>
int main()
{
    char str[100],temp;
    int n,i,j;
    clrscr();

    printf("Enter a string : ");
    gets(str);

    n=strlen(str);

    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            if(tolower(str[j])>tolower(str[j+1]))
            {
                temp=str[j];
                str[j]=str[j+1];
                str[j+1]=temp;
            }
        }
    }
    printf("\nPass %d : ",i+1);
    printf("\t%s",str);
}

```

```

    }
    printf("\n\nFinal Sorted String : ");
    printf("\t%s",str);

    getch();
    return 0;
}

```

8. Write a 'C' Program to Sort an Array of numbers using the Selection Sort Method.

```

#include<stdio.h>
#include<string.h>
#include<ctype.h>
int main()
{
    int n,a[20],temp,i,j;
    // clrscr();
    printf("\n ENTER THE LENGTH OF AN ARRAY :");
    scanf("%d",&n);
    printf("\n ENTER THE ELEMENT OF AN ARRAY :");
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    for(i=0;i<n-1;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(a[i]>a[j])
            {
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }
        }
        printf("\n ARRAY AFTER PASS %d :",i+1);
        for(j=0;j<n;j++)
            printf(" %d",a[j]);
        printf("\n");
    }
    // getch();
    return 0;
}

```

9. Write a 'C' Program to Sort an Array of Strings using Selection Sort Method.

```

#include<stdio.h>

```



```

#include<string.h>
#include<ctype.h>
int main()
{
    int n,i,j;
    char a[20],temp;
    printf("\n ENTER THE STRING :");
    gets(a);
    n = strlen(a);
    for(i=0;i<n-1;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(tolower(a[i]) > tolower(a[j]))
            {
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }
        }
        printf("\n ARRAY AFTER PASS %d :",i+1);
        for(j=0;j<n;j++)
            printf(" %c",a[j]);
        printf("\n");
    }
    return 0;
}

```

10. Write a ‘C’ Program to Sort an Array of numbers using Insertion Sort Method.

```

#include<stdio.h>
int main()
{
    int a[20], i, j, temp, n;
    printf("Enter the length of array: ");
    scanf("%d", &n);
    for ( i = 0; i < n; i++)
    {
        printf("Enter element %d : ", i+1);
        scanf("%d", &a[i]);
    }
    for ( i = 1; i < n; i++)
    {
        temp = a[i];
        j = i - 1;

```

```

while (a[j] > temp && j>=0)
{
    a[j+1] = a[j];
    j--;
}
a[j+1] = temp;
printf("\n Array after pass %d: ",i);
for ( j = 0; j < n; j++)
{
    printf("\t%d",a[j]);
}
}
return 0;
}

```

11. Write a 'C' Program to Implement Singly Linked List with Operations: (i) Insert at beginning, (ii) Search, (iii) Display

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
};
struct node *head=NULL;

void insert_begin()
{
    int el;
    struct node *new_node;

    new_node = (struct node*)malloc(sizeof(struct node));

    if(new_node==NULL)
    {
        printf("Memory Overflow!");
    }
    else
    {

```

```
printf("Enter the element : ");
scanf("%d",&el);
```

```
new_node -> data = el;
new_node -> next = head;
head = new_node;
```

```
printf("Element Inserted");
}
}
```

```
void search()
```

```
{
int s,i;
struct node *temp;
if(head==NULL)
{
printf("List is Empty");
}
else
{
printf("Enter the Element you want to search : ");
scanf("%d",&s);
```

```
temp=head;
i=0;
while(temp!=NULL)
{
if(temp->data==s)
{
printf("Element Found at Index %d and Position %d",i,i+1);
break;
}
temp=temp->next;
i=i+1;
}
if(temp==NULL)
{
printf("Element Not Found");
}
}
```

```

}
void display()
{
    struct node *temp;
    if(head==NULL)
    {
        printf("List is Empty");
    }
    else
    {
        temp = head;
        printf("Printing Elements : ");
        while(temp!=NULL)
        {
            printf("\t%d",temp->data);
            temp=temp->next;
        }
    }
}
int main()
{
    int ch;
    clrscr();

    while(1)
    {
        printf("\n****Menu****\n");
        printf("1.Insert at Beginning\n2.Search\n3.Display\n4.Exit");
        printf("\nEnter Your Choice Here : ");
        scanf("%d",&ch);

        switch(ch)
        {
            case 1:
                insert_begin();
                break;

            case 2:
                search();
                break;

```

```

case 3:
    display();
    break;

case 4:
    exit(1);

default:
    printf("Invalid Choice");
}
}
getch();
return 0;
}

```

12. Write a C Program to Implement Singly Linked List with Operations: (i) Insert at end, (ii) Insert After, (iii) Delete (iv) Display

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
};
struct node *head=NULL;

void insert_end()
{
    int el;
    struct node *new_node,*temp;

    new_node = (struct node*)malloc(sizeof(struct node));

    if(new_node==NULL)
    {
        printf("Memory Overflow!");
    }
}

```

```

    }
    else
    {
        printf("Enter the element : ");
        scanf("%d",&el);

        new_node -> data = el;
        if(head==NULL)
        {
            new_node -> next = head;
            head = new_node;
        }
        else
        {
            temp=head;
            while(temp->next!=NULL)
            {
                temp=temp->next;
            }
            temp->next=new_node;
            new_node->next=NULL;
        }
        printf("Element Inserted");
    }
}

void insert_after()
{
    int el,loc,i;
    struct node *new_node,*temp;

    new_node = (struct node *)malloc(sizeof(struct node));

    if(new_node==NULL)
    {
        printf("Memory Overflow");
    }
    else
    {
        printf("Enter the element : ");

```

```
scanf("%d",&el);
```

```
new_node->data=el;
```

```
if(head==NULL)
```

```
{
```

```
    new_node->next=head;
```

```
    head=new_node;
```

```
}
```

```
else
```

```
{
```

```
    printf("Enter the location after which you want to insert the element : ");
```

```
    scanf("%d",&loc);
```

```
    temp=head;
```

```
    for(i=0;i<loc-1;i++)
```

```
    {
```

```
        temp=temp->next;
```

```
    }
```

```
    new_node->next=temp->next;
```

```
    temp->next=new_node;
```

```
}
```

```
    printf("Element Inserted");
```

```
}
```

```
}
```

```
void delete_begin()
```

```
{
```

```
    struct node *temp;
```

```
    if(head==NULL)
```

```
    {
```

```
        printf("List Underflow");
```

```
    }
```

```
    else
```

```
    {
```

```
        temp=head;
```

```
        head=temp->next;
```

```

    printf("Deleted Element is %d",temp->data);
    free(temp);
}
}

void delete_last()
{
    struct node *temp,*temp1;

    if(head==NULL)
    {
        printf("List Underflow");
    }
    else if(head->next==NULL)
    {
        temp=head;
        head=NULL;
        printf("The only Element in the list %d is deleted",temp->data);
        free(temp);
    }
    else
    {
        temp=head;
        while(temp->next!=NULL)
        {
            temp1=temp;
            temp=temp->next;
        }
        temp1->next=NULL;
        printf("Deleted Element is %d",temp->data);
        free(temp);
    }
}

void delete_random()
{
    int loc,i;
    struct node *temp,*temp1;

    if(head==NULL)

```



```

{
    printf("List Underflow");
}
else if(head->next==NULL)
{
    temp=head;
    head=NULL;
    printf("The only Element in the list %d is deleted",temp->data);
    free(temp);
}
else
{
    printf("Enter the location of the node you want to delete : ");
    scanf("%d",&loc);
    temp=head;
    for(i=0;i<loc;i++)
    {
        temp1=temp;
        temp=temp->next;
    }
    temp1->next=temp->next;
    printf("Element deleted is %d",temp->data);
    free(temp);
}
}

```

```

void display()
{
    struct node *temp;
    if(head==NULL)
    {
        printf("List is Empty");
    }
    else
    {
        temp = head;
        printf("Printing Elements : ");
        while(temp!=NULL)
        {
            printf("\t%d",temp->data);

```

```

        temp=temp->next;
    }
}

int main()
{
    int ch;
    clrscr();

    while(1)
    {
        printf("\n****Menu****\n");
        printf("1. Insert at End\n2. Insert at random\n3. Delete Beginning\n4. Delete Last\n5. Delete Random\n6. Display\n7. Exit");
        printf("\nEnter Your Choice Here : ");
        scanf("%d",&ch);

        switch(ch)
        {
            case 1:
                insert_end();
                break;

            case 2:
                insert_after();
                break;

            case 3:
                delete_begin();
                break;

            case 4:
                delete_last();
                break;

            case 5:
                delete_random();
                break;

```

```

case 6:
    display();
    break;

case 7:
    exit(1);

default:
    printf("Invalid Choice");
}
}
getch();
return 0;
}

```

13. Write a 'C' Program to perform PUSH and POP Operations on Stack using an Array.

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
#include<ctype.h>
#define max 5
int arr[max];
int top=-1;

void push()
{
    int el;
    if(top==max-1)
    {
        printf("Stack Overflow");
    }
    else
    {
        printf("Enter the Element : ");
        scanf("%d",&el);
        if(top==max-1)
        {

```

```
    top=0;
}
else
{
    top++;
}
arr[top]=el;
printf("Element Pushed in Stack");
}
}
```

```
void pop()
{
    int val;
    if(top==-1)
    {
        printf("Stack Underflow");
    }
    else
    {
        val=top;
        printf("Element %d Poped from the Stack",arr[val]);
        top--;
    }
}
```

```
void display()
{
    int i;
    if(top==-1)
    {
        printf("Stack is Empty");
    }
    else
    {
        printf("Printing Stack Elements : ");
        for(i=0;i<=top;i++)
        {
            printf("\t%d",arr[i]);
        }
    }
}
```

```
}  
}
```

```
int main()
```

```
{
```

```
int ch;
```

```
clrscr();
```

```
while(1)
```

```
{
```

```
printf("\n1. Insert\n2. Delete\n3. Display\n4. Exit");
```

```
printf("\nEnter Your Choice Here : ");
```

```
scanf("%d",&ch);
```

```
switch(ch)
```

```
{
```

```
case 1:
```

```
    push();
```

```
    break;
```

```
case 2:
```

```
    pop();
```

```
    break;
```

```
case 3:
```

```
    display();
```

```
    break;
```

```
case 4:
```

```
    exit(1);
```

```
default:
```

```
    printf("Invalid Choice");
```

```
}
```

```
}
```

```
getch();
```

```
return 0;
```

```
}
```

14. Write a 'C' Program to perform PUSH and POP Operations on a Stack using a Linked List.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

struct node
{
    int data;
    struct node *next;
};

struct node *top = NULL;

void push()
{
    int val;
    struct node *new_node;

    new_node = (struct node*)malloc(sizeof(struct node));

    printf("Enter a Number: ");
    scanf("%d",&val);

    new_node -> data = val;

    new_node->next = top;

    top = new_node;
    printf("Element Pushed in Stack");

}

void pop()
{
    struct node *temp;
    if(top == NULL)
```

```
{  
    printf("Stack is Empty");  
}  
else  
{  
    temp = top;  
    printf("Element Poped is %d",temp -> data);  
    top = temp -> next;  
    free(temp);  
}  
}
```

```
void display()  
{  
    struct node *temp;  
  
    if(top==NULL)  
    {  
        printf("Stack is Empty");  
    }  
    else  
    {  
        temp = top;  
        printf("Printing Elements of Stack:\n");  
        while(temp!=NULL)  
        {  
            printf("%d\t",temp -> data);  
            temp = temp -> next;  
        }  
    }  
}
```

```
int main()  
{  
    int ch;  
    clrscr();  
  
    while(1)
```

```

{
printf("\n\n*****Menu*****");
printf("\n1. Push\n2. Pop\n3. Display\n4. Exit\n");
printf("Enter your choice here: ");
scanf("%d",&ch);

switch(ch)
{
case 1:
    push();
    break;

case 2:
    pop();
    break;

case 3:
    display();
    break;

case 4:
    exit(1);
    break;
default:
    printf("Invalid Choice");
}
}
getch();
return 0;
}

```

15. Write a 'C' program to perform multiplication of two numbers using recursion.

```

#include<stdio.h>
#include<conio.h>
int mul(int x,int y)
{
    if(y>1)
    {
        return x+mul(x,y-1);
    }
}

```



```

    }
    int main()
    {
        int num,num1;
        clrscr();
        printf("ENTER TWO NUMBERS FOR MULTIPLICATION :- ");
        scanf("%d %d",&num,&num1);

        printf("%d X %d = %d",num,num1,mul(num,num1));

        getch();
        return 0;
    }

```

16. Write a 'C' program to print given string in reverse using recursion.

```

#include<stdio.h>
#include<string.h>
#include<conio.h>
int rev(char str[100],int n)
{
    printf("%c",str[n]);

    if(n>0)
        return rev(str,n-1);
}
int main()
{
    char str[100];
    int n;
    clrscr();
    printf("ENTER ANY STRING :- ");
    gets(str);
    n=strlen(str);
    printf("\nYOUR STRING : %s", str);
    printf("\n\nREVERSED STRING : ");
    rev(str,n);
}

```

```
        getch();
        return 0;
    }
```

17. Write a 'C' Program to perform INSERT and DELETE Operations on Linear Queue using an Array.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#define max 5
int front=-1,rear=-1;
int arr[max];
void enqueue()
{
    int el;
    if(rear==max-1)
    {
        printf("Queue Overflow");
    }
    else
    {
        printf("Enter the element : ");
        scanf("%d",&el);

        if(front==max-1)
        {
            rear=0;
            front=0;
        }
        else
        {
            rear++;
        }
        arr[rear]=el;
        printf("Element Enqueued!");
    }
}
```

```
}
```

```
void dequeue()
```

```
{
```

```
    int val;
```

```
    if(front==-1||front>rear)
```

```
    {
```

```
        printf("Queue Underflow");
```

```
    }
```

```
    else
```

```
    {
```

```
        val=front;
```

```
        printf("Deleted Element is %d",arr[val]);
```

```
        front++;
```

```
    }
```

```
}
```

```
void display()
```

```
{
```

```
    int i;
```

```
    if(front==-1||front>rear)
```

```
    {
```

```
        printf("Queue is Emmpy");
```

```
    }
```

```
    else
```

```
    {
```

```
        for(i=front;i<=rear;i++)
```

```
        {
```

```
            printf("\t%d",arr[i]);
```

```
        }
```

```
    }
```

```
}
```

```
int main()
```

```
{
```

```
    int ch;
```

```
    clrscr();
```

```
    while(1)
```

```
    {
```

```

printf("\n*****Menu*****\n");
printf("1. Insert\n2. Delete\n3. Display\n4. Exit");
printf("\nEnter Your Choice Here : ");
scanf("%d",&ch);

switch(ch)
{
    case 1:
        enqueue();
        break;

    case 2:
        dequeue();
        break;

    case 3:
        display();
        break;

    case 4:
        exit(1);

    default:
        printf("Invalid Choice");
}
}
getch();
return 0;
}

```

18. Write a 'C' Program to perform INSERT and DELETE operations on Linear Queue using a Linked List.

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
#include<ctype.h>
struct node

```

```

{
    int data;
    struct node *next;
};

struct node *front=NULL;
struct node *rear=NULL;

void enqueue()
{
    int el;
    struct node *ptr;

    ptr=(struct node*)malloc(sizeof(struct node));

    if(ptr==NULL)
    {
        printf("Queue Overflow");
    }
    else
    {
        printf("Enter the Element : ");
        scanf("%d",&el);

        ptr->data=el;
        ptr->next=NULL;

        if(front==NULL&&rear==NULL)
        {
            front=rear=ptr;
        }
        else
        {
            rear->next=ptr;
            rear=ptr;
        }
        printf("Element Inserted");
    }
}

```

```

void del()
{
    struct node *temp;

    if(front==NULL&&rear==NULL)
    {
        printf("Queue Underflow");
    }
    else
    {
        temp=front;
        printf("Deleted Element is %d",temp->data);
        front=front->next;
        if(front==NULL)
        {
            rear=NULL;
        }
        free(temp);
    }
}

```

```

void display()
{
    struct node *temp;

    if(front==NULL&&rear==NULL)
    {
        printf("Queue is Empty");
    }
    else
    {
        temp=front;
        printf("Printing Queue Elements : ");
        while(temp!=NULL)
        {
            printf("\t%d",temp->data);
            temp=temp->next;
        }
    }
}

```

```

int main()
{
    int ch;
    clrscr();

    while(1)
    {
        printf("\n1. Insert\n2. Delete\n3. Display\n4. Exit");
        printf("\nEnter Your Choice Here : ");
        scanf("%d",&ch);

        switch(ch)
        {
            case 1:
                enqueue();
                break;

            case 2:
                del();
                break;

            case 3:
                display();
                break;

            case 4:
                exit(1);

            default:
                printf("Invalid Choice");
        }
    }

    getch();
    return 0;
}

```

19. Write a 'C' Program to perform INSERT and DELETE operations on Circular Queue using an Array.

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
#include<ctype.h>
#define max 5
int rear=-1,front=-1;
int arr[max];

void enqueue()
{
    int el;
    if((rear==max-1&&front==0)||((front==rear+1))
    {
        printf("Queue Overflow!");
    }
    else
    {
        printf("Enter the Element : ");
        scanf("%d",&el);

        if(front==-1&&rear==-1)
        {
            front=0;
            rear=0;
        }
        else if(rear==max-1&&front!=0)
        {
            rear=0;
        }
        else
        {
            rear++;
        }
        arr[rear]=el;
        printf("Element Inserted!");
    }
}

void del()
{
    int val;
    if(front==-1&&rear==-1)
    {

```



```

    printf("Queue is Empty");
}
else
{
    val=front;
    printf("Element Deleted is %d",arr[val]);

    if(front==rear)
    {
        front=-1;
        rear=-1;
    }
    else if(front==max-1)
    {
        front=0;
    }
    else
    {
        front++;
    }
}
}

```

```

void display()
{
    int i;
    if(front== -1 && rear== -1)
    {
        printf("Queue is Empty");
    }
    else
    {
        printf("Printing Queue Elements : ");
        if(front<=rear)
        {
            for(i=front;i<max;i++)
            {
                printf("\t%d",arr[i]);
            }
        }
        else
        {
            for(i=front;i<max;i++)
            {

```

```

        printf("\t%d",arr[i]);
    }
    for(i=0;i<=rear;i++)
    {
        printf("\t%d",arr[i]);
    }
    }
}2
int main()
{
    int ch;
    clrscr();

    while(1)
    {
        printf("\n1. Insert\n2. Delete\n3. Display\n4. Exit");
        printf("\nEnter Your Choice Here : ");
        scanf("%d",&ch);

        switch(ch)
        {
            case 1:
                enqueue();
                break;

            case 2:
                del();
                break;

            case 3:
                display();
                break;

            case 4:
                exit(1);

            default:
                printf("Invalid Choice");
        }
    }
    getch();
    return 0;
}

```