

PROGRAM01-Reverse a String using Pointer

```
#include<stdio.h>
#include<conio.h>
int main()
{
    char str[50];
    char revstr[50];
    char *strptr=str;
    char *revptr=revstr;
    int len=-1;
    printf("enter the string:\n");
    scanf("%s",str);
    while(*strptr)
    {
        strptr++;
        len++;
    }
    while(len>=0)
    {
        strptr--;
        *revptr=*strptr;
        revptr++;
        --len;
    }
    *revptr='\0';
    printf("reverse of a string is \n\n%s",revstr);

    return 0;
}
```

```
+++++
+++++
```

PROGRAM02 -Implement Pattern Matching Algorithm

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
int main()
{
    char txt[20],pat[20];
    int a,b,i,j;

    printf("enter the string:\n");
    gets(txt);
    printf("enter the pattern to find:\n");
    gets(pat);
    a=strlen(pat);
    b=strlen(txt);
    for(i=0;i<=b-a;i++)
    {
        for(j=0;j<a;j++)
            if(txt[i+j]!=pat[j])
                break;
    }
}
```

```

if(j==a)
printf("\n pattern found at index %d\n",i+1);
}
return 0;
}

```

```

+++++
+++++

```

PROGRAMt03 –Search an element in the 2- dimensional Array

```

#include<stdio.h>
int main(){
int m,n,i,j,count=0,item,a[20][20];
printf("enter the rows and columns");
scanf("%d%d",&m&n);
printf("enter %d numbers",m*n);
for(i=0;i<m;i++){
for(j=0;j<n;j++){
scanf("%d",&a[i][j]);
}
}
printf("enter the item to find:");
scanf("%d",&item);
for(i=0;i<m;i++){
for(j=0;j<n;j++){
if (a[i][j]==item){
printf("item found at :[%d,%d]\n",i,j);
count++;
}
}
}

}
printf("\n");
if (count==0)
printf("item not found");
return 0;
}

```

```

+++++
+++++

```

PROGRAMt04 Append two Arrays

```

#include<stdio.h>
#include<string.h>
int main(){
int a[20],b[20],c[40],i,j,k,m,n,a_len,b_len;

printf("Enter limit of elements in first array : ");
scanf("%d",&m);
printf("Enter the elements of 1st array : ");
for(i = 0; i< m; i++){
scanf("%d",&a[i]);
}
}

```

```

printf("Enter limit of elements in sec array : ");
scanf("%d",&n);
printf("Enter the elements of 2nd array: ");
for(j = 0; j < n; j++){
    scanf("%d",&b[j]);
}

for(i = 0; i < m; i++){
    c[i]=a[i];
}

for(j = 0; j < n; j++){
    c[j+m]=b[j];
}
printf("\n");
printf("Arrays after appending : ");
for(i=0;i<m+n;i++){
    printf("%d\t",c[i]);
}
return 0;
}

```

```

+++++
+++++

```

PROGRAM05- Search an element in the array using iterative binary search.

```

#include<stdio.h>
int main()
{
    int list[25],max,first,last,middle,i,item,loc=-1;

    printf("\n enter the limit:");
    scanf("%d",&max);
    printf("\n enter array elements:");
    for(i=0;i<max;i++){
        scanf("%d",&list[i]);
    }
    printf("\n Enter item to be search:");
    scanf("%d",&item);
    first=0;
    last=max-1;
    while(first<=last){
        middle=(first+last)/2;
        if(item==list[middle]){
            loc=middle;
            break;
        }
        if(item<list[middle])
            last=middle-1;
        else
            first=middle+1;
    }
    if(loc!=-1)
        printf("\n the item is found at position %d",loc+1);
}

```

```

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

```

```

+++++
+++++

```

PROGRAM8- Read a sparse matrix and display its triplet representation using array

```

#include<stdio.h>
#include<conio.h>
int main()
{
    int i,j,m,n,ar[10][10],br[10][10],s=0;
    printf("\n enter order of matrix :");
    scanf("%d%d",&m,&n);
    printf("\n elements of matrix:");
    for(i=0;i<m;i++){
        for(j=0;j<n;j++){
            scanf("%d",&ar[i][j]);
        }
    }
    printf("\n the given matrix is:\n");
    for(i=0;i<m;i++){
        for(j=0;j<n;j++){
            printf("%d\t",ar[i][j]);
        }
    }
    printf("\n");
    for(i=0;i<m;i++){
        for(j=0;j<n;j++){
            if(ar[i][j]!=0){
                br[s][0]=i;
                br[s][1]=j;
                br[s][2]=ar[i][j];
                s++;
            }
        }
    }
    printf("\n the sparse matrix is:\n");
    for(i=0;i<s;i++){
        for(j=0;j<3;j++){
            printf("%d\t",br[i][j]);
        }
    }
    printf("\n");
    return 0;
}

```

```

+++++
+++++

```

PROGRAM12-Implement Stack using array

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

int stack[100], choice, top, n, val, i;

void push();
void pop();
void display();

int main()
{
    top = -1;
    printf("enter size of the stack:\t");
    scanf("%d", &n);
    printf("-----stack operation-----");
    printf("\n 1.push 2.pop 3.display 4.exit\n");

    do
    {
        printf("\n enter your choice:");
        scanf("%d", &choice);

        switch(choice)
        {
            case 1:
                push();
                break;
            case 2:
                pop();
                break;
            case 3:
                display();
                break;
            case 4:
                printf("exit point");
                break;
            default:
                printf("invalid choice");
        }
    } while(choice != 4);

    return 0;
}

void push()
{
    if(top < n-1)
    {
        printf("\n enter element to be pushed:");
        scanf("%d", &val);
        top++;
        stack[top] = val;
    }
    else
```

```

    {
        printf("stack overflow");
    }
}

void pop()
{
    if(top > -1)
    {
        printf("the popped element is %d", stack[top]);
        top--;
    }
    else
    {
        printf("stack underflow");
        // Or you can return an appropriate error code indicating underflow.
    }
}

void display()
{
    if(top >= 0)
    {
        printf("the elements of the stack are:\n");
        for(i = top; i >= 0; i--)
        {
            printf("%d\n", stack[i]);
        }
    }
    else
    {
        printf("stack is empty");
    }
}
}

```

```

+++++
+++++

```

PROGRAM:15-Implement Queue using Array

```

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

int array[100], n, front = -1, rear = -1, val, choice, i;

void insert();
void delete_element();
void display();

int main()
{
    printf("Enter size of the queue: ");
    scanf("%d", &n);
}

```

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

```
do
{
    printf("\nEnter your choice: ");
    scanf("%d", &choice);

    switch (choice)
    {
        case 1:
            insert();
            break;
        case 2:
            delete_element();
            break;
        case 3:
            display();
            break;
        case 4:
            printf("Exit point\n");
            break;
        default:
            printf("Invalid choice\n");
    }
} while (choice != 4);
```

```
return 0;
```

```
}
```

```
void insert()
```

```
{
    if (rear==n-1)
    {
        printf("Overflow\n");
    }
    else
    {
        if (front == -1)

            front = 0;
        printf("\nEnter element to be inserted: ");
        scanf("%d", &val);
        rear = rear + 1;
        array[rear] = val;
    }
}
```

```
void delete_element()
```

```
{
    if (front == -1)
    {
        printf("Queue underflow\n");
        return;
    }
}
```

```

        else
        {
            printf("The deleted element is %d\n", array[front]);
            front = front + 1;
        }
    }
}

```

```

void display()
{
    if (front == -1)
    {
        printf("Queue is empty\n");
    }
    else
    {
        printf("The elements of the queue are:\n");
        for (i = front; i <= rear; i++)
        {
            printf("%d\n", array[i]);
        }
    }
}

```

```

+++++
+++++

```

PROGRAM:19- selection_sort

```

#include<stdio.h>
#include<conio.h>
int main()
{
    int ar[25],n,i,j,min,pos;

    printf("Enter the limit:\n");
    scanf("%d",&n);
    printf("Enter elements:\n");
    for(i=0;i<n;i++)
        scanf("%d",&ar[i]);
    for(i=0;i<n-1;i++){
        min=ar[i];
        pos=i;
        for(j=i+1;j<n;j++){
            if(ar[j]<min){
                min=ar[j];
                pos=j;
            }
        }
        if(pos!=i){
            ar[pos]=ar[i];
            ar[i]=min;
        }
    }
    printf("sorted array is \n");
    for(i=0;i<n;i++){
        printf("%d\t",ar[i]);
    }
}

```



```

}
return 0;
}

```

```

+++++
+++++

```

PROGRAM:20-insertion_sort

```

#include<stdio.h>
#include<conio.h>
int main()
{
    int a[100],i,n,j,temp;

    printf("enter no of elements:");
    scanf("%d",&n);
    printf("Enter the elements:");
    for(i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    for(i=1;i<=n-1;i++){
        j=i;
        while(j>0 && a[j-1]>a[j]){
            temp=a[j];
            a[j]=a[j-1];
            a[j-1]=temp;
            j--;
        }
    }

    printf("The sorted elements are:\n");
    for(i=0;i<=n-1;i++){
        printf("%d\n",a[i]);
    }
    return 0;
}

```