

Solve count:**26.5**

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>
<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>	<b>29</b>	<b>30</b>
<b>31</b>	<b>32</b>	<b>33</b>	<b>34</b>	<b>35</b>	<b>36</b>	<b>37</b>	<b>38</b>	<b>39</b>	<b>40</b>

**Hajee Mohammad Danesh Science and Technology University**  
**B.Sc. (Engineering) in ECE**  
**Course Code: CSE 212**  
**Data Structure and Algorithm Sessional**  
**Course Teacher: Professor Dr. Ashis Kumar Mandal**

### **Programming Problems:**

**1)write a program that finds the prime numbers using sieve method.**

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

int main()
{
    int n;
    printf("Enter n value: ");
    scanf("%d",&n);
    int arr[n+1];

    for(int i = 1; i <= n; i++)
    {
        arr[i] = i;
    }
    for(int i = 2; i<=sqrt(n); i++)
    {
        if(arr[i] != -1)
        {
            for(int j=2*i; j<=n ; j=j+i)
                arr[j]=-1;
        }
    }
    printf("Prime numbers are: ");
```

```

        for(int i=2; i<= n; i++)
        {
            if(arr[i]!=-1)
            {
                printf("%d ",i);
            }
        }
        printf("\n");

        return 0;
    }

```

**Sample output:**

```

Enter n value: 100
Prime numbers are: 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
Process returned 0 (0x0)  execution time : 1.913 s
Press any key to continue.

```

**2) Write a program that finds the largest and smallest elements in an array.**

**code:**

```

#include<stdio.h>
int main()
{
    int arr[50],i,n,large,small;
    printf("How many elements: ");
    scanf("%d",&n);
    printf("Enter the Array: ");

    for(i=0; i<n; i++)
        scanf("%d",&arr[i]);

    large=arr[0];
    small=arr[0];
    for(i=1; i<n; i++)
    {
        if(arr[i]>large)
            large=arr[i];
        if(arr[i]<small)
            small=arr[i];
    }
}

```

```
        }
        printf("The largest element is %d\n",large);
        printf("The smallest element is %d\n",small);

        return 0;
    }
```

**Sample output:**

```
How many elements: 5
Enter the Array: 1 2 4 5 0
The largest element is 5
The smallest element is 0

Process returned 0 (0x0) execution time : 7.659 s
Press any key to continue.
```

3) An array A Containing N elements is given. Write a program that captures the sum of array elements.

**Code:**

```
#include<stdio.h>
int main()
{
    int arr[50],i,n,sum=0;
    printf("How many elements: ");
    scanf("%d",&n);
    printf("Enter the Array: ");

    for(i=0; i<n; i++)
    {
        scanf("%d",&arr[i]);
        sum=sum+arr[i];
    }
    printf("The sum of array is %d.\n",sum);

    return 0;
}
```

**Sample output:**

```
How many elements: 5
Enter the Array: 5 3 2 1 4
The sum of array is 15.
```

```
Process returned 0 (0x0) execution time : 4.260 s
Press any key to continue.
```

4) Write a program to find whether the array of integers contains a duplicate number.

**Code:**

```
#include<stdio.h>
int main()
{
    int arr[100], i, n, j, flag=0;
    printf("Enter the size of the array:");
    scanf("%d", &n);
    printf("Enter %d element in the array:",n);

    for(i=0; i<n; i++)
        scanf("%d", &arr[i]);

    for(i=0; i<n; i++)
    {
        for(j=i+1; j<n; j++)
        {
            if(arr[i] == arr[j])
            {
                flag=1;
                break;
            }
        }
    }

    if (flag==0)
        printf("No Duplicate in this array\n");
    else
        printf("Duplicate value is present in this array\n");
    return 0;
}
```

**Sample output:**

```
Enter the size of the array:10
Enter 10 element in the array:1 2 3 4 5 6 7 8 9 2
Duplicate value is present in this array

Process returned 0 (0x0)    execution time : 9.208 s
Press any key to continue.
```

**5) Write a program to insert a number at a given location in an array.**

#include <stdio.h>

**code:**

```
#include<stdio.h>
int main()
{
    int arr[50], pos, i, n, value, size;

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

    printf("\nEnter %d elements: ", n);

    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    printf("\nPlease enter the location where you want to insert a new element : ");
    scanf("%d", &pos);

    printf("\nEnter the value : ");
    scanf("%d", &value);

    for (i = n - 1; i >= pos-1; i--)
        arr[i+1] = arr[i];

    arr[pos-1] = value;

    printf("\nResultant array is : ");

    for (i = 0; i <= n; i++)
        printf("%d ", arr[i]);
```

```
    return 0;
```

**Sample output:**

```
Enter number of elements in the array: 5
Enter 5 elements: 0 1 2 4 5
Please enter the location where you want to insert a new element : 4
Please enter the value : 3
Resultant array is : 0 1 2 3 4 5
Process returned 0 (0x0)   execution time : 10.631 s
Press any key to continue.
```

**6) Write a program to delete a number from a given location in an array.**

**Code:**

```
#include <stdio.h>

int main()
{
    int arr[100], pos,i,n;

    printf("Enter number of elements in array : ");
    scanf("%d",&n);

    printf("Enter %d elements : ",n);
    for (i=0; i<n; i++)
        scanf("%d",&arr[i]);

    printf("Enter the location where you wish to delete element : ");
    scanf("%d",&pos);

    if (pos>= n+1)
        printf("Deletion not possible\n");

    else
    {
        for (i= pos-1 ;i<n-1 ;i++ )
            arr[i] = arr[i+1];

        printf("Resultant array is : ");
        for(i=0; i<n-1; i++)
            printf("%d\n", arr[i]);
    }
}
```

```
        return 0;  
    }
```

**Sample output:**

```
Enter number of elements in array : 4  
Enter 4 elements : 0 1 2 4  
Enter the location where you wish to delete element : 3  
Resultant array is : 0 1 4  
Process returned 0 (0x0) execution time : 8.923 s  
Press any key to continue.
```

**7) Write a program to merge two sorted arrays.**

**code:**

```
#include <stdio.h>  
  
int main()  
{  
    int x,y,arr[1000];  
    printf("Enter numbers of 1st array : ");  
    scanf("%d",&x);  
    int s1[x+1];  
    printf("Enter %d numbers in this array : ",x);  
  
    for(int i=0; i<x; i++)  
    {  
        scanf("%d",&s1[i]);  
        arr[i]=s1[i];  
    }  
  
    printf("\nEnter numbers of 2nd array : ");  
    scanf("%d",&y);  
    int s2[y+1];  
    printf("Enter %d numbers in this array : ",y);  
  
    for(int i=x; i<x+y; i++)  
    {  
        scanf("%d",&s2[i]);  
        arr[i]=s2[i];  
    }
```

```
 }

printf("\nmarge element : "y);
for(int i=0; i<x+y; i++)
{
printf("%d ",arr[i]);
}
return 0;
}
```

**Sample output:**

```
Enter numbers of 1st array : 5
Enter 5 numbers in this array : 1 2 3 4 5

Enter numbers of 2nd array : 5
Enter 5 numbers in this array : 6 7 8 9 10

marge element : 1 2 3 4 5 6 7 8 9 10
Process returned 0 (0x0) execution time : 11.079 s
Press any key to continue.
```

8) Write programs for implementing the following sorting methods to arrange a list of integers/strings in ascending/descending order:

a) Bubble Sort

**Code: (integer)**

```
#include <stdio.h>

int main()
{
    int arr[100],i,j,n,swap;

    printf("Enter number of elements : ");
    scanf("%d", &n);

    printf("Enter %d integers : ", n);
    for (i=0; i<n; i++)
        scanf("%d", &arr[i]);

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

```

        for (j=0; j<n-i-1; j++)
        {
            if(arr[j]>arr[j+1])
            {
                swap=arr[j];
                arr[j]=arr[j+1];
                arr[j+1]=swap;
            }
        }
    }

    printf("Sorted list in ascending order : ");
    for (i=0; i<n; i++)
        printf("%d ",arr[i]);

    printf("\n");

    printf("Sorted list in decending order : ");
    for (i=n-1; i>=0; i--)
        printf("%d ",arr[i]);

    printf("\n");

    return 0;
}

```

**Sample output:**

```

Enter number of elements : 5
Enter 5 integers : 2 3 1 4 5
Sorted list in ascending order : 1 2 3 4 5
Sorted list in decending order : 5 4 3 2 1

Process returned 0 (0x0)  execution time : 6.264 s
Press any key to continue.

```

**b) Selection Sort**

**code:(integer)**

```

#include <stdio.h>
int main()
{
    int arr[100],i,j,n,swap,pos;

```

```

printf("Enter number of elements : ");
scanf("%d", &n);

printf("Enter %d integers : ", n);
for (i=0; i<n; i++)
scanf("%d",&arr[i]);

for (i=0; i<n-1; i++)
{
    pos = i;
    for (j=i+1; j<n; j++)
    {
        if (arr[pos]>arr[j])
        pos = j;
    }
    if (pos != i)
    {
        swap=arr[i];
        arr[i]=arr[pos];
        arr[pos]=swap;
    }
}
printf("Sorted list in ascending order : ");
for (i=0; i<n; i++)
printf("%d ",arr[i]);

printf("\n");

printf("Sorted list in decending order : ");
for (i=n-1; i>=0; i--)
printf("%d ",arr[i]);

printf("\n");
return 0;
}

```

**Sample output:**

```
Enter number of elements : 5
Enter 5 integers : 2 1 3 5 4
Sorted list in ascending order : 1 2 3 4 5
Sorted list in decending order : 5 4 3 2 1

Process returned 0 (0x0)    execution time : 23.152 s
Press any key to continue.
```

c) Insertion Sort

code:(integer)

```
#include<stdio.h>
int main ()
{
    int arr[100],i,j,n,swap;

    printf("Enter number of elements : ");
    scanf("%d", &n);

    printf("Enter %d integers : ", n);
    for (i=0; i<n; i++)
        scanf("%d", &arr[i]);

    for(i=1;i<n;i++)
    {
        int current=arr[i];
        j=i-1;

        while(arr[j]>current && j>=0)
        {
            arr[j+1]=arr[j];
            j--;
        }
        arr[j+1]=current;
    }

    printf("Sorted list in ascending order : ");
    for (i=0; i<n; i++)
        printf("%d ",arr[i]);
```

```

        printf("\n");

        printf("Sorted list in decending order : ");
        for (i=n-1; i>=0; i--)
            printf("%d ",arr[i]);

        printf("\n");
    }

```

**Sample output:**

```

Enter number of elements : 5
Enter 5 integers : 5 2 3 4 1
Sorted list in ascending order : 1 2 3 4 5
Sorted list in decending order : 5 4 3 2 1

Process returned 0 (0x0)    execution time : 11.147 s
Press any key to continue.

```

**9) Write programs for search an element from a list of integers/strings**

a) Liner Search

**code:(integer)**

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

```

int main()
{
    int arr[50],i,target,n;
    printf("How many elements do you want in the array : ");
    scanf("%d",&n);

    printf("Enter array elements : ");
    for(i=0; i<n; i++)
        scanf("%d",&arr[i]);

    printf("Enter element to search : ");
    scanf("%d",&target);
    for(i=0; i<n; ++i)
    {

```

```

        if(arr[i]==target)
        break;
    }

    if(i<n)
        printf("Target element found at location %d \n",i);
    else
        printf("Target element not found in an array.\n");

    return 0;
}

```

**Sample output:**

```

How many elements do you want in the array : 10
Enter array elements : 1 5 4 2 3 9 8 7 6 10
Enter element to search : 8
Target element found at location  6

Process returned 0 (0x0)  execution time : 32.103 s
Press any key to continue.

```

**b) Binary Search**

**code:(integer)**

```

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

    printf("Enter number of elements : ");
    scanf("%d", &n);

    printf("Enter %d integers : ", n);
    for (i=0; i<n; i++)
        scanf("%d", &arr[i]);

    printf("Enter element to be search: ");
    scanf("%d", &search);
    first = 0;
    last = n;
    mid = (first+last)/2;

```

```

        while(first <= last)
        {
            mid = (first+last)/2;
            if(arr[mid]<search)
                first = mid+1;
            else if(arr[mid]==search)
            {
                printf("\nThe number %d is found at Position %d", search, mid+1);
                break;
            }
            else
                last = mid-1;
        }
        if(first>last)
            printf("\nThe number, %d is not found in given Array", search);

        return 0;
    }

```

**Sample output:**

```

Enter number of elements : 10
Enter 10 integers : 1 2 3 4 5 6 7 8 9 10
Enter element to be search: 7

The number 7 is found at Position 7
Process returned 0 (0x0) execution time : 12.707 s
Press any key to continue.

```

**10) Write a program to read and display a matrix.**

**code:**

```

#include <stdio.h>

int main()
{
    int r,c;
    printf("Enter numbers of row : ");
    scanf("%d",&r);
    printf("Enter numbers of column : ");
    scanf("%d",&c);

```

```

int arr[r][c],i,j;

for(i=0; i<r; i++)
{
    for(j=0; j<c; j++)
        scanf("%d",&arr[i][j]);
}

printf("\nThe matrix is :");
for(i=0; i<r; i++)
{
    printf("\n");
    for(j=0; j<c; j++)
        printf("%d ",arr[i][j]);
}
return 0;
}

```

**Sample output:**

```

Enter numbers of row : 3
Enter numbers of column : 3
1 2 3
4 5 6
7 8 9

The matrix is :
1 2 3
4 5 6
7 8 9
Process returned 0 (0x0)   execution time : 13.198 s
Press any key to continue.

```

**11) Write a program to add and multiply two matrices.**

**code:**

```

#include<stdio.h>
int main()
{
    int mat1[10][10], mat2[10][10], mul[10][10], n, i, j, k, sum[10][10];

    printf("Enter the value of N : ");
    scanf("%d", &n);
    printf("Enter the elements of Matrix-A: \n");

```

```
for (i = 0; i < n; i++)
{
    for (j = 0; j < n; j++)
    {
        scanf("%d", &mat1[i][j]);
    }
}
```

```
printf("Enter the elements of Matrix-B:\n");
```

```
for (i = 0; i < n; i++)
{
    for (j = 0; j < n; j++)
    {
        scanf("%d", &mat2[i][j]);
    }
}
```

```
printf("\nThe sum of the two matrices is: ");
```

```
for(i=0; i<n; i++)
{
    printf("\n");
    for(j=0; j<n; j++)
    {
        sum[i][j] = mat1[i][j]+mat2[i][j];
        printf("%d ",sum[i][j]);
    }
}
printf("\n");
printf("\nThe product of the two matrices is: \n");
```

```
for (i = 0; i < n; i++)
{
    for (j = 0; j < n; j++)
    {
        mul[i][j] = 0;
        for (k = 0; k < n; k++)
        {
            mul[i][j] = mul[i][j] + mat1[i][k] * mat2[k][j];
        }
    }
}
```

```
    printf("%d ", mul[i][j]);
}
printf("\n");
}
return 0;
}
```

**Sample output:**

```
Enter the value of N : 3
Enter the elements of Matrix-A:
1 1 1
1 1 1
1 1 1
Enter the elements of Matrix-B:
1 1 1
1 1 1
1 1 1

The sum of the two matrices is:
2 2 2
2 2 2
2 2 2

The product of the two matrices is:
3 3 3
3 3 3
3 3 3

Process returned 0 (0x0)    execution time : 13.082 s
Press any key to continue.
```

**12) Write a program that takes a matrix A and finds its transpose AT and displays it.**

**code:**

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

```
int main()
```

```

{
    int r,c;
    printf("Enter numbers of row : ");
    scanf("%d",&r);
    printf("Enter numbers of column : ");
    scanf("%d",&c);

    int arr[r][c],i,j;

    for(i=0; i<r; i++)
    {
        for(j=0; j<c; j++)
        scanf("%d",&arr[i][j]);
    }

    printf("\nThe matrix is :");
    for(i=0; i<r; i++)
    {
        printf("\n");
        for(j=0; j<c; j++)
        printf("%d ",arr[j][i]);
    }
    return 0;
}

```

**Sample output:**

```

Enter numbers of row : 3
Enter numbers of column : 3
1 1 1
2 2 2
3 3 3

The Transpose matrix is :
1 2 3
1 2 3
1 2 3
Process returned 0 (0x0) execution time : 6.647 s
Press any key to continue.

```

**13) Write a program that computes the sum of diagonal elements of a square matrix.**

**code:**

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

```
int main()
```

```

{
    int r,c,sum=0;
    printf("Enter numbers of row : ");
    scanf("%d",&r);
    printf("Enter numbers of column : ");
    scanf("%d",&c);

    int arr[r][c],i,j;

    for(i=0; i<r; i++)
    {
        for(j=0; j<c; j++)
        scanf("%d",&arr[i][j]);
    }

    for(i=0; i<r; i++)
    {
        for(j=0; j<c; j++)
        {
            if(i==j)
            sum=sum+arr[i][j];
        }
    }
    printf("\nThe sum of diagonal elements of a square matrix is : %d\n",sum);

    return 0;
}

```

**Sample output:**

```

Enter numbers of row : 3
Enter numbers of column : 3
1 2 3
4 5 6
7 8 9
The sum of diagonal elements of a square matrix is : 15
Process returned 0 (0x0) execution time : 9.556 s
Press any key to continue.

```

**14) Write a program to find the length of a string.**

**code:**

```

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

```

```
int len=0;
char s[1000];
printf("Enter the string : ");
gets(s);

for (int i = 0; s[i] != '\0'; i++)
len++;
printf("\nLength of the string : %d\n", len);
return 0;
}
```

**Sample output:**

```
Enter the string : mamun nature
Length of the string : 12
Process returned 0 (0x0) execution time : 3.921 s
Press any key to continue.
```

**15) Write a program to concatenate two strings.**

**code:**

```
#include <stdio.h>
int main()
{
    char str1[100],str2[100];
    printf("Enter first string: ");
    gets(str1);
    printf("Enter second string: ");
    gets(str2);

    printf("\nConcatenated string = '%s %s'\n",str1,str2);

    return 0;
}
```

**Sample output:**

```
Enter first string: mamun
Enter second string: natore

Concatenated string = 'mamun natore'

Process returned 0 (0x0)  execution time : 7.306 s
Press any key to continue.
```

**16) Write a program to compare two strings.**

**code:**

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

int main()
{
    int i,count=0;
    char str1[100],str2[100];
    printf("Enter 1st string : ");
    gets(str1);
    printf("Enter 2nd string : ");
    gets(str2);
    int len1=strlen(str1);
    int len2=strlen(str2);

    if(len1!=len2)
        printf("both are not same\n");
    else
    {
        for(i=0; i<len1; i++)
        {
            if(str1[i]==str2[i])
                count++;
        }

        if(count==len1)
            printf("both are same\n");
        else
            printf("both are not same\n");
    }
    return 0;
}
```

**Sample output:**

```
Enter the first string : mamun
Enter the second string : mamun

comparison result : strings are same

Process returned 0 (0x0)  execution time : 5.284 s
Press any key to continue.
```

**17) Write a program to reverse a given string.**

**code:**

```
#include<stdio.h>
#include<string.h>
int main()
{
    char str[100];
    printf("Enter the string : ");
    gets(str);
    int length=strlen(str);

    printf("Reverse array : ");
    for(int i=length-1; i>=0; i--)
    {
        printf("%c",str[i]);
    }

    return 0;
}
```

**Sample output:**

```
Enter the string : aaaaabbbbbccccc
Reverse array : cccccccbbbbbbaaaaa
Process returned 0 (0x0)  execution time : 5.024 s
Press any key to continue.
```

**18) Write a program to extract a substring from a given string.**

**code:**

```

#include<stdio.h>
#include<string.h>
int main()
{
    int x,y;
    char str[100];
    printf("Enter the string : ");
    gets(str);
    int length=strlen(str);

    printf("Input the position to start extraction : ");
    scanf("%d",&x);
    printf("Input the length of substring : ");
    scanf("%d",&y);

    printf("The substring retrieve from the string is : ");
    for(int i=x; i<x+y; i++)
    {
        printf("%c",str[i]);
    }
    printf("\n");

    return 0;
}

```

**Sample output:**

```

Enter the string : mamun nature
Input the position to start extraction : 6
Input the length of substring : 5
The substring retrieve from the string is : nator

```

```

Process returned 0 (0x0)  execution time : 6.663 s
Press any key to continue.

```

**19) Write a program to insert a string in the main text.**

**code:**

```

#include <stdio.h>
#include <string.h>

int main()

```

```

{
    char arr1[100],arr2[100],arr3[100];

    printf("Enter First String:");
    gets(arr1);
    printf("Enter Second String:");
    gets(arr2);

    int pos;
    printf("Enter the position where the item has to be inserted: ");
    scanf("%d",&pos);

    int len1 = strlen(arr1);
    int len2 = strlen(arr2);
    int len3=len1+len2+1;

    int k=0;

    for(int i=0; i<len3; i++)
    {
        if(i<pos)
            arr3[i]=arr1[i];
        else if(i>=pos && i<pos+len2)
        {
            arr3[i]=arr2[k];
            k++;
        }
        else
        {
            arr3[i]=arr1[k];
            k++;
        }
    }

    for(int i=0; i<len3-1; i++)
    {
        printf("%c",arr3[i]);
    }

    return 0;
}

```

**Sample output:**

```
Enter First String:mamun natre
Enter Second String:islam
Enter the position where the item has to be inserted: 6
mamun islam nator
Process returned 0 (0x0) execution time : 14.368 s
Press any key to continue.
```

20) Write a program to delete every occurrence of a pattern (**character**) from a text.

**code:**

```
#include<stdio.h>
#include<string.h>
int main()
{
    char str[100],ch;
    printf("Enter the string : ");
    gets(str);
    int len=strlen(str);
    printf("Enter the character for delete : ");
    scanf("%c",&ch);

    printf("\nAfter removing '%c' ,we get the string : ",ch);
    for(int i=0; i<len; i++)
    {
        if(str[i]!=ch)
        printf("%c",str[i]);
    }
    printf("\n");

    return 0;
}
```

**Sample output:**

```
Enter the string : geeksforgeeks
Enter the character for delete : e

After removing 'e' ,we get the string : gksforgks

Process returned 0 (0x0) execution time : 7.370 s
Press any key to continue.
```

**21) Write a program to replace a pattern (**character**) with another pattern in the text.**

**code:**

```
#include<stdio.h>
#include<string.h>
int main()
{
    char str[100],old,New;
    printf("Enter the string : ");
    gets(str);
    int len=strlen(str);
    printf("Enter the character for replace : ");
    scanf("%c %c",&old,&New);

    printf("\nReplace array : ");
    for(int i=0; i<len; i++)
    {
        if(str[i]==old)
            printf("%c",New);
        else
            printf("%c",str[i]);
    }
    return 0;
}
```

**Sample output:**

```
Enter the string : geeksforgeeks
Enter the character for replace : e i

Replace array : giiksforgiiks

Process returned 0 (0x0)  execution time : 14.551 s
Press any key to continue.
```

**22) Write a program to develop the first pattern matching algorithm (Brute Force based).**

**code:**

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

int main()
```

```
{  
    int i,j,len1,len2,count=0;  
    char arr[] = "AAB AAC AAD AABA AABA";  
    char str[] = "AABA";  
  
    len1 = strlen(arr);  
    len2 = strlen(str);  
  
    for (i=0; i<=len1-len2; i++)  
    {  
        for (j=0; j<len2; j++)  
        {  
            if (arr[i + j] != str[j])  
                break;  
        }  
        if (j == len2)  
        {  
            count++;  
            printf("Pattern found at index %d \n", i+1,i+1);  
        }  
    }  
    if(count==0)  
        printf("Pattern is not found\n");  
    return 0;  
}
```

**Sample output:**

```
Pattern found at index 1  
Pattern found at index 10  
Pattern found at index 14  
  
Process returned 0 (0x0)  execution time : 0.028 s  
Press any key to continue.
```

**23) Write a program to develop the second pattern matching algorithm (Finite Automata based).**

**24) Write a program that uses functions to perform the following operations on singly linked list:**

- a) Creation
- b) Insertion
- c) Deletion
- d) Traversal

**25) Write a program to create a circular linked list. Perform insertion and deletion at the beginning and end of the list.**

**26) Write programs that uses stack operations to convert a given infix expression into its postfix equivalent. Implement the stack using an array.**

**27) Write a Program in C to Implement**

- a) Stacks using arrays
- b) Stacks using linked list
- c) Queue using arrays
- d) circular queue using arrays

**28) Write a program to evaluate a postfix expression.**

**29) Write a program to convert an infix notation to postfix notation**

**30) Write a program to calculate the factorial of a given number.**

**code:**

```
#include<stdio.h>

int main()
{
    int n,sum=1;
    printf("Enter an integer number : ");
    scanf("%d",&n);

    for(int i=1;i<=n;i++)
    {
        sum=sum*i;
```

```
    }
    printf("%d! = %d\n",n,sum);

    return 0;
}
```

**Sample output:**

```
Enter an integer number : 5
5! = 120

Process returned 0 (0x0)  execution time : 1.253 s
Press any key to continue.
```

31) Write a program to print the Fibonacci series using recursion.

**code:**

```
#include<stdio.h>

int Fibonacci(int c)
{
    if (c==0)
        return 0;
    else if(c==1)
        return 1;
    else
        return (Fibonacci(c-1) + Fibonacci(c-2));
}

int main()
{
    int n,c=0;

    printf("Enter an integer number : ");
    scanf("%d",&n);

    printf("\nFibonacci series : ");

    for (int i=1 ; i<=n; i++)
    {
        printf("%d ",Fibonacci(c));
        c++;
    }
}
```

```
    }
    printf("\n");
    return 0;
}
```

**Sample output:**

```
Enter an integer number : 10

Fibonacci series : 0 1 1 2 3 5 8 13 21 34

Process returned 0 (0x0)  execution time : 2.838 s
Press any key to continue.
```

**32) Write a program to solve the Towers of Hanoi Problem using recursion. ([copy from online](#)) code:**

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<math.h>
#include<assert.h>
#include<stdbool.h>
#include<limits.h>

void towers(int, char, char, char);
int main()
{
    int num;
    printf("Enter the number of disks : ");
    scanf("%d", &num);
    printf("The sequence of moves involved in the Tower of Hanoi are :\n");
    towers(num, 'A', 'C', 'B');
    printf("\n");
    return 0;
}
void towers(int num, char frompeg, char topeg, char auxpeg)
{
    if (num == 1)
    {
        printf("\n Move disk 1 from peg %c to peg %c", frompeg, topeg);
        return;
    }
    towers(num - 1, frompeg, auxpeg, topeg);
    printf("\n Move disk %d from peg %c to peg %c", num, frompeg, topeg);
    towers(num - 1, auxpeg, topeg, frompeg);
}
```

```

    }
    towers(num - 1, frompeg, auxpeg, topeg);
    printf("\n Move disk %d from peg %c to peg %c", num, frompeg, topeg);
    towers(num - 1, auxpeg, topeg, frompeg);
}

```

**Sample output:**

```

Enter the number of disks : 3
The sequence of moves involved in the Tower of Hanoi are :

Move disk 1 from peg A to peg C
Move disk 2 from peg A to peg B
Move disk 1 from peg C to peg B
Move disk 3 from peg A to peg C
Move disk 1 from peg B to peg A
Move disk 2 from peg B to peg C
Move disk 1 from peg A to peg C

Process returned 0 (0x0)    execution time : 1.134 s
Press any key to continue.

```

**33) Write a program to calculate the factorial of a given number.(recursion)**

**code:**

```

#include<stdio.h>
int find_factorial(int n)
{
    if(n>=1)
        return n*find_factorial(n-1);
    else
        return 1;
}
int main()
{
    int n;
    printf("Enter a positive integer: ");
    scanf("%d",&n);
    printf("\n%d! = %d\n", n,find_factorial(n));
    return 0;
}

```

**Sample output:**

```

Enter a positive integer: 5
5! = 120

Process returned 0 (0x0)    execution time : 0.852 s
Press any key to continue.

```

**34) Write a program to demonstrate several tree operations**

- a) Insertion
- b) Inorder
- c) Preorder
- d) Postorder

**35) Write a program to create a binary search tree**

**36) Write a program to create a graph of n vertices using an adjacency list.**

**37) Write a program to implement Warshall's algorithm to find the path matrix**

**38) Write a program to implement Warshall's algorithm to find the all pair shortest path**

**39) Create a word processor using C/C++. It should be a menu driven program.**

- a) Text must be read from the file and after processing written into file
- b) Number of line, characters, words, etc
- c) Find a pattern from the text
- d) Insert, delete, append a string
- e) Replace a string

**40) Using C structure create student records of ECE L2-I students. It should be menu driven program.**

- a) Fields are Roll no, Name, CGPA, address
- b) Display the records.
- c) Insert a new record in desire location
- d) Delete a record from a desire location
- e) Searching a record by Roll no
- f) Sorting the records

**code:**

```
#include <stdio.h>
struct student
{
    int roll;
    char firstName[50];
```

```

        double cgpa;
        char address[100];
    };

int main()
{
    struct student s[70];

    int i,n;
    printf("How many student in class :");
    scanf("%d",&n);

    for (i=1; i<=n; i++)
    {
        printf("\nEnter information of students%d:\n",i);

        printf("Enter roll: ");
        scanf("%d", &s[i].roll);
        printf("Enter first name: ");
        scanf("%s", s[i].firstName);
        printf("Enter CGPA: ");
        scanf("%lf", &s[i].cgpa);
        printf("Enter address: ");
        scanf("%s", s[i].address);
        printf("\n");
    }

    printf("\nDisplaying Information:\n\n");
    for (i=1; i<=n; i++)
    {
        printf("Information of student%d:\n",i);
        printf("Roll: %d\n",s[i].roll);
        printf("First name: ");
        puts(s[i].firstName);
        printf("CGPA: %.2lf\n",s[i].cgpa);
        printf("Address: ");
        puts(s[i].address);
        printf("\n");
    }
    return 0;
}

```

**Sample output:**

```
How many student in class :2
```

```
Enter information of students1:  
Enter roll: 2002130  
Enter first name: chanchol  
Enter CGPA: 3.44  
Enter address: Pirogachha,Rangpur
```

```
Enter information of students2:  
Enter roll: 20221  
Enter first name: mamun  
Enter CGPA: 3.25  
Enter address: Natore
```

**Displaying Information:**

```
Information of student1:  
Roll: 2002130  
First name: chanchol  
CGPA: 3.44  
Address: Pirogachha,Rangpur
```

```
Information of student2:  
Roll: 20221  
First name: mamun  
CGPA: 3.25  
Address: Natore
```

```
Process returned 0 (0x0) execution time : 95.913 s  
Press any key to continue.
```