

S.No: 1	Exp. Name: Design a C program which sorts the strings using array of pointers	Date: 2023-05-01
---------	---	------------------

Aim:
Design a C program that sorts the strings using array of pointers.

Sample Input output:

```
Sample Input-output -1:
Enter the number of strings: 2
Enter string 1: Tanya
Enter string 2: Code
Before Sorting
Tanya
Code
After Sorting
Code
Tanya
India
Japan
USA
Japan
Before Sorting
India
Japan
India
Japan
USA
```

Source Code:

```
stringsort.c
```

[ID: 22461059] Page No: 1

Srinidhi Parasharji Institute of Technology [2022-2026-CSE-B]

```
#include<stdio.h>
void main()
{
    char temp;
    int i,j,k;
    char str[100][10];
    printf("Enter the number of strings: ");
    scanf("%d",&k);
    for(i=0;i<k;i++)
    {
        printf("Enter string %d: ",i+1);
        scanf("%s",str[i]);
        if(strlen(str[i])>20)
            scanf("%s",str[i]);
    }
    printf("Before Sorting:\n");
    for(i=0;i<k;i++)
    {
        printf("%s\n",str[i]);
    }
    for(i=0;i<k-1;i++)
    {
        for(j=i+1;j>i;j--)
        {
            if(strcmp(str[i],str[j])>0)
            {
                temp=str[i];
                str[i]=str[j];
                str[j]=temp;
            }
        }
    }
    printf("After Sorting:\n");
    for(i=0;i<k;i++)
    {
        printf("%s\n",str[i]);
    }
}
```

[ID: 22461059] Page No: 2

Srinidhi Parasharji Institute of Technology [2022-2026-CSE-B]

After Sorting
Code
Tanya

User Output

Enter the number of strings:
3
Enter string 1:
Tanya
Enter string 2:
Code
Enter string 3:
India
Before Sorting
Tanya
Code
India
After Sorting
Code
Tanya
India

[ID: 22461059] Page No: 3

Srinidhi Parasharji Institute of Technology [2022-2026-CSE-B]

S.No: 2	Exp. Name: Write a C Program to Search a Key element using Linear search Technique	Date: 2023-05-01
---------	--	------------------

Aim:
Write a program to search a key element with in the given array of elements using [Linear search](#) process.

At the time of execution, the program should print the message on the console as:

Enter value of n :

For example, if the user gives the input as:

Enter value of n : 3

Next, the program should print the messages one by one on the console as:

Error element for a[0] :

Error element for a[1] :

Error element for a[2] :

If the user gives the input as:

Enter element for a[0] :

Enter element for a[1] :

Enter element for a[2] :

Enter value of n : 3

Next, the program should print the message on the console as:

Enter key element :

If the user gives the input as:

Enter key element : 56

then the program should print the result as:

The key element 56 is found at the position 0

Similarly if the key element is given as 25 for the above one dimensional array elements then the program should print the output as "The key element 25 is not found in the array".

Fill in the missing code so that it produces the desired result:

Source Code:

```
linearSearch.c
```

[ID: 22461059] Page No: 4

Srinidhi Parasharji Institute of Technology [2022-2026-CSE-B]

```
#include<stdio.h>
int main()
{
    int i,j,n,flag=0;
    printf("Enter value of n : ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter element for a[%d] : ",i);
        scanf("%d",&a[i]);
    }
    printf("Enter key element : ");
    scanf("%d",&k);
    for(i=0;i<n;i++)
    {
        if(a[i]==k)
        {
            flag=1;
            break;
        }
    }
    if(flag==1)
    {
        printf("The key element %d is found at the position %d",k,i);
    }
    else
    {
        printf("The key element %d is not found in the array",k);
    }
    printf("\n");
}
```

[ID: 22461059] Page No: 5

Srinidhi Parasharji Institute of Technology [2022-2026-CSE-B]

Test Case -2

User Output

Enter value of n :
3
Enter element for a[0]:
101
Enter element for a[1]:
102
Enter element for a[2]:
103
Enter element for a[3]:
104
Enter element for a[4]:
105
Enter element for a[5]:
106
Enter key element :
107
Enter key element :
108

The key element 108 is not found in the array

[ID: 22461059] Page No: 6

Srinidhi Parasharji Institute of Technology [2022-2026-CSE-B]

S.No: 3	Exp. Name: Write a C Program to Search a Key element using Binary search Technique	Date: 2023-05-03
---------	--	------------------

Aim:
Write a program to search a key element in the given array of elements using [Binary search](#).

At the time of execution, the program should print the message on the console as:

Enter value of n :

For example, if the user gives the input as:

Enter value of n : 3

Next, the program should print the messages one by one on the console as:

Error element for a[0] :

Error element for a[1] :

Error element for a[2] :

If the user gives the input as:

Enter element for a[0] :

Enter element for a[1] :

Enter element for a[2] :

Enter value of n : 3

Next, the program should print the message on the console as:

Enter key element :

If the user gives the input as:

Enter key element : 56

then the program should print the result as:

After sorting the elements in the array are

Value of a[0] = 33

Value of a[1] = 56

Value of a[2] = 89

The key element 56 is found at the position 2

Similarly if the key element is given as 25 for the above one dimensional array elements then the program should print the output as "The key element 25 is not found in the array".

Fill in the missing code so that it produces the desired result:

Source Code:

```
binarySearch.c
```

[ID: 22461059] Page No: 7

Srinidhi Parasharji Institute of Technology [2022-2026-CSE-B]

```
#include<stdio.h>
void main()
{
    int a[3],i,temp,k,flag=0;
    printf("Enter value of n : ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter element for a[%d] : ",i);
        scanf("%d",&a[i]);
    }
    for(i=0;i<n-1;i++)
    {
        for(j=i+1;j>i;j--)
        {
            if(a[i]>a[j])
            {
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }
        }
    }
    printf("After sorting the elements in the array are\n");
    for(i=0;i<n;i++)
    {
        printf("Value of a[%d] = %d\n",i,a[i]);
    }
    printf("Enter key element : ");
    scanf("%d",&k);
    printf("After sorting the elements in the array are\n");
    for(i=0;i<n;i++)
    {
        if(a[i]==k)
        {
            flag=1;
            break;
        }
    }
    if(flag==1)
    {
        printf("The key element %d is found at the position %d",k,i);
    }
    else
    {
        printf("The key element %d is not found in the array",k);
    }
}
```

[ID: 22461059] Page No: 8

Srinidhi Parasharji Institute of Technology [2022-2026-CSE-B]

25
Enter element for a[0]:
15
Enter element for a[1]:
23
Enter key element :
45

After sorting the elements in the array are

Value of a[0] = 33
Value of a[1] = 56
Value of a[2] = 89

The key element 56 is not found in the array

[ID: 22461059] Page No: 9

Srinidhi Parasharji Institute of Technology [2022-2026-CSE-B]

Test Case -1

User Output

Enter value of n :
2
Enter elements for a[0]:
80
Enter elements for a[1]:
80
Enter key element :
50

After sorting the elements in the array are

Value of a[0] = 80
Value of a[1] = 80

The key element 50 is not found in the array


```
#include<stdio.h>
int main()
{
    int size;
    int arr[10];
    printf("Enter array size : ");
    scanf("%d",&size);
    arr = (int*) malloc(sizeof(size));
    printf("Enter elements : ");
    for(int i=0;i<size;i++)
    {
        scanf("%d",&arr[i]);
    }
    printf("Before sorting the elements are : ");
    printarray(arr,size);
    return 0;
}
int shellSort(int arr[],int n)
{
    int gap=n/10;
    for(gap=gap/10;gap>0;
    {
        for(int i=gap;i<n;i++)
        {
            temp=arr[i];
            for(j=i-gap;j>=0;j-=gap)
            {
                if(arr[j]>temp)
                {
                    arr[j+gap]=arr[j];
                }
                else
                {
                    arr[j+gap]=temp;
                }
            }
        }
    }
    void printarray(int arr[],int n)
    {
        for(int i=0;i<n;i++)
        {
            printf("%d ",arr[i]);
        }
        printf("\n");
    }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1	
User Output	<pre>Enter array size : 5</pre>

<pre>int arr[5] elements : 12 32 43 56 78 Before sorting the elements are : 12 32 43 56 78 After sorting the elements are : 12 32 43 56 78</pre>	<p>ID: 2245161982 Page No. 19</p>
--	-------------------------------------

S.No: 8 Exp. Name: Write a C program to sort the elements using Bubble Sort technique Date: 2023-05-04

Aim:
Write a program to sort the given elements using bubble sort technique.

At the time of execution, the program should print the message on the console as
Enter value of n :

For example, if the user gives the input as:
Enter value of n : 3

Next, the program should print the messages one by one on the console as:
Enter element for a[0] :
Enter element for a[1] :
Enter element for a[2] :

If the user gives the input as:
Enter element for a[0] : 22
Enter element for a[1] : 33
Enter element for a[2] : 12

The program should print the result as:
Before sorting the elements in the array are
Value of a[0] = 22
Value of a[1] = 33
Value of a[2] = 12
After sorting the elements in the array are
Value of a[0] = 12
Value of a[1] = 22
Value of a[2] = 33

Fill in the missing code so that it produces the desired result.
Source Code:
[BubbleSortDemo3.c](#)

ID: 2245161982 | Page No. 21

Sirishtha Banerjee| Institute of Technology | 2022-2026-CSE-B

```
#include<stdio.h>
void main()
{
    int arr[10],i,j,n,temp;
    printf("Enter value of n : ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter element for a[%d] : ",i);
        scanf("%d",&arr[i]);
    }
    printf("Before sorting the elements in the array are: ");
    for(i=0;i<n;i++)
    {
        printf("%d ",arr[i]);
    }
    for(i=0;i<n;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(arr[i]>arr[j])
            {
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
    printf("After sorting the elements in the array are: ");
    for(i=0;i<n;i++)
    {
        printf("Value of a[%d] = %d\n",i,arr[i]);
    }
    printf("\n");
}
```

Execution Results - All test cases have succeeded!

Test Case - 1	
User Output	<pre>Enter value of n : 5 Enter element for a[0] : 24 Enter element for a[1] : 34 Enter element for a[2] : 28 Enter element for a[3] : 22 before sorting the elements in the array are Value of a[0] : 24</pre>

<pre>value of a[0] = 25 value of a[1] = 28 value of a[2] = 33 value of a[3] = 32 value of a[4] = 36</pre>	<p>ID: 2245161982 Page No. 22</p>
---	-------------------------------------

S.No: 9 Exp. Name: Write a program to sort Ascending order the given elements using quick sort technique Date: 2023-05-07

Aim:
Write a program to sort the given elements using quick sort technique.

Note: Pick the first element as pivot. You will not be awarded marks if you do not follow this instruction.

At the time of execution, the program should print the message on the console as
Enter array size :

For example, if the user gives the input as:
Enter array size :

Next, the program should print the following message on the console as:
Enter 5 elements :

If the user gives the Input as:
Enter 5 elements : 34 67 12 45 22

then the program should print the result as:
Before sorting the elements are : 34 67 12 45 22
After sorting the elements are : 12 22 34 45 67

Note: Do use the printf() function with a newline character (\n).

Source Code:
[QuickSortIn1.c](#)

ID: 2245161982 | Page No. 24

Sirishtha Banerjee| Institute of Technology | 2022-2026-CSE-B

```
#include<stdio.h>
void print()
{
    int arr[10], i, n;
    printf("Enter array size : ");
    scanf("%d",&n);
    printf("Enter %d elements : ", n);
    for (i = 0; i < n; i++)
    {
        start ("%",arr[i]);
    }
    printf("Before sorting the elements are : ");
    display(arr,n);
    swap(arr,n);
    printf("After sorting the elements are : ");
    display(arr,n);
}

void display(int arr[10], int n) {
    int i;
    for(i=0;i<n;i++)
    {
        printf("%d ",arr[i]);
    }
    printf("\n");
}

int partition(int arr[10], int lb, int ub)
{
    int pivot,low,high;
    pivot=arr[lb];
    low=lb;
    high=ub;
    while (arr[low]>arr[pivot])
    {
        low++;
    }
    while (arr[high]<arr[pivot])
    {
        high--;
    }
    if (low>high)
    {
        temp=arr[low];
        arr[low]=arr[high];
        arr[high]=temp;
    }
    return high;
}

void quickSort(int arr[10], int low, int high)
{
    if(low>high)
    {
        return;
    }
    int partition(int arr[10], int low, int high);
    quickSort(arr,low,partition);
    quickSort(arr,partition+1,high);
}
```

Execution Results - All test cases have succeeded!

Test Case - 1	
User Output	<pre>Enter array size : 5 Enter 5 elements : 34 67 12 45 22 Before sorting the elements are : 34 67 12 45 22 After sorting the elements are : 12 22 34 45 67</pre>

<pre>Test Case - 1 User Output Enter array size : 5 Enter 5 elements : 34 67 12 45 22 Before sorting the elements are : 34 67 12 45 22 After sorting the elements are : 12 22 34 45 67</pre>	<p>ID: 2245161982 Page No. 25</p>
--	-------------------------------------

S.No: 10 Exp. Name: Write a C program to sort the given elements using Heap sort technique Date: 2023-05-07

Aim:
Write a program to sort (descending order) the given elements using heap sort technique.

Note: Do use the printf() function with a newline character (\n).

Source Code:
[HeapSortIn1.c](#)

ID: 2245161982 | Page No. 27

Sirishtha Banerjee| Institute of Technology | 2022-2026-CSE-B


```
prlist(node);
```

Execution Results - All test cases have succeeded!

Test Case - 1

```
User Output
how many numbers you want to enter: 4
Enter number: 1
Enter number: 2
Enter number: 3
Enter number: 4
Given linked list: 1<-->2<-->3<-->4
Reversed linked list: 4<-->3<-->2<-->1
```

```
Test Case - 2
```

```
User Output
how many numbers you want to enter: 3
Enter number: 1
Enter number: 2
Enter number: 3
Given linked list: 1<-->2<-->3
Reversed linked list: 3<-->2<-->1
```

[ID: 224516582] Page No: 91

S.No: 23 Exp. Name: Program to insert into BST and Inorder using In-order, Pre-order and Post-order Date: 2023-06-23

Aim:
Write a program to create a binary search tree of integers and perform the following operations using linked list:
1. Insert a node
2. In-order traversal
3. Pre-order traversal
4. Post-order traversal
Source Code:
BinarySearchTree.c

[ID: 224516582] Page No: 92

```
#include <cslib.h>
#include <csliblib.h>
struct node
{
    int data;
    struct node *left,*right;
};

typedef struct node *BSTMDE;

BSTMDE readnode(BSTMDE *item)
{
    BSTMDE temp = (BSTMDE)malloc(sizeof(struct node));
    temp->data=item->data;
    temp->left=item->left;
    temp->right=item->right;
    return temp;
}

void inorder(BSTMDE root)
{
    if(item==NULL)
        return;
    inorder(item->left);
    printf("%d ", item->data);
    inorder(item->right);
}

void preoder(BSTMDE root)
{
    if(item==NULL)
        return;
    printf("%d ", item->data);
    preoder(item->left);
    preoder(item->right);
}

void postorder(BSTMDE root)
{
    if(item==NULL)
        return;
    postorder(item->left);
    postorder(item->right);
    printf("%d ", item->data);
}

BSTMDE inserthead(BSTMDE node,int ele)
{
    if(item==NULL)
        printf("Successfully inserted.\n");
    else if(item->data>ele)
        inserthead(item->left,ele);
    else if(item->data<ele)
        inserthead(item->right,ele);
    else
        inserthead(item,ele);
}

BSTMDE insertnode(BSTMDE node,int ele)
{
    if(item==NULL)
        printf("Successfully inserted.\n");
    else if(item->data>ele)
        insertnode(item->left,ele);
    else if(item->data<ele)
        insertnode(item->right,ele);
    else
        inserthead(item,ele);
}
```

Srinivas Ramanaiah Institute of Technology [ID: 224516582] Page No: 93

```
void main()
{
    int n,opt;
    BSTMDE root=NULL;
    while(1)
    {
        printf("1.Insert 2.Inorder Traversal 3.Preorder Traversal 4.Postorder Traversal\n5.Exit\n");
        scanf("%d",&opt);
        switch(opt)
        {
            case 1:
                printf("Enter an element to be inserted : ");
                scanf("%d",&n);
                root=inserthead(root,n);
                break;
            case 2:
                printf("Elements of the BST (In-order traversal): ");
                postorder(root);
                printf("\n");
                break;
            case 3:
                printf("Elements of the BST (Pre-order traversal): ");
                preoder(root);
                printf("\n");
                break;
            case 4:
                printf("Elements of the BST (Post-order traversal): ");
                postorder(root);
                printf("\n");
                break;
            case 5:
                if(root==NULL)
                    printf("Binary Search Tree is empty.\n");
                else
                    printf("Elements of the BST (In-order traversal): ");
                    postorder(root);
                    printf("\n");
                break;
            default:
                printf("Enter an option : ");
                scanf("%d",&n);
                if(n==1)
                    inserthead(root,n);
                else if(n==2)
                    postorder(root);
                else if(n==3)
                    preoder(root);
                else if(n==4)
                    postorder(root);
                else if(n==5)
                    printf("Binary Search Tree is empty.\n");
                else
                    printf("Elements of the BST (Post-order traversal): ");
                    postorder(root);
                    printf("\n");
                break;
        }
    }
}
```

[ID: 224516582] Page No: 94

Execution Results - All test cases have succeeded!

Test Case - 1

```
User Output
1.Insert 2.Inorder Traversal 3.Preorder Traversal 4.Postorder Traversal 5.Exit
Enter your option : 1
Enter an element to be inserted : 100
Successfully inserted.
```

```
1.Insert 2.Inorder Traversal 3.Preorder Traversal 4.Postorder Traversal 5.Exit
Enter your option : 2
```

```
Enter an element to be inserted : 200
Successfully inserted.
```

```
1.Insert 2.Inorder Traversal 3.Preorder Traversal 4.Postorder Traversal 5.Exit
Enter your option : 3
```

```
Elements of the BST (Pre-order traversal): 100 200 200 200
```

```
1.Insert 2.Inorder Traversal 3.Preorder Traversal 4.Postorder Traversal 5.Exit
Enter your option : 4
```

```
Elements of the BST (Post-order traversal): 200 200 200 200
```

```
1.Insert 2.Inorder Traversal 3.Preorder Traversal 4.Postorder Traversal 5.Exit
Enter your option : 5
```

[ID: 224516582] Page No: 95

```
65
Successfully inserted.
1.Insert 2.Inorder Traversal 3.Preorder Traversal 4.Postorder Traversal 5.Exit
Enter your option : 1
Enter an element to be inserted : 50
50
Successfully inserted.
1.Insert 2.Inorder Traversal 3.Preorder Traversal 4.Postorder Traversal 5.Exit
Enter your option : 4
Elements of the BST (Post-order traversal): 20 45 45 50 65 25
1.Insert 2.Inorder Traversal 3.Preorder Traversal 4.Postorder Traversal 5.Exit
Enter your option : 5
```

[ID: 224516582] Page No: 97

S.No: 24 Exp. Name: Write a Program to Search an element using Binary Search and Recursion Date: 2023-06-26

Aim:
Write a program to search the given element from a list of elements with [binary search](#) technique using recursion.

At the time of execution, the program should print the message on the console as:

Enter value of n :

For example, if the user gives the input as:

Enter n : 5

Next, the program should print the following message one by one on the console as:

Enter 5 elements :

If the user gives the input as:

Enter 5 elements : 33 55 22 44 11

Then the program should print the result as:

After sorting the elements are : 11 22 33 44 55

Next, the program should print the message on the console as:

Enter key element :

If the user gives the input as:

Enter key element : 11

Then the program should print the result as:

The given key element 11 is found at position : 8

Simply, If the key element is given as for the above example, then the program should print the output as:

The given key element 11 is not found

Note: Write the functions `read()`, `bubbleSort()`, `display()` and `binarySearch()` in [BinarySearch.c](#).

Source Code:

BinarySearch.c

[ID: 224516582] Page No: 98

```
#include <cslib.h>
void main()
{
    int arr[10],key,flag;
    printf("Enter value of n : ");
    scanf("%d", &n);
    read(arr);
    bubbleSort(arr);
    printf("The sorted elements are : ");
    display(arr);
    printf("Enter key element : ");
    scanf("%d", &key);
    flag=binarySearch(arr, 0, n - 1, key);
    if(flag==1)
        printf("The given key element %d is not found(%d, key);", key);
    else
        printf("The given key element %d is found at position : %d(n, key, flag);"
    }
}

void read(int arr[], int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        arr[i]=rand()%(100+1)+1;
    }
}

void bubbleSort(int arr[], int n)
{
    int i,j,temp;
    for(i=0;i<n-1;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(arr[j]<arr[i])
            {
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
}

void display(int arr[], int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        printf("%d", arr[i]);
    }
}
```

Srinivas Ramanaiah Institute of Technology [ID: 224516582] Page No: 99

Enter an element to be inserted : 300
Successfully inserted.
1.Insert 2.Inorder Traversal 3.Preorder Traversal 4.Postorder Traversal 5.Exit
Enter your option : 1
Enter an element to be inserted : 100
Successfully inserted.
1.Insert 2.Inorder Traversal 3.Preorder Traversal 4.Postorder Traversal 5.Exit
Enter your option : 2
Elements of the BST (Pre-order traversal): 100 300 200 200 200
1.Insert 2.Inorder Traversal 3.Preorder Traversal 4.Postorder Traversal 5.Exit
Enter your option : 3
Elements of the BST (In-order traversal): 100 200 200 200 300
1.Insert 2.Inorder Traversal 3.Preorder Traversal 4.Postorder Traversal 5.Exit
Enter your option : 4
Elements of the BST (Post-order traversal): 200 200 200 300 100
1.Insert 2.Inorder Traversal 3.Preorder Traversal 4.Postorder Traversal 5.Exit
Enter your option : 5

[ID: 224516582] Page No: 96

Test Case - 2

User Output

1.Insert 2.Inorder Traversal 3.Preorder Traversal 4.Postorder Traversal 5.Exit
Enter your option : 1
Enter an element to be inserted : 25
Successfully inserted.
1.Insert 2.Inorder Traversal 3.Preorder Traversal 4.Postorder Traversal 5.Exit
Enter your option : 2
Enter an element to be inserted : 89
Successfully inserted.
1.Insert 2.Inorder Traversal 3.Preorder Traversal 4.Postorder Traversal 5.Exit
Enter your option : 3
Enter an element to be inserted : 45
Successfully inserted.
1.Insert 2.Inorder Traversal 3.Preorder Traversal 4.Postorder Traversal 5.Exit
Enter your option : 4
Enter an element to be inserted : 89
Successfully inserted.
1.Insert 2.Inorder Traversal 3.Preorder Traversal 4.Postorder Traversal 5.Exit
Enter your option : 5

Srinivas Ramanaiah Institute of Technology [ID: 224516582] Page No: 96

```

int low=0,high=d-1,mid;
while(low<high)
{
    mid=(low+high)/2;
    if(a[mid]==key)
    {
        found=1;
        break;
    }
    else if(a[mid]<key)
    {
        low=mid+1;
    }
    else
        high=mid-1;
}
return found;
}

```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output
Enter value of n : 5
Enter 5 elements :
33 55 22 44 11
After sorting the elements are : 11 22 33 44 55
Enter key element :
11
The given key element 11 is found at position : 0

Test Case - 2

User Output
Enter value of n : 4
Enter 4 elements :
23 9 45 15
After sorting the elements are : 9 15 23 45
Enter key element :
23
The given key element 23 is not found

S.No: 25 Exp. Name: Graph traversals implementation - Breadth First Search Date: 2023-06-26

Aim:
Write a program to implement Breadth First Search of a graph.

Source Code:

GraphBFS.c

[ID: 224610586] Page No: 10

Srinivas Ramanujan Institute of Technology [ID: 2022-2026-CS-E8]

```

#include<csddin.h>
#include<csddlib.h>
struct node
{
    struct node *next;
    int vertex;
};
typedef struct node *GRAPH;
GRAPH graph[20];
int visited[20];
int source[20];
int rear=-1,front=-1;
int n;
void insertQueue(int vertex)
{
    if(front==MAX)
        printf("Queue overflow\n");
    else
    {
        front++;
        queue[front]=vertex;
    }
}
int isEmptyQueue()
{
    if(front==rear)
        return 1;
    else
        return 0;
}
int deleteQueue()
{
    if(front==rear)
        printf("Queue Underflow\n");
    else
    {
        int deleted;
        deleted=queue[front];
        front++;
        return deleted;
    }
}
void BFS(int v)
{
    int i;
    insertQueue(v);
    while(!isEmptyQueue())
    {
        deleted=queue[front];
        printf("%d ", deleted);
        visited[deleted]=1;
        for(i=1;i<=n;i++)
        {
            if(graph[v][i]==1)
                if(visited[i]==0)
                    insertQueue(i);
        }
    }
}

```

[ID: 224610586] Page No: 11

Srinivas Ramanujan Institute of Technology [ID: 2022-2026-CS-E8]

```

5
Enter source
1
Enter destination
2
Enter source
1
Enter destination
2
Enter source
2
Enter destination
1
Enter source
4
Enter destination
4
Enter source
4
Enter destination
3
Enter source
3
Enter destination
2
Enter source
2
Enter destination
1
Enter Start vertex for BFS :
1
BFS of graph
1
2
3
4

```

[ID: 224610586] Page No: 104

Srinivas Ramanujan Institute of Technology [ID: 2022-2026-CS-E8]

```

Test Case - 2

User Output
Enter the number of vertices :
4
Enter the number of edges :
3
1
2
3
4
Enter source
1
Enter destination
2
Enter source
2
Enter destination
1
Enter source
3
Enter destination
2
Enter source
4
Enter destination
3

```

[ID: 224610586] Page No: 105

Srinivas Ramanujan Institute of Technology [ID: 2022-2026-CS-E8]

```

S.No: 26 Exp. Name: Graph traversals implementation - Depth First Search Date: 2023-06-26

Aim:  
Write a program to implement Depth first Search for a graph.
Source Code:
```

GraphDFS.c

```

#include<csddin.h>
#include<csddlib.h>
struct node
{
    struct node *next;
    int vertex;
};
typedef struct node *GRAPH;
GRAPH graph[20];
int visited[20];
int source[20];
int dest[20];
int n;
void DFS(int i)
{
    GRAPH p;
    printf("Visiting %d", i);
    p=graph[i]-1;
    while(p!=NULL)
    {
        if(visited[p]==0)
        {
            printf(" Visiting %d", p);
            visited[p]=1;
            p=p->next;
        }
    }
}
void visit()
{
    int N,i,j,k,g,V;
    OMOC G,p;
    printf("Enter the number of vertices : ");
    scanf("%d",&N);
    printf("Enter the number of edges : ");
    scanf("%d",&E);
    g=(OMOC)nalloc(sizeof(struct node));
    p=(GRAPH)malloc(sizeof(G));
    q=(GRAPH)malloc(sizeof(G));
    if(graph[s]==NULL)
        graph[s]=q;
    else
    {
        p=graph[s];
        while(p->next!=NULL)
            p=p->next;
        p->next=q;
    }
    for(i=1;i<=N;i++)
    {
        printf("Enter source : ");
        scanf("%d",&s);
        printf("Enter destination : ");
        scanf("%d",&d);
        q=(OMOC)nalloc(sizeof(struct node));
        p=(GRAPH)malloc(sizeof(G));
        if(graph[s]==NULL)
            graph[s]=q;
        else
        {
            p=graph[s];
            while(p->next!=NULL)
                p=p->next;
            p->next=q;
        }
        q->vertex=i;
        q->next=NULL;
        p=q;
        p->next=q;
    }
}
void DFS()
{
    int i;
    printf("Visiting %d", i);
    visited[i]=0;
    printf("Visiting %d", i);
}

```

[ID: 224610586] Page No: 107

Srinivas Ramanujan Institute of Technology [ID: 2022-2026-CS-E8]

Execution Results - All test cases have succeeded!

Test Case - 1

User Output
Enter the number of vertices :
6
Enter the number of edges :
5
1
Enter source :
1
Enter destination :
2
Enter source :
1
Enter destination :
3
Enter source :
1
Enter destination :
4
Enter source :
1
Enter destination :
5
Enter source :
1
Enter destination :
6
Enter Start Vertex for DFS :
1
BFS of graph
1
2
3
4
5
6

Srinivas Ramanujan Institute of Technology [ID: 2022-2026-CS-E8]

Test Case - 2	
User Output	Enter the number of vertices : Enter the number of edges : Enter source : Enter destination : Enter destination : Enter source : Enter destination : Enter source : Enter destination : Enter destination : Enter source : Enter destination : Enter destination : Enter Start vertex for DFS : 1 DFS of graph : 1 2 3 4 5

Srinidhi Parameter Institute of Technology [2022-2026-CSE-8] Page No: 109

Execution Results - All test cases have succeeded!	
User Output	path =\nminimum cost is 50,cost:\nreturn 4;

Srinidhi Parameter Institute of Technology [2022-2026-CSE-8] Page No: 112

S.No: 29 Exp Name: Write a C program to Copy contents of one file into another File Date: 2023-06-26	
User Output	Enter the text with @ at end : CodeFutura is a Starting Company recognized by Government of India Given message is : CodeFutura is a Startup Company recognised by Government of India

Srinidhi Parameter Institute of Technology [2022-2026-CSE-8] Page No: 115

Execution Results - All test cases have succeeded!

S.No: 27 Exp Name: Travelling Sales Person problem using Dynamic programming Date: 2023-06-26

Aim: Write a C program to implement Travelling Sales Person problem using Dynamic programming.

Source Code:

TSP.c

[ID: 224610386] Page No: 110

```
#include<cs50.h>
int arr[10][10], completed[10];
int n, cost=0;
void toans(int i)
{
    int j,l;
    printf("The path is: ");
    for(j=0;j); j++)
        if(arr[i][j]==1)
            printf("%d ",j+1);
    printf("\n");
}
void tanscost(int city)
{
    int l,city1;
    completed[city]=1;
    printf("%d ",city+1);
    cost+=ans[city][city1];
    if(city1==0)
    {
        ans[0][0]=0;
        printf("%d ",ans[0][0]);
        cost+=ans[0][0];
        return;
    }
    else toans(city1);
}
int least(int c)
{
    int l,i,min=999;
    min=999;
    for(i=0;i<c;i++)
    {
        if((ans[c][i]<min) && (completed[i]==0))
        {
            min=ans[c][i];
            ans[c][i]=0;
            ans[i][c]=1;
            l=i-1;
        }
    }
    if(min==999)
        cost+=min;
    else return l;
}
int main()
{
    int i,j,k;
    for(i=0;i<10;i++)
        for(j=0;j<10;j++)
            for(k=0;k<10;k++)
                if(i!=j && j!=k && i!=k)
                    ans[i][j]=1;
    toans(0);
    tanscost(0);
}
```

[ID: 224610386] Page No: 111

S.No: 28 Exp Name: Write a C program to Open a File and to Print its contents on the screen Date: 2023-06-26

Aim: Follow the instructions given below to write a program to open a file and to print its contents on the screen.
Open your file "SampleText.txt" in write mode

- With the content in the file
- Close the file
- Open the file in read mode
- Read the content from file and print them on the screen
- Close the file

Source Code:

```
file.c
#include<cs50.h>
void main()
{
    FILE *fp;
    char ch;
    fp = fopen("SampleText.txt","w");
    printf("Enter the text with @ at end : ");
    while((ch = getchar()) != '@')
    {
        putchar(ch);
    }
    putchar(ch);
    fclose(fp);
    fp=fopen("SampleText.txt", "r");
    printf("Given message is : ");
    while(ch = getchar()) != '@'
    {
        putchar(ch);
    }
    printf("\n");
    fclose(fp);
}
```

[ID: 224610386] Page No: 113

Execution Results - All test cases have succeeded!

Test Case - 1	
User Output	

Enter the text with @ at end :
CodeFutura is a
Starting Company recognized by Government of India
Given message is : CodeFutura is a
Startup Company recognised by Government of India

Srinidhi Parameter Institute of Technology [2022-2026-CSE-8] Page No: 114

S.No: 29 Exp Name: Write a C program to Merge two files and stores their contents in another File Date: 2023-06-26

Aim: Write a program to copy contents of one file into another file. Follow the instructions given below to write a program to copy contents of one file into another file.

- Open a new file "Copyfile1.txt" in write mode
- Write the content onto the file
- Close the file
- Open existing file "SampleText1.txt" in read mode
- Create a new file "Copyfile2.txt" in write mode
- Copy the content from existing file to new file
- Close the file
- Open the copied file in read mode
- Read the file from it and print on the screen
- Close the file

Source Code:

```
Copyfile.c
#include<cs50.h>
void main()
{
    FILE *fp1,*fp2,*fp3;
    char ch;
    fp1=fopen("Copyfile1.txt","w");
    printf("Enter the text with @ at end : ");
    while((ch = getchar()) != '@')
    {
        putchar(ch);
    }
    putchar(ch);
    fclose(fp1);
    fp2=fopen("SampleText1.txt","r");
    fp3=fopen("Copyfile2.txt","w");
    while((ch = getchar()) != '@')
    {
        putchar(ch);
    }
    putchar(ch);
    fclose(fp2);
    fp3=fopen("Copyfile2.txt","r");
    while((ch = getchar()) != '@')
    {
        putchar(ch);
    }
    putchar(ch);
    fclose(fp3);
}
```

[ID: 224610386] Page No: 116

Test Case - 1	
User Output	

Enter the text with @ at end :
Copyfile1 started in the year 2016@
Copied text is : Copyfile1 started in the year 2016@

Srinidhi Parameter Institute of Technology [2022-2026-CSE-8] Page No: 117

S.No: 30 Exp Name: Write a C program to Merge two files and stores their contents in another File Date: 2023-06-26

Aim: Write a program to merge two files and stores their contents in another file.

- Open a new file "Copyfile1.txt" in write mode
- Write the content onto the file
- Close the file
- Open a new file "Copyfile2.txt" in write mode
- Write the content onto the file
- Close the file
- Open existing file "Copyfile1.txt" in read mode
- Open a new file "Copyfile2.txt" in write mode
- Copy the content from first existing file to new file
- Open a new file "Copyfile2.txt" in read mode
- Open another existing file "Copyfile2.txt" in read mode
- Copy its content from existing file to new file
- Close the reading file
- Close the writing file

Source Code:

Merge.c

Srinidhi Parameter Institute of Technology [2022-2026-CSE-8] Page No: 118

```

#include<stdio.h>
void main()
{
    FILE *fp1,*fp2;
    char ch;
    fp1=fopen("SampleDatafile1.txt","w");
    printf("Enter the text with @ at end for file1 :\\n");
    while(fgets(ch,100,fp1)) {
        if(ch=='@') {
            putchar(ch,fp1);
        }
        putchar(fp1);
    }
    fclose(fp1);
    fp2=fopen("SampleDatafile2.txt","r");
    fp3=fopen("SampleDatafile3.txt","w");
    while(ch=getchar(fp2)) {
        if(ch=='@') {
            putchar(fp3);
        }
        putchar(fp3);
    }
    fclose(fp2);
    fp3=fopen("SampleDatafile3.txt","r");
    while(fgets(ch,100,fp3)) {
        if(ch=='@') {
            putchar(fp3);
        }
        putchar(fp3);
    }
    fclose(fp3);
    fp1=fopen("SampleDatafile1.txt","r");
    fp2=fopen("SampleDatafile3.txt","r");
    while(fgets(ch,100,fp1)) {
        if(ch=='@') {
            putchar(fp2);
        }
        putchar(fp2);
    }
    fclose(fp1);
    fp1=fopen("SampleDatafile1.txt","w");
    fp2=fopen("SampleDatafile3.txt","r");
    while(fgets(ch,100,fp1)) {
        if(ch=='@') {
            putchar(fp2);
        }
        putchar(fp2);
    }
    fclose(fp1);
    fp2=fopen("SampleDatafile3.txt","r");
    fp3=fopen("SampleDatafile1.txt","w");
    while(ch=getchar(fp2)) {
        if(ch=='@') {
            putchar(fp3);
        }
        putchar(fp3);
    }
    fclose(fp2);
    fp3=fopen("SampleDatafile1.txt","r");
    fp1=fopen("SampleDatafile3.txt","w");
    while(ch=getchar(fp3)) {
        if(ch=='@') {
            putchar(fp1);
        }
        putchar(fp1);
    }
    fclose(fp3);
    fp1=fopen("SampleDatafile1.txt","w");
    fp2=fopen("SampleDatafile3.txt","r");
    while(ch=getchar(fp2)) {
        if(ch=='@') {
            putchar(fp1);
        }
        putchar(fp1);
    }
    fclose(fp2);
    fp1=fopen("SampleDatafile1.txt","w");
    fp2=fopen("SampleDatafile3.txt","r");
    while(ch=getchar(fp2)) {
        if(ch=='@') {
            putchar(fp1);
        }
        putchar(fp1);
    }
    fclose(fp2);
}

```

Execution Results - All test cases have succeeded!

Test Case - 1	
User Output	Enter the text with @ at end for file1 : CodeWarrior developed an interactive tool
Test Case Result	CodeWarrior got best Startup award in 2016 File deleted with @ at end for file1
Source Code	[Copy]

Test Case - 2	
User Output	This is CodeWarrior Given message is : This is CodeWarrior
Test Case Result	Text file is deleted successfully
Source Code	[Copy]

```

#include<stdio.h>
#include<string.h>
int main()
{
    FILE *fp;
    char ch;
    fp=fopen("SampleDatafile1.txt","w");
    printf("Enter the text with @ at end for file1 :\\n");
    while(fgets(ch,100,fp)) {
        if(ch=='@') {
            putchar(ch,fp);
        }
        putchar(fp);
    }
    fclose(fp);
    fp=fopen("SampleDatafile1.txt","r");
    printf("File contents are :\\n");
    while(fgets(ch,100,fp)) {
        if(ch=='@') {
            putchar(ch,fp);
        }
        putchar(fp);
    }
    fclose(fp);
}

```

[ID: 224610386] Page No: 118

```

#include<stdio.h>
#include<string.h>
int main()
{
    FILE *fp;
    char ch;
    fp=fopen("SampleDatafile1.txt","w");
    printf("Enter the text with @ at end for file1 :\\n");
    while(fgets(ch,100,fp)) {
        if(ch=='@') {
            putchar(ch,fp);
        }
        putchar(fp);
    }
    fclose(fp);
    fp=fopen("SampleDatafile1.txt","r");
    printf("File contents are :\\n");
    while(fgets(ch,100,fp)) {
        if(ch=='@') {
            putchar(ch,fp);
        }
        putchar(fp);
    }
    fclose(fp);
}

```

[ID: 224610386] Page No: 119

```

#include<stdio.h>
#include<string.h>
int main()
{
    FILE *fp;
    char ch;
    fp=fopen("SampleDatafile1.txt","w");
    printf("Enter the text with @ at end for file1 :\\n");
    while(fgets(ch,100,fp)) {
        if(ch=='@') {
            putchar(ch,fp);
        }
        putchar(fp);
    }
    fclose(fp);
    fp=fopen("SampleDatafile1.txt","r");
    printf("File contents are :\\n");
    while(fgets(ch,100,fp)) {
        if(ch=='@') {
            putchar(ch,fp);
        }
        putchar(fp);
    }
    fclose(fp);
}

```

[ID: 224610386]

S.No: 31	Exp. Name: Write a C program to Delete a File	Date: 2023-06-26
Aim: Write a program to delete a file.		
Note: Use the fremove() function to delete an existing file.		
Source Code:		
<pre> #include<stdio.h> #include<string.h> void main() { FILE *fp; int status; char filename[20]; printf("Enter the new file name : "); gets(filename); fp=fopen(filename,"w"); puts(fp); fp=fopen(filename,"w"); printf("File deleted with @ at end : "); while(fgets(ch,100,fp)) { if(ch=='@') { putchar(ch,fp); } putchar(fp); } fclose(fp); fp=fopen(filename,"r"); if(fp==NULL) { printf("Given message is : %s"); while(fgets(ch,100,fp)) { if(ch=='@') { putchar(ch,fp); } putchar(fp); } } printf("File deleted successfully\\n",filename); else { printf("unable to delete the file:\\n"); perror("Error\\n"); } } </pre>		

[ID: 224610386] Page No: 120

Execution Results - All test cases have succeeded!

Test Case - 1	
User Output	Enter a new file name : Testfile1
Test Case Result	Testfile1.txt is deleted successfully
Source Code	[Copy]

S.No: 32	Exp. Name: Write a C program to Copy last n characters from one File to another File	Date: 2023-07-03
Aim: Write a program to copy last n characters from file1 to file2. Open file1 in read mode • write the content onto the file • close the file • open a new file "Testfile1.txt" in read mode • open a new file "Testfile2.txt" in write mode • read the number of characters to copy • read the content from file1 into a string • copy the content from existing file to new file • close the files • open a copied file "Testfile1.txt" in read mode • read the text from file and print on the screen • close the file		
Source Code:		
<pre> #include<stdio.h> void main() { FILE *fp1,*fp2; int n,m,length; char ch; fp1=fopen("Testfile1.txt","r"); fp2=fopen("Testfile2.txt","w"); printf("Enter the text with @ at end : "); while(fgets(ch,100,fp1)) { if(ch=='@') { putchar(ch,fp2); } putchar(fp2); } fclose(fp1); fp1=fopen("Testfile1.txt","r"); fp2=fopen("Testfile2.txt","w"); printf("Enter the number of characters to copy : "); scanf("%d",&n); length=fread(fp1,1,n,fp1); fp2=fopen("Testfile2.txt","w"); while(fgets(ch,100,fp1)) { if(ch=='@') { putchar(ch,fp2); } putchar(fp2); } fclose(fp1); fp1=fopen("Testfile2.txt","r"); fp2=fopen("Testfile2.txt","w"); while(fgets(ch,100,fp1)) { if(ch=='@') { putchar(ch,fp2); } putchar(fp2); } fclose(fp2); } </pre>		

[ID: 224610386] Page No: 123

Execution Results - All test cases have succeeded!

Test Case - 1	
User Output	Enter the text with @ at end : We should not allow the file to delete itself and we should not allow the problem to defeat us else, number of characters to copy : 15
Test Case Result	Copied text is : @ or to defeat us
Source Code	[Copy]

S.No: 33	Exp. Name: Write a C program to Reverse first n characters in a file	Date: 2023-07-03
Aim: Write a program to reverse the first n characters in a file. Open a file "Testfile1.txt" in read/write mode • write the content onto the file • read the number of characters to copy • read the content from file into a string • reverse the string • overwrite the entire string into the file from the beginning • open a copied file "Testfile1.txt" in read mode • read the text from file and print on the screen • close the file		
Source Code:		
<pre> #include<string.h> #include<string.h> void stringReverse(char arr[]) { int i,j; for(i=0;i<j;i++) swap(arr[i],arr[j]); } void stringReverse(char data[100]) { int i,j; for(i=0;i<j;i++) swap(data[i],data[j]); for(i=0;i<100;i++) printf("%c",data[i]); } </pre>		

[ID: 224610386] Page No: 125

S.No: 34	Exp. Name: Write a C program to Reverse first n characters in a file	Date: 2023-07-03
Aim: Write a program to reverse the first n characters in a file. Open a file "Testfile1.txt" in read/write mode • write the content onto the file • read the number of characters to copy • read the content from file into a string • reverse the string • overwrite the entire string into the file from the beginning • open a copied file "Testfile1.txt" in read mode • read the text from file and print on the screen • close the file		
Source Code:		
<pre> #include<string.h> #include<string.h> void stringReverse(char arr[]) { int i,j; for(i=0;i<j;i++) swap(arr[i],arr[j]); } void stringReverse(char data[100]) { int i,j; for(i=0;i<j;i++) swap(data[i],data[j]); for(i=0;i<100;i++) printf("%c",data[i]); } </pre>		

[ID: 224610386] Page No: 126


```

D
Original linked list
Nameade (N) -> Veera (V) -> Raj (R) -> Venice (C) -> Ganga (F)
Modified linked list
Nameade (N) -> Venice (C) -> Veer (K) -> Veer (K) -> Raj (R)

```

Test Case - 2

```

User Output
Enter Data:
Enter Name:
Enter Age:
Enter Gender:
M
2
1. Insert into linked list
2. Exit
Enter your option:
1
Enter Name:
Nameade
Enter Age:
14
Enter Gender:
M

3. Insert into linked list
4. Exit
Enter your option:
3
Original linked list
Nameade (N) -> Veera (V) -> Raj (R)
modified linked list
Nameade (N) -> veer (K) -> Raj (R)

Test Case - 3
User Output
Enter Data:
Enter Name:
Age:
Enter Gender:
M
1. Insert into linked list
2. Exit

```

Page No: 15

Srinidhi Samanya Institute of Technology [2022-2026-CSE-B]

Execution Results - All test cases have succeeded!

```

Test Case - 1
User Output
How many numbers do you want to enter?
3
Enter 3 numbers:
1 5 9 12
Enter a number to search:
5
Number found at position:2

Test Case - 2
User Output
How many numbers do you want to enter?
7
Enter 7 numbers:
2 3 6 9 12 20 21
Enter a number to search:
20
Number found at position:5

```

Page No: 15

Srinidhi Samanya Institute of Technology [2022-2026-CSE-B]

```

princ("Invalid symbols in infix expression. Only alphanumeric
and ('+', '-', '*', '/', '^') are allowed.\n");
exit(0);

}
else{
    for(i=0; i<n; i++){
        if(exp[i] == '+'){
            if(exp[i+1] == ')'){
                princ("Unbalanced infix expression : unbalanced parenthesis.\n");
                exit(0);
            }
            else{
                for(j=i+1; j<n; j++){
                    if(exp[j] == '('){
                        princ("Unbalanced infix expression : unbalanced parenthesis.\n");
                        exit(0);
                    }
                    else{
                        for(k=j+1; k<n; k++){
                            if(exp[k] == ')'){
                                princ("Unbalanced infix expression : unbalanced parenthesis.\n");
                                exit(0);
                            }
                            else{
                                if(exp[k] == '+') || exp[k] == '-' || exp[k] == '*' || exp[k] == '/' || exp[k] == '^' ){
                                    princ("Priority mismatch in the expression : \")\n");
                                    exit(0);
                                }
                                else{
                                    for(l=k+1; l<n; l++){
                                        if(exp[l] == '+') || exp[l] == '-' || exp[l] == '*' || exp[l] == '/' || exp[l] == '^' ){
                                            princ("Priority mismatch in the expression : \")\n");
                                            exit(0);
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

Page No: 142

Srinidhi Samanya Institute of Technology [2022-2026-CSE-B]

--	--

--	--

Page No: 17

Srinidhi Samanya Institute of Technology [2022-2026-CSE-B]

--	--

S.No: 37 Exp. Name: Indexing of a file Date: 2023-06-29

Aim: Write a C program to illustrate indexing of a file. Take an array of integers and find whether the given integer is present or not using file indexing method and print the output as shown in the sample output.

Source Code:

```

#include<stdio.h>
#include<limits.h>
#define MAX 25
struct student
{
    int id;
    int roll;
};

int main()
{
    int numbers[MAX];
    struct studentfile index[MAX];
    int i;
    int n;
    int m;
    printf("How many numbers do you want to enter:\n");
    scanf("%d", &n);
    printf("Enter %d numbers:\n", n);
    for(i=0;i<n;i++)
    {
        scanf("%d", &numbers[i]);
    }
    printf("Enter a number to search:\n");
    scanf("%d", &m);
    for(i=0;i<n;i++)
    {
        if(numbers[i] == m)
        {
            printf("Entered number is %d", numbers[i]);
            break;
        }
    }
    if(i==n)
    {
        printf("Number not found.");
        return 0;
    }
    else
    {
        printf("Number found at position:%d", i);
        return 1;
    }
}

```

Srinidhi Samanya Institute of Technology [2022-2026-CSE-B]

Page No: 141

Srinidhi Samanya Institute of Technology [2022-2026-CSE-B]

--	--

Execution Results - All test cases have succeeded!

--	--

Test Case - 1

User Output

Enter the expression :

A+B*C

Postfix expression : ABC+

Test Case - 2

User Output

Enter the expression :

A*B/C

Postfix expression : ABC*

Srinidhi Samanya Institute of Technology [2022-2026-CSE-B]

Page No: 144

--	--

Aim:
Write a C program to convert an Infix expression to Prefix expression.

Source Code:

infixToPrefix.c

```
#include<cs50.h>
#include<string.h>
#include<ctype.h>
#include<stdio.h>
char* strrev(char* str)
{
    char c, *front, *back;
    if(strlen(str)==1||!str)
    {
        return str;
    }
    for(front=0, back=str+strlen(str)-1; front<back; front++, back--)
    {
        c=*(front);
        *(front)=*(back);
        *(back)=c;
    }
    return str;
}
int isDigit(int n)
{
    int top=-1;
    void push(char elem)
    {
        s[++top]=elem;
    }
    char pop()
    {
        if(top<-1)
            return 0;
        else
            return s[top--];
    }
    int prior(char elem)
    {
        switch(elem)
        {
            case '+':
            case '-':
            case '*':
            case '/':
                return 3;
            case '^':
                return 2;
            case ')':
                return 1;
            case '(':
                return 0;
        }
    }
    void main()
    {
        char Inf[10],Pref[50],ch,ele;
        int i=0;
        printf("Enter Infix Expression:");
        scanf("%s",Inf);
        start("a",Inf);
        push("a");
        priStr();
        while((Inf[i]==Inf[i+1])&&i<9)
        {
            i++;
        }
    }
}
```

```
else if(isalnum(ch))
    pref[i++]=ch;
else if(ch=='')
{
    while(s[i]>' ')
        i--;
    prfx[i]=s[i]-pop();
}
else
{
    while(prfx[i]>ch)
        i--;
    prfx[i]=pop();
}
i=i+1;
}
while(s[i]>' ')
{
    i--;
    prfx[i]=s[i];
}
strrev(pref);
strrev(pref);
printf("Prefix Expression:%s\n",pref);
}
```

Execution Results - All test cases have succeeded!

Test Case - 1		
User Output	Enter Infix Expression:	A+B
		Prefix Expression:A+B
Test Case - 2		
User Output	Enter Infix Expression:	A+B+C/D
		Prefix Expression:A+B+C/D

Aim:
Write a C program to convert a Postfix expression to Infix expression.

Source Code:

postfixToInfix.c

```
#include<cs50.h>
#include<string.h>
#include<ctype.h>
#include<stdio.h>
#define MAX 20
char Inf[10],stack[MAX];
int top=-1;
void push(char ch)
{
    stack[++top]=ch;
}
char pop()
{
    return stack[top--];
}
char* strrev(char* str)
{
    char c, *front, *back;
    if(strlen(str)==1||!str)
    {
        return str;
    }
    for(front=0, back=str+strlen(str)-1; front<back; front++, back--)
    {
        c=*(front);
        *(front)=*(back);
        *(back)=c;
    }
    return str;
}
void postfix()
{
    int i,n=0;
    char a,b,c;
    printf("Enter a Postfix expression:");
    scanf("%s",Inf);
    start("a",Inf);
    strrev(Inf);
    maxRev(Inf);
    for(i=0;i<n;i++)
    {
        stack[i]='*#';
    }
    printf("Postfix expression:%s",Inf);
    for(i=0;i<n;i++)
    {
        if(str[i]=='+'||str[i]=='-'||str[i]=='*'||str[i]=='/'||str[i]=='^')
        {
            push(str[i]);
        }
        else
        {
            if(i==0)
            {
                a=pop();
                push('1');
                if(str[i]=='^')
                {
                    printf("%c",a);
                }
                else
                {
                    printf("%c,%c",a,op);
                }
            }
            else
            {
                if(str[i]=='+'||str[i]=='-'||str[i]=='*'||str[i]=='/'||str[i]=='^')
                {
                    push(str[i]);
                }
                else
                {
                    op=pop();
                    push('1');
                    if(str[i]=='^')
                    {
                        printf("%c",a);
                    }
                    else
                    {
                        printf("%c,%c",a,op);
                    }
                }
            }
        }
    }
    if(top==0)
    {
        printf("\n\n",1,top--);
    }
    else
    {
        printf("\n\n",1,top);
    }
}
```

```
{    postfix();
}

Execution Results - All test cases have succeeded!

| Test Case - 1      |                             |                          |
|--------------------|-----------------------------|--------------------------|
| <b>User Output</b> | Enter a Postfix expression: | A+B                      |
|                    |                             | Infix expression:A+B     |
|                    |                             |                          |
| Test Case - 2      |                             |                          |
| <b>User Output</b> | Enter a Postfix expression: | A+B+C/D                  |
|                    |                             | Infix expression:A+B+C/D |


```

Aim:
Write a C program to convert a Prefix expression to Infix expression.

Source Code:

prefixToInfix.c

```
#include<cs50.h>
#include<string.h>
#include<ctype.h>
#include<stdio.h>
#define MAX 20
char Inf[10],stack[MAX];
int top=-1;
void push(char ch)
{
    stack[++top]=ch;
}
char pop()
{
    return stack[top--];
}
void prefix()
{
    int i,n;
    char a,b,c;
    printf("Enter a Prefix expression:");
    scanf("%s",Inf);
    start("a",Inf);
    maxRev(Inf);
    for(i=0;i<n;i++)
    {
        stack[i]='*#';
    }
    printf("Prefix expression:%s",Inf);
    for(i=0;i<n;i++)
    {
        if(str[i]=='+'||str[i]=='-'||str[i]=='*'||str[i]=='/'||str[i]=='^')
        {
            push(str[i]);
        }
        else
        {
            if(i==0)
            {
                a=pop();
                push('1');
                if(str[i]=='^')
                {
                    printf("%c",a);
                }
                else
                {
                    printf("%c,%c",a,op);
                }
            }
            else
            {
                if(str[i]=='+'||str[i]=='-'||str[i]=='*'||str[i]=='/'||str[i]=='^')
                {
                    push(str[i]);
                }
                else
                {
                    op=pop();
                    push('1');
                    if(str[i]=='^')
                    {
                        printf("%c",a);
                    }
                    else
                    {
                        printf("%c,%c",a,op);
                    }
                }
            }
        }
    }
    if(top==0)
    {
        printf("\n\n",1,top--);
    }
    else
    {
        printf("\n\n",1,top);
    }
}
```

```
} void exit()
{
    prefix();
}

Execution Results - All test cases have succeeded!

| Test Case - 1      |                            |                          |
|--------------------|----------------------------|--------------------------|
| <b>User Output</b> | Enter a Prefix expression: | +AB                      |
|                    |                            | Infix expression:A+B     |
|                    |                            |                          |
| Test Case - 2      |                            |                          |
| <b>User Output</b> | Enter a Prefix expression: | /ABC/CD                  |
|                    |                            | Infix expression:A/B/C/D |


```

Aim:
Write a C program to convert a Postfix expression to Prefix expression.

Source Code:

```
postfixToPrefix.c
```

ID: 2246745880 | Page No: 154

Aim:
Write a C program to convert a Prefix expression to Postfix expression.

Source Code:

```
prefixToPostfix.c
```

ID: 2246745880 | Page No: 153

```
#include<stdio.h>
#include<conio.h>
#include<malloc.h>
#include<string.h>
#define MAX 20
char *strcpy(char*,char)
{
    char s[10],*start,*back;
    int i,j,k;
    start=s;
    back=s+strlen(s);
    for(i=0;i<=k;i++)
    {
        s[i]=start[i];
        start++;
    }
    return s;
}
char *strchr(char*,char)
{
    char s[MAX],*back;
    int i,tom;
    void push(char c)
    {
        stack[top]=c;
    }
    char pop()
    {
        return stack[top--];
    }
    void peek()
    {
        int i=top;
        char c[10];
        printf("Enter the postfix expression:");
        scanf("%s",s);
        start="s";
        i=0;
        for(i=0;i<MAX;i++)
        {
            if(s[i]==' ')
            {
                continue;
            }
            stack[i]=s[i];
            printf("Postfix expression is:");
            for(j=i;j>0;j--)
            {
                if(stack[j]==')'||stack[j]=='}'||stack[j]==';'||stack[j]==',')
                {
                    push(stack[i]);
                }
                else
                {
                    if(stack[j]=='+')
                    {
                        while((top>-1)&&(stack[top]=='+'))
                        {
                            A=pop();
                            B=pop();
                            C=(A+B)/2;
                            push(C);
                        }
                    }
                    else if(stack[j]=='-')
                    {
                        while((top>-1)&&(stack[top]=='-'))
                        {
                            A=pop();
                            B=pop();
                            C=(B-A)/2;
                            push(C);
                        }
                    }
                    else if(stack[j]=='*')
                    {
                        while((top>-1)&&(stack[top]=='*'))
                        {
                            A=pop();
                            B=pop();
                            C=A*B;
                            push(C);
                        }
                    }
                    else if(stack[j]=='/')
                    {
                        while((top>-1)&&(stack[top]=='/'))
                        {
                            A=pop();
                            B=pop();
                            C=B/A;
                            push(C);
                        }
                    }
                }
            }
        }
        if(s[i]!=';')
        {
            printf("Postfix expression is: %s\n",stack);
        }
    }
    void main()
    {
        clrscr();
    }
}
```

Syntex Samanya Institute of Technology ID: 2246745880 | Page No: 155

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter the postfix expression:

ABC*D+

Prefix expression is:-ADBC

Test Case - 2

User Output

Enter the postfix expression:

ABCD*

Prefix expression is:-ABCD

Syntex Samanya Institute of Technology ID: 2246745880 | Page No: 156