```
#include<stdio.h>
#include<stdlib.h>
struct node
{
        struct node *next;
        int vertex;

};
typedef struct node * GNODE;
GNODE graph[20];
int visited[20];
int n;
void DFS(int i)
{
        GNODE p;
        printf("\n%d",i);
        p=graph[i];
        visited[i]=1;
        while(p!=NULL)
        {
                i=p->vertex;
                if(!visited[i]) DFS(i); p=p->next;
        }
}
void main()
{
        int N,E,i,s,d,v;
        GNODE q,p;
        printf("Enter the number of vertices : ");
        scanf("%d",&N);
        printf("Enter the number of edges : ");
        scanf("%d",&E);
        for(i=1;i<=E;i++)
        {
                printf("Enter source : ");
                scanf("%d",&s);
                printf("Enter destination : ");
                scanf("%d",&d);
                q=(GNODE)malloc(sizeof(struct node));
                q->vertex=d;
                q->next=NULL;
                if(graph[s]==NULL)
                graph[s]=q;
                else
                {
                        p=graph[s];
                        while(p->next!=NULL)
                        p=p->next;
                        p->next=q;
                }

        }
        for(i=0;i<n;i++)
        visited[i]=0;
        printf("Enter Start Vertex for DFS : ");
```

```
        DFS(v);
        printf("\n");

}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the number of vertices :
6
Enter the number of edges :
7
Enter source :
1
Enter destination :
2
Enter source :
1
Enter destination :
4
Enter source :
4
Enter destination :
2
Enter source :
2
Enter destination :
3
Enter source :
4
Enter destination :
5
Enter source :
1
Enter destination :
3
Enter source :
3
Enter destination :
6
Enter Start Vertex for DFS :
1
DFS of graph :
1
2

3
6
4
5

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

S.No: 27	Exp. Name: ***Travelling Sales Person problem using Dynamic programming***	Date: 2023-08-09

Aim:

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

Source Code:

TSP.c

```
#include<stdio.h>
int ary[10][10], completed[10], n, cost = 0;
void takeInput()
{
         int i, j;
          printf("Number of villages: ");
           scanf("%d", & n);
            for (i = 0; i < n; i++)
             {
                 for (j = 0; j < n; j++)
                  scanf("%d", & ary[i][j]);
                      completed[i] = 0;
             }
              printf("The cost list is:");
               for (i = 0; i < n; i++)
                {
                         printf("\n");
                          for (j = 0; j < n; j++)
                           printf("\t%d", ary[i][j]);
                }
}
void mincost(int city)
{
         int i, ncity;
          completed[city] = 1;
           printf("%d-->", city + 1);
            ncity = least(city);
             if (ncity == 999)
              {
                 ncity = 0;
                  printf("%d", ncity + 1);
                   cost += ary[city][ncity];
                         return;
              }
                 mincost(ncity);
}
int least(int c)
{
         int i, nc = 999;
          int min = 999, kmin;
           for (i = 0; i < n; i++)
                {
                            if ((ary[c][i] != 0) && (completed[i] == 0))
                                 if (ary[c][i] + ary[i][c] < min)
                                    {
                                          min = ary[i][0] + ary[c][i];
                                             kmin = ary[c][i];
                                                nc = i;
                                    }
                }
                     if (min != 999)
                          cost += kmin;
                               return nc;
}
int main()
```

```
        printf("\nThe Path is:\n");
             mincost(0);
              printf("\nMinimum cost is %d", cost);
               return 0;
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Number of villages:
3
0 10 15
10 0 35
15 35 0
The cost list is:
0 10 15
10 0 35
15 35 0
The Path is:
1-->2-->3-->1
Minimum cost is 60

S.No: 28	Exp. Name: ***Write a C program to Open a File and to Print its contents on the screen***	Date: 2023-04-24

ID: 224G1A05B1

Srinivasa Ramanujan Institute of Technology

Aim:

Follow the instructions given below to write a program to `open` a file and to `print` its **contents** on the screen.

- Open a new file "`SampleText1.txt`" in write mode
- Write the content in the file
- Close the file
- Open the same file in read mode
- Read the content from file and print them on the screen
- Close the file

Source Code:

file1.c

```
#include<stdio.h>
void main()
{
FILE *fp;
char ch;
fp=fopen("SampleText1.txt", "w");
printf("Enter the text with @ at end : ");
while((ch =getchar()) != '@')
{
        putc(ch,fp);
}
putc(ch,fp);
fclose(fp);
fp=fopen("SampleText1.txt","r");
printf("Given message is : ");
while((ch=getc(fp)) != '@')
{
        putchar(ch);
}
printf("\n");
fclose(fp);
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
`Enter the text with @ at end :`
CodeTantra is a
Startup Company recognized by Government of India@
`Given message is : CodeTantra is a`
`Startup Company recognized by Government of India`

Test Case - 2
User Output
Enter the text with @ at end :
CodeTantra is
increasing development of Languages Year
by Year@
Given message is : CodeTantra is
increasing development of Languages Year
by Year

S.No: 29	Exp. Name: ***Write a C program to Copy contents of one File into another File***	Date: 2023-06-26

ID: 224G1A05B1

Srinivasa Ramanujan Institute of Technology

Aim:

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

- Open a new file "`SampleTextFile1.txt`" in write mode
- Write the content onto the file
- Close the file
- Open an existing file "`SampleTextFile1.txt`" in read mode
- Open a new file "`SampleTextFile2.txt`" in write mode
- Copy the content from existing file to new file
- Close the files
- Open the copied file in read mode
- Read the text from file and print on the screen
- Close the file

Source Code:

CopyFile.c

```
#include <stdio.h>
void main()
{
        FILE *fp, *fp1, *fp2;char ch;
        fp = fopen("SampleTextFile1.txt", "w");
        printf("Enter the text with @ at end : ");
        while ((ch = getchar()) != '@')
        {
                putc(ch, fp);

        }
        putc(ch, fp);
        fclose(fp);
        fp1 = fopen("SampleTextFile1.txt", "r");
        fp2 = fopen("SampleTextFile2.txt", "w");
        while ((ch = getc(fp1)) != '@')
        {
                putc(ch, fp2);

        }
        putc(ch, fp2);
        fclose(fp1);
        fclose(fp2);
        fp2 = fopen("SampleTextFile2.txt", "r");
        printf("Copied text is : ");
        while ((ch = getc(fp2)) != '@')
        {
                putchar(ch);

        }
        printf("\n");
        fclose(fp2);

}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the text with @ at end :
CodeTantra started in the year 2014@
Copied text is : CodeTantra started in the year 2014

Test Case - 2
User Output
Enter the text with @ at end :
CodeTantra received
best Startup award from Hysea in 2016@
Copied text is : CodeTantra received
best Startup award from Hysea in 2016

ID: 224G1A05B1
2022-2026-CSE-B
Srinivasa Ramanujan Institute of Technology

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 "`SampleDataFile1.txt`" in write mode
- Write the content onto the file
- Close the file
- Open another new file "`SampleDataFile2.txt`" in write mode
- Write the content onto the file
- Close the file
- Open first existing file "`SampleDataFile1.txt`" in read mode
- Open a new file "`SampleDataFile3.txt`" in write mode
- Copy the content from first existing file to new file
- Close the first existing file
- Open another existing file "`SampleDataFile2.txt`" in read mode
- Copy its content from existing file to new file
- Close that existing file
- Close the merged file

Source Code:

`Merge.c`

ID: 224G1A05B1

Srinivasa Ramanujan Institute of Technology

```c
#include <stdio.h>
void main()
{
        FILE *fp1, *fp2, *fp3;
        char ch;
        fp1 = fopen("SampleDataFile1.txt", "w");
        printf("Enter the text with @ at end for file-1 :\n");
        while ((ch = getchar()) != '@')
        {
                putc(ch, fp1);

        }
        putc(ch, fp1);
        fclose(fp1);
        fp2 = fopen("SampleDataFile2.txt", "w");
        printf("Enter the text with @ at end for file-2 :\n");
        while ((ch = getchar()) != '@')
        {
                putc(ch, fp2);

        }
        putc(ch, fp2);
        fclose(fp2);
        fp1 = fopen("SampleDataFile1.txt", "r");
        fp3 = fopen("SampleDataFile3.txt", "w");
        while ((ch = getc(fp1)) != '@')
        {
                putc(ch, fp3);

        }
        fclose(fp1);
        fp2 = fopen("SampleDataFile2.txt", "r");
        while ((ch = getc(fp2)) != '@')
        {
                putc(ch, fp3);

        }
        putc(ch, fp3);
        fclose(fp2);
        fclose(fp3);
        fp3 = fopen("SampleDataFile3.txt", "r");
        printf("Merged text is : ");
        while ((ch = getc(fp3)) != '@')
        {
                putchar(ch);
        }
        printf("\n");
        fclose(fp3);
}
```


Execution Results - All test cases have succeeded!

Test Case - 1

User Output
Enter the text with @ at end for file-1 :
CodeTantra developed an interactive tool
in the year 2014
CodeTantra got best Startup award in 2016@
Enter the text with @ at end for file-2 :
Now lot of Companies and Colleges using CodeTantra Tool@
Merged text is : CodeTantra developed an interactive tool
in the year 2014
CodeTantra got best Startup award in 2016
Now lot of Companies and Colleges using CodeTantra Tool

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 `remove(fileName)` function to delete an existing file.

Source Code:

Delete.c

```
#include <stdio.h>
void main()
{
        FILE *fp;
        int status;
        char fileName[40], ch;
        printf("Enter a new file name : ");
        gets(fileName);
        fp = fopen(fileName, "w");
        printf("Enter the text with @ at end : ");
        while ((ch = getchar()) != '@')
        {
                putc(ch, fp);

        }
        putc(ch, fp);
        fclose(fp);
        fp = fopen(fileName, "r");
        printf("Given message is : ");
        while ((ch = getc(fp)) != '@')
        {
                putchar(ch);

        }
        printf("\n");
        fclose(fp);
        status = remove(fileName);
        if (status == 0)
        printf("%s file is deleted successfully\n", fileName);
        else
        {
                printf("Unable to delete the file -- ");
                perror("Error\n");
        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter a new file name :

Text1.txt
Enter the text with @ at end :
This is CodeTantra@
Given message is : This is CodeTantra
Text1.txt file is deleted successfully

Test Case - 2
User Output
Enter a new file name :
Text2.txt
Enter the text with @ at end :
C developed by Dennis Ritchie@
Given message is : C developed by Dennis Ritchie
Text2.txt file is deleted successfully

S.No: 32	Exp. Name: ***Write a C program to Copy last n characters from one File to another File***	Date: 2023-06-26

Aim:

Write a program to `copy` last `n` characters from **file-1** to **file-2**.

- open a new file "`TestDataFile1.txt`" in write mode
- write the content onto the file
- close the file
- open an existing file "`TestDataFile1.txt`" in read mode
- open a new file "`TestDataFile2.txt`" in write mode
- read the number of characters to copy
- set the cursor position by using fseek()
- copy the content from existing file to new file
- close the files
- open the copied file "`TestDataFile2.txt`" in read mode
- read the text from file and print on the screen
- close the file

Source Code:

`Copy.c`

```c
#include <stdio.h>
void main()
{
        FILE *fp, *fp1, *fp2;
        int num, length;
        char ch;
        fp = fopen("TestDataFile1.txt", "w");
        printf("Enter the text with @ at end : ");
        while ((ch = getchar()) != '@')
        {
                putc(ch, fp);

        }
        putc(ch, fp);
        fclose(fp);
        fp1 = fopen("TestDataFile1.txt", "r");
        fp2 = fopen("TestDataFile2.txt", "w");
        printf("Enter number of characters to copy : ");
        scanf("%d", &num);
        fseek(fp1, 0L, SEEK_END);
        length = ftell(fp1);
        fseek(fp1, (length - num - 1), SEEK_SET);
        while ((ch = getc(fp1)) != '@')
        {
                putc(ch, fp2);

        }
        putc(ch, fp2);
        fclose(fp1);
        fclose(fp2);
        fp2 = fopen("TestDataFile2.txt", "r");
        printf("Copied text is : ");
        while ((ch = getc(fp2)) != '@')
        {
                putchar(ch);
        }
        printf("\n");
        fclose(fp2);
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the text with @ at end :
We should not give up
and we should not allow the problem to defeat us@
Enter number of characters to copy :
15
Copied text is : em to defeat us

Test Case - 2
User Output
Enter the text with @ at end :
You have to dream
before
Your dreams can come true@
Enter number of characters to copy :
20
Copied text is : dreams can come true

S.No: 33	Exp. Name: ***Write a C program to Reverse first n characters in a File***	Date: 2023-06-26

Aim:

Write a program to `reverse` the first `n` characters in a file.

- open a new file "`TestDataFile3.txt`" in read/write mode
- write the content onto the file
- read the number of characters to copy
- copy the specified number of characters into a string
- reverse the string
- overwrite the entire string into the file from the begining
- close the file
- open the copied file "`TestDataFile3.txt`" in read mode
- read the text from file and print on the screen
- close the file

Source Code:

Program1506.c

ID: 224G1A05B1
2022-2026-CSE-B
Srinivasa Ramanujan Institute of Technology

```
#include <stdio.h>
#include <string.h>
void stringReverse(char[]);
void main()
{
        FILE *fp;
        int num, i;
        char ch, data[100];
        fp = fopen("TestDataFile3.txt", "w+");
        printf("Enter the text with @ at end : ");
        while ((ch = getchar()) != '@')
        {
                putc(ch, fp);

        }
        putc(ch, fp);
        printf("Enter number of characters to copy : ");
        scanf("%d", &num);
        i = 0;
        rewind(fp);
        while (i < num)
        {
                data[i] = getc(fp);
                i++;
        }
        data[i] = '\0';
        rewind(fp);
        stringReverse(data);
        fputs(data, fp);
        fclose(fp);
        fp = fopen("TestDataFile3.txt", "r");
        printf("Result is : ");
        while ((ch = getc(fp)) != '@')
        {
                putchar(ch);
        }
        printf("\n");
        fclose(fp);
}
void stringReverse(char data[100])
{
        int i, j;
        char temp;
        i = j = 0;
        while (data[j] != '\0')
        {
                j++;
        }
        j--;
        while (i < j)
        {
                temp = data[i];
                data[i] = data[j];
                data[j] = temp;
                i++;
                j--;
```

```
        }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the text with @ at end :
Teaching is a
very noble profession that shapes the
character, caliber and future of an individual@
Enter number of characters to copy :
18
Result is : yrev
a si gnihcaeT noble profession that shapes the
character, caliber and future of an individual

Test Case - 2
User Output
Enter the text with @ at end :
Small aim
is a crime; have great aim@
Enter number of characters to copy :
11
Result is : i
mia llamSs a crime; have great aim

S.No: 34	Exp. Name: ***Write a C program to Append data to an existing File***	Date: 2023-06-26

Aim:

Write a program to `append` data to an existing file and **display** its contents.

- open a new file "`DemoTextFile1.txt`" in write mode
- write the content onto the file
- close the file
- open a new same file in append mode
- write the content onto the file
- close the file
- open the same file in read mode
- read the text from file and print them on the screen
- close the file

Source Code:

appendDataToFile.c

```
#include <stdio.h>
void main()
{
        FILE *fp;
        char ch;
        fp = fopen("DemoTextFile1.txt", "w");
        printf("Enter the text with @ at end : ");
        while ((ch = getchar()) != '@')
        {
                putc(ch, fp);
        }
        fclose(fp);
        fp = fopen("DemoTextFile1.txt", "a");
        printf("Enter the text to append to a file with @ at end : ");
        while ((ch = getchar()) != '@')
        {
                putc(ch, fp);

        }
        putc(ch, fp);
        fclose(fp);
        fp = fopen("DemoTextFile1.txt", "r");
        printf("File content after appending : ");
        while ((ch = getc(fp)) != '@')
        {
                putchar(ch);
        }
        printf("\n");
        fclose(fp);
}
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output
Enter the text with @ at end :
I am studying@
Enter the text to append to a file with @ at end :
Life skills in University@
File content after appending : I am studying
Life skills in University

Test Case - 2
User Output
Enter the text with @ at end :
CodeTantra
developed@
Enter the text to append to a file with @ at end :
an interactive tool
to learn Programming@
File content after appending : CodeTantra
developed
an interactive tool
to learn Programming

S.No: 35	Exp. Name: ***Write a C program to Count number of Characters, Words and Lines of a given File***	Date: 2023-07-02

ID: 224G1A05B1

2022-2026-CSE-B

Srinivasa Ramanujan Institute of Technology

Aim:

Write a program to `count` number of **characters**, **words** and **lines** of given text file.

- open a new file "`DemoTextFile2.txt`" in write mode
- write the content onto the file
- close the file
- open the same file in read mode
- read the text from file and find the characters, words and lines count
- print the counts of characters, words and lines
- close the file

Source Code:

countCharWordLines.c

```
#include <stdio.h>
void main()
{
        FILE *fp;
        char ch;
        int charCount = 0, wordCount = 0, lineCount = 0;
        fp = fopen("DemoTextFile2.txt", "w");
        printf("Enter the text with @ at end : ");
        while ((ch = getchar()) != '@')
        {
                putc(ch, fp);}putc(ch, fp);
                fclose(fp);
                fp = fopen("DemoTextFile2.txt", "r");
                do
                {
                        if ((ch == ' ') || (ch == '\n') || (ch == '@'))
                        wordCount++;
                        else
                        charCount++;
                        if (ch == '\n' || ch == '@')
                        lineCount++;
                }
                while ((ch = getc(fp)) != '@');
                fclose(fp);
                printf("Total characters : %d\n", charCount);
                printf("Total words : %d\n", wordCount);
                printf("Total lines : %d\n", lineCount);
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
`Enter the text with @ at end :`
Arise! Awake!

and stop not until
the goal is reached@
Total characters : 43
Total words : 10
Total lines : 3

Test Case - 2
User Output
Enter the text with @ at end :
All power is with in you
you can do anything
and everything@
Total characters : 48
Total words : 12
Total lines : 3

S.No: 36	Exp. Name: ***Linked list Female gender first***	Date: 2023-07-02

Aim:

Consider a linked list consisting of name of a person and gender as a node. Arrange the linked list using 'Ladies first' principle. You may create new linked lists if necessary.

Note: Add node at the beginning.

Source Code:

```
rearrangeList.c
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct Node
{
        int data;
        char name[20];
        char gender;
        struct Node *next;
};
void segregateEvenOdd(struct Node **head_ref)
{
        struct Node *end = *head_ref;
        struct Node *prev = NULL;
        struct Node *curr = *head_ref;
        while (end->next != NULL)
        end = end->next;
        struct Node *new_end = end;
        while (curr->data %2 != 0 && curr != end)
        {
                new_end->next = curr;
                curr = curr->next;
                new_end->next->next = NULL;
                new_end = new_end->next;
        }
        if (curr->data%2 == 0)
        {
                *head_ref = curr;
                while (curr != end)
                {
                        if ( (curr->data)%2 == 0 )
                        {
                                prev = curr;
                                curr = curr->next;
                        }
                        else
                        {
                                prev->next = curr->next;
                                curr->next = NULL;
                                new_end->next = curr;
                                new_end = curr;
                                curr = prev->next;
                        }
                }
        }
        else
        prev = curr;
        if (new_end!=end && (end->data)%2 != 0)
        {
                prev->next = end->next;
                end->next = NULL;
                new_end->next = end;
        }
        return;
}
```

```
        struct Node* new_node = (struct Node*) malloc(sizeof(struct Node));
        strcpy(new_node->name, new_name);
        new_node->gender = new_gender;
        if (new_gender == 'F')
        new_node->data = 0;
        else if (new_gender == 'M')
        new_node->data = 1;
        new_node->next = (*head_ref);
        (*head_ref) = new_node;
}
void printList(struct Node *node)
{
        while (node!=NULL)
        {
                printf("%s (%c)", node->name, node->gender);
                node = node->next;
                if (node!=NULL)
                printf(" --> ");
        }
}
int main()
{
        struct Node* head = NULL;
        char name[20];
        char gender;
        int noOfInputs, i;
        int option;
        printf("Insert Data\n");
        do
        {
                printf("Enter Name: ");
                scanf(" %s", name);
                printf("Enter Gender: ");
                scanf(" %c", &gender);
                push(&head, name, gender);
                printf("1 : Insert into Linked List\n");
                printf("0 : Exit\n");
                printf("Enter your option: ");
                scanf(" %d", &option);
        }
        while(option == 1);
        printf("Original Linked list \n");
        printList(head);
        segregateEvenOdd(&head);
        printf("\nModified Linked list \n");
        printList(head);
        printf("\n");
        return 0;
}
```


Execution Results - All test cases have succeeded!

Test Case - 1

User Output
Insert Data
Enter Name:
Ganga
Enter Gender:
F
1 : Insert into Linked List
0 : Exit
Enter your option:
1
Enter Name:
Yamuna
Enter Gender:
F
1 : Insert into Linked List
0 : Exit
Enter your option:
1
Enter Name:
Raj
Enter Gender:
M
1 : Insert into Linked List
0 : Exit
Enter your option:
1
Enter Name:
Veer
Enter Gender:
M
1 : Insert into Linked List
0 : Exit
Enter your option:
1
Enter Name:
Narmada
Enter Gender:
F
1 : Insert into Linked List
0 : Exit
Enter your option:
1
Enter Name:
Amar
Enter Gender:
M
1 : Insert into Linked List
0 : Exit

Enter your option:
0
Original Linked list
Amar (M) --> Narmada (F) --> Veer (M) --> Raj (M) --> Yamuna (F) --> Ganga (F)
Modified Linked list
Narmada (F) --> Yamuna (F) --> Ganga (F) --> Amar (M) --> Veer (M) --> Raj (M)

Test Case - 2
User Output
Insert Data
Enter Name:
Ganga
Enter Gender:
F
1 : Insert into Linked List
0 : Exit
Enter your option:
1
Enter Name:
Yamuna
Enter Gender:
F
1 : Insert into Linked List
0 : Exit
Enter your option:
1
Enter Name:
Narmada
Enter Gender:
F
1 : Insert into Linked List
0 : Exit
Enter your option:
0
Original Linked list
Narmada (F) --> Yamuna (F) --> Ganga (F)
Modified Linked list
Narmada (F) --> Yamuna (F) --> Ganga (F)

Test Case - 3
User Output
Insert Data
Enter Name:
Raj
Enter Gender:
M

1 : Insert into Linked List
0 : Exit
Enter your option:
1
Enter Name:
Veer
Enter Gender:
M
1 : Insert into Linked List
0 : Exit
Enter your option:
1
Enter Name:
Amar
Enter Gender:
M
1 : Insert into Linked List
0 : Exit
Enter your option:
0
Original Linked list
Amar (M) --> Veer (M) --> Raj (M)
Modified Linked list
Amar (M) --> Veer (M) --> Raj (M)

S.No: 37	Exp. Name: ***Indexing of a file***	Date: 2023-07-02

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:

fileIndexing.c

```c
#include <stdio.h>
#define MAX 25
struct indexfile
{
        int indexId;
        int kIndex;
};
int main()
{
        int numbers[MAX];
        struct indexfile index[MAX];
        int i, num, low, high, br = 4;
        int noOfStudents;
        printf("How many numbers do you want to enter:");
        scanf(" %d", &noOfStudents);
        printf("Enter %d numbers:", noOfStudents);
        for (i = 0; i < noOfStudents; i++)
        {
                scanf("%d", &numbers[i]);
        }
        for (i = 0; i < (noOfStudents / 5); i++)
        {
                index[i].indexId = numbers[br];
                index[i].kIndex = br; br = br + 5;
        }
        printf("Enter a number to search:");
        scanf("%d", &num);
        for (i = 0; (i < noOfStudents / 5) && (index[i].indexId <= num); i++);
        if(i != 0)
        low = index[i - 1].kIndex;
        else
        low = 0;
        if(index[i].kIndex != 0 && index[i].kIndex <= noOfStudents)
        high = index[i].kIndex;
        else
        high = noOfStudents;
        for (i = low; i <= high; i++)
        {
                if (num == numbers[i])
                {
                        printf("Number found at position:%d", i);
                        return 0;
                }
        }
        printf("\nNumber not found.");
        return 0;
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
How many numbers do you want to enter:

5
Enter 5 numbers:
1 5 6 9 12
Enter a number to search:
6
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 25
Enter a number to search:
20
Number found at position:5

S.No: 38	Exp. Name: ***Write a C program to Convert an Infix expression into Postfix expression***	Date: 2023-07-02

Aim:

Write a program to convert an `infix` expression into `postfix` expression.

Source Code:

`Infix2PostfixMain.c`

```
#include<stdlib.h>
#include<string.h>
#include<stdio.h>
#include<ctype.h>
#define STACK_MAX_SIZE 20
char stack [STACK_MAX_SIZE];
int top = -1;
//Return 1 if stack is empty else return 0.
int isEmpty()
{
        if(top<0)
        return 1;
        else
        return 0;
}//Push the character into stackvoid
push(char x)
{
        if(top == STACK_MAX_SIZE - 1)
        {
                printf("Stack is overflow.\n");
        } \
        else
        {
                top = top + 1;
                stack[top] = x;
        }
}//pop a character from stackchar
pop()
{
        if(top < 0)
        {
                printf("Stack is underflow : unbalanced parenthesis\n");
                exit(0);
        }
        else return
        stack[top--];
}// Return 0 if char is '('
// Return 1 if char is '+' or '-'
// Return 2 if char is '*' or '/' or '%'
int priority(char x)
{
        if(x == '(')
        return 0;
        if(x == '+' || x == '-')
        return 1;
        if(x == '*' || x == '/' || x == '%')
        return 2;
}
void convertInfix(char * e)
{
        int x;
        int k=0;
        char * p = (char *)malloc(sizeof(char)*strlen(e));
        while(*e != '\0')
        {
```

```
                else if(*e == '(')
                push(*e);
                else if(*e == ')')
                {
                        while(!isEmpty() && (x = pop()) != '(')
                        p[k++]=x;
                }
                else if (*e == '+' || *e == '-' || *e == '*' || *e == '/' || *e == '%')
                {
                        while(priority(stack[top]) >= priority(*e))
                        p[k++]=pop();
                        push(*e);
                }
                else
                {
                        printf("Invalid symbols in infix expression. Only alphanumeric and {
'+', '-','*', '%%', '/' } are allowed.\n");
                        exit(0);
                }
                e++;
        }
        while(top != -1)
        {
                x=pop();
                if(x == '(')
                {
                        printf("Invalid infix expression : unbalanced parenthesis.\n");
                        exit(0);
                }
                p[k++] = x;
        }
        p[k++]='\0';
        printf("Postfix expression : %s\n",p);
}
int main()
{
        char exp[20];
        char *e, x;
        printf("Enter the expression : ");
        scanf("%s",exp);
        e = exp;
        convertInfix(e);
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the expression :
A+B*(C-D)
Postfix expression : ABCD-*+

Test Case - 2
User Output
Enter the expression :
A+B*C
Postfix expression : ABC*+

S.No: 39	Exp. Name: ***Infix to Prefix Conversion***	Date: 2023-07-02

Aim:

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

Source Code:

infixToPrefix.c

```
#define SIZE 50
#include<string.h>
#include <ctype.h>
#include<stdio.h>
char *strrev(char *str)
{
        char c, *front, *back;
        if(!str || !*str)
        {
                return str;
        }
        for(front=str,back=str+strlen(str)-1;front < back;front++,back--)
        {
                c=*front;
                *front=*back;
                *back=c;
        }
        return str;
}
char s[SIZE];
int top = -1;
void push (char elem)
{
        s[++top] = elem;
}
char pop ()
{
        return (s[top--]);
}
int pr (char elem)
{
        switch (elem)
        {
                case '#':
                return 0;
                case ')':
                return 1;
                case '+':
                case '-':
                return 2;
                case '*':
                case '/':
                return 3;
        }
}
void main ()
{
        char infx[50], prfx[50], ch, elem;
        int i = 0, k = 0;
        printf ("Enter Infix Expression:");
        scanf ("%s", infx);
        push ('#');
        strrev (infx);
        while ((ch = infx[i++]) != '\0')
        {
```

```
                prfx[k++] = ch;
                else if (ch == '(')
                {
                        while (s[top] != ')')
                        {
                                prfx[k++] = pop ();
                        }
                        elem = pop ();
                }
                else
                {
                        while (pr (s[top]) >= pr (ch))
                        {
                                prfx[k++] = pop ();
                        }
                        push (ch);
                }
        }
        while (s[top] != '#')
        {
                prfx[k++] = pop ();
        }
        prfx[k] = '\0';
        strrev (prfx);
        strrev (infx);
        printf ("Prefix Expression:%s\n", prfx);
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter Infix Expression:
A+B
Prefix Expression:+AB

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

S.No: 40	Exp. Name: ***Postfix to Infix Conversion***	Date: 2023-07-02

Aim:

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

Source Code:

postfixToInfix.c

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

```
        x[j]=str[top--];
        strrev(x);
        printf("%s\n",x);
}
void main()
{
        postfix();
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter a Postfix expression:
AB+
Infix expression:A+B

Test Case - 2
User Output
Enter a Postfix expression:
ABC*+D+
Infix expression:A+B*C+D

S.No: 41	Exp. Name: ***Prefix to Infix Conversion***	Date: 2023-07-02

Aim:

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

Source Code:

prefixToInfix.c

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

```
    {
        prefix();
    }
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter a Prefix expression:
+AB
Infix expression:A+B

Test Case - 2
User Output
Enter a Prefix expression:
+/AB/CD
Infix expression:A/B+C/D

S.No: 42	Exp. Name: ***Postfix to Prefix Conversion***	Date: 2023-07-02

Aim:

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

Source Code:

postfixToPrefix.c

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
# define MAX 20
char *strrev(char *str)
{
        char c, *front, *back;
        if(!str || !*str)
        return str;
        for(front=str,back=str+strlen(str)-1;
        front < back;front++,back--)
        {
                c=*front;
                *front=*back;
                *back=c;
        }
        return str;
}
char str[MAX],stack[MAX];
int top=-1;
void push(char c)
{
        stack[++top]=c;
}
char pop()
{
        return stack[top--];
}
void post_pre()
{
        int n,i,j=0;
        char c[20];
        char a,b,op;
        printf("Enter the postfix expression:");
        scanf("%s", str);
        n=strlen(str);
        for(i=0;i<MAX;i++)
        stack[i]='\0';
        printf("Prefix expression is:");
        for(i=n-1;i>=0;i--)
        {
                if(str[i]=='+'||str[i]=='-'||str[i]=='*'||str[i]=='/')
                {
                        push(str[i]);
                }
                else
                {
                        c[j++]=str[i];
                        while((top!=-1)&&(stack[top]=='@'))
                        {
                                a=pop();
                                c[j++]=pop();
                        }
                        push('@');
```

```
        c[j]='\0';
        strrev(c);
        printf("%s\n",c);
}
void main()
{
        post_pre();
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the postfix expression:
AB+
Prefix expression is:+AB

Test Case - 2
User Output
Enter the postfix expression:
ABC*+D+
Prefix expression is:++A*BCD

S.No: 43	Exp. Name: ***Prefix to Postfix Conversion***	Date: 2023-07-02

Aim:

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

Source Code:

prefixToPostfix.c

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
# define MAX 20
char str[MAX],stack[MAX];
int top=-1;
void push(char c)
{
        stack[++top]=c;
}
char pop()
{
        return stack[top--];
}
void pre_post()
{
        int n,i,j=0;
        char c[20];
        char a,b,op;
        printf("Enter a Prefix expression:");
        scanf("%s", str);
        n=strlen(str);
        for(i=0;i<MAX;i++)
        stack[i]='\0';
        printf("Postfix expression is:");
        for(i=0;i<n;i++)
        {
                if(str[i]=='+'||str[i]=='-'||str[i]=='*'||str[i]=='/')
                {
                        push(str[i]);
                }
                else
                {
                        c[j++]=str[i];
                        while((top!=-1)&&(stack[top]=='@'))
                        {
                                a=pop();
                                c[j++]=pop();
                        }
                        push('@');
                }
        }
        c[j]='\0';
        printf("%s\n",c);
}
void main()
{
        pre_post();
}
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output
Enter a Prefix expression:
+AB
Postfix expression is:AB+

Test Case - 2
User Output
Enter a Prefix expression:
+/AB/CD
Postfix expression is:AB/CD/+