Department of Computer Science

Gujarat University

Certificate

Roll No: 36   Seat No: _______

This is to certify that Mr./Ms. PREKSHA K. SHETH student of MCA Semester – II has duly completed his/her term work for the semester ending in June 2020, in the subject of ADVANCED PROGRAMMING CONCEPTS towards partial fulfillment of his/her Degree of Masters in Computer Applications.

Date of Submission   Internal Faculty

2nd - JULY - 2020

Head of Department

DEPARTMENT OF COMPUTER SCIENCE
ROLLWALA COMPUTER CENTRE
GUJARAT UNIVERSITY
M.C.A. – II

ROLLNO : 36
NAME : Preksha Sheth
SUBJECT : Advance programming concepts

NO.	TITLE	PAGE NO.	DATE	SIGN
	ASSIGNMENT:- 1	**1 to 21**	**1st july 2020**	
1	**Create a structure name cricket and display the information team wise of a player.**			
2	**Write a program to create a structure of a team i.e baseball team and football team and enter the details regarding it.**			
3	**Write a program to create a structure of bank customer. And perform the functionality.**			
4	**Write a program to maintain the inventory of the books in a bookshop. The details of the book using structure**			
5	**Write a complete 'C' program that will accept the following information for each vehicle using structure**			
	ASSIGNMENT:-2	**21 to 67**	**1st july 2020**	
1	**Write a program using pointers to read array of integers and print its elements in reverse order.**			
2	**Write a program using pointers to find minimum and maximum element of an array and display its address.**			
3	**Write a program to count the number of vowels, consonants, digits and white space characters using pointers**			
4	**WAP using pointers to implement the transpose of a matrix.**			
5	**WAP using pointers to implement the matrix multiplication.**			
6	**WAP to perform summation of a matrix using pointers.**			
7	**Write a program to sort the list of strings using pointers.**			
8	**. Write function that receives a sorted array of integers and**			

Scanned with CamScanner

DEPARTMENT OF COMPUTER SCIENCE
ROLLWALA COMPUTER CENTRE
GUJARAT UNIVERSITY
M.C.A. – II

ROLLNO : 36
NAME : Preksha Sheth
SUBJECT : Advance programming concepts

	an integer value, and inserts the value in correct place			
9.	**Write a function that will round a floating point number to**			
	an indicated decimal place eg: The number 17.457 would			
	yield the value 17.46 when it is rounded off to two decimal			
	places.			
10	**Write a function using pointers to exchange the value stored**			
	in two locations in the memory			
	Perform following through pointers:-			
a.	**Find the first occurrence of a character in the given**			
	string.The function should return the position in the string.			
b.	**Find the first occurrence of a string in another string.The**			
	function should return the position in the string.			
c.	**Delete all occurrences of a character from a string.**			
d.	**Delete all occurrences of a string from another string**			
e.	**Delete all occurrences of a character from a string. Ignore**			
	Case.			
f.	**Delete all occurrences of a string from another string.**			
	Ignore Case			
g.	**Copy one string to another string.**			
h.	**Copy n characters of one string to another string.**			
i.	**Find length of the string and toggle the characters of string**			
j.	**Convert string to all upper case.**			
k.	**Convert string to all lower case.**			
l.	**Sort an array of string**			
m.	**Append one string to another string.**			
n.	**Append at most n characters of one string to another string**			
o.	**Reverse all the characters in the string.**			

Scanned with CamScanner

DEPARTMENT OF COMPUTER SCIENCE
ROLLWALA COMPUTER CENTRE
GUJARAT UNIVERSITY
M.C.A. – II

ROLLNO : 36
NAME : Preksha Sheth
SUBJECT : Advance programming concepts

p.	Compare two strings S1 and S2. The function should return -1, 0 or 1 if S1 < S2, S1 = S2 and S1 > S2 respectively.			
q.	Compare two strings S1 and S2. The function should return -1, 0 or 1 if S1 < S2, S1 = S2 and S1 > S2 respectively.			
r.	Compare at most n characters of two strings S1 and S2. The function should return -1, 0 or 1 if S1 < S2, S1 = S2 and S1 >S2 respectively			
s.	Compare at most n characters of two strings S1 and S2. The function should return -1, 0 or 1 if S1 < S2, S1 =S2 and S1 >S2 respectively.Ignore case.			
	ASSIGNMENT:-3	67 to 110	1st july 2020	
1	Write a program to create a singly linked list and display its elements in FIFO pattern. Display the number of elements.			
2	Write a program to create a singly linked list and display its elements in LIFO pattern. Display the number of elements.			
3	create a singly linked list and perform: i) Insert an element ii)Delete an element iii) Display the list			
4	Write a program to create an ordered linked list.			
5	Write a program to reverse a given linked list.			
6	Write a program to calculate the summation of all elements of the linked list.			
7	Write a program to create two linked list and append the second list after the first.			
8	Write a program to swap two consecutive elements of the given linked list. (Swap only values)			

Scanned with CamScanner

DEPARTMENT OF COMPUTER SCIENCE
ROLLWALA COMPUTER CENTRE
GUJARAT UNIVERSITY
M.C.A. – II

ROLLNO : 36
NAME : Preksha Sheth
SUBJECT : Advance programming concepts

9	**Write a program to swap two consecutive elements of the given linked list. (Swap only addresses)**			
10	**Write a C program to split a given linked list into two.**			
	ASSIGNMENT:-4	**110 to** **147**	**1st july 2020**	
1	**WAP to read line from input file & print alternate character**			
	in the output file. Display message for file i/o errors.			
2	**Write a program to copy the contents of one file to another**			
	and also print the no. of lines in the first file.			
3	**Write a program to search a particular word in an existing**			
	file and display the no. of occurrences and the position of			
	first occurrence of that word. If the word is not found			
	display the appropriate message.			
4	**The files DATA1 and DATA2 contain sorted list of integers.**			
	Write a program to produce a third file DATA which holds			
	a single sorted merged list of these two lists.			
5	**WAP to read line by line from a file and print all repeated**			
	characters on the screen along with their frequency.			
6	**WAF to read a file and count the no. of characters, spaces,**			
	tabs, newlines and no. of words in text file.			
7	**Write a program to remove all the blank lines from a file.**			
8	**Write a function to accept a string from the keyboard and**			
	remove all occurrences of that string from a given file.			
9	**WAP program to remove all the comments from a C file**			
10	**Write a program that will generate a data file containing the**			
	list of customers and their corresponding telephone			
	numbers. Use a structure variable to store the name and			

Scanned with CamScanner

DEPARTMENT OF COMPUTER SCIENCE
ROLLWALA COMPUTER CENTRE
GUJARAT UNIVERSITY
M.C.A. – II

ROLLNO : 36
NAME : Preksha Sheth
SUBJECT : Advance programming concepts

	telephone number of each customer. Create a data file.			
11	**WAP menu driven program that will access the data file**			
	from above program and perform the functionality			
12	**Use a structure of Employee to write records of employee to**			
	a file. Include a menu that will allow the user to select any of			
	the following features			
13	**Write a program that will generate a data file containing the**			
	list of countries and their corresponding capitals. Place the			
	name of each country and its corresponding capital in a			
	separate structure. Treat each structure as a separate record			
14	**Write an interactive, menu-driven C program that will**			
	access the data file generated in the preceding problem and			
	then allow one of the following operations to be executed: a.			
	Determine the capital of a specified country. b. Determine			
	the country whose capital is specified. c.			
15.	**Write a C Program to build utilities for performing**			
	following tasks (Use Command Line Arguments)			
	a. For computing the average of given numbers			
	b. For computing factorial of given numbers			
	c. List all the files in current directory containing word			
	ROLLWALA.			
	d. Rename given file.			
	e. List all EXE files in a given diectory.			
	F . Merge two files into third file.			

Scanned with CamScanner

```
*************************************************************************
*************************************************************************
RollNo      : 36
Name        : Preksha Sheth
Class       : MCA - II
Subject     : Advance Programming
*************************************************************************
*************************************************************************
                             ASSIGNMENT:- 1
*************************************************************************
*************************************************************************
1. Define a structure called cricket that will classify the following
information:-
           a.Player name
           b.Team name
           c.Batting average
      Using   cricket  declare  and   array  player  with   50  elements
and   write  a  program   to  read
      information about the players and print team wise listing
containing names of players with
      their batting average.
      Given a player name write a program to show all the details of the
player.
*************************************************************************
*************************************************************************
#include<stdio.h>
#include<conio.h>
#include<string.h>
struct player
{
      char pname[50];
      float avg;
};
struct teams
{
      char tname[50];
      struct player p[50];
      int pcount;
};
struct teams getdata(int i)
{
      struct teams t1;
      int j;
      printf("\nEnter the name of Team %d: ",i);
      scanf("%s",t1.tname);
      printf("\nHow many players: ");
      scanf("%d",&t1.pcount);
      for(j=0; j<t1.pcount; j++)
      {
            printf("\nEnter the name of Player %d: ",j+1);
            scanf("%s",t1.p[j].pname);
            printf("\nEnter the Batting Average of Player %d: ",j+1);
            scanf("%f",&t1.p[j].avg);
      }
      return t1;
}
void display(struct teams t1[],int n)
{
      int i,j;
      printf("\n\n**************************** Displaying Sorted Data
****************************\n\n");
      for(i=0; i<n; i++)
      {
            printf("\n\nTeam %s\n",t1[i].tname);
            for(j=0; j<t1[i].pcount; j++)
            {
                  printf("\nPlayer %d Name: %s",j+1,t1[i].p[j].pname);
                  printf("\nPlayer %d Batting Average:
%f\n",j+1,t1[i].p[j].avg);
            }
      }
}
void sortdata(struct teams t1[],int n)
{
      struct teams temp;
      int i,j,k;
      for (i = 0; i < n-1; i++)
      {
                  for (j = 0; j < n-i-1; j++)
                  {
                        if(strcmp(t1[j].tname,t1[j+1].tname)>0)
                        {
                              temp=t1[j];
                              t1[j]=t1[j+1];
                              t1[j+1]=temp;
                        }
                  }
      }
      for(k=0; k<n; k++)
      {
            for (i = 0; i < t1[k].pcount-1; i++)
            {
                  for (j = 0; j < t1[k].pcount-i-1; j++)
                  {
                        if(t1[k].p[j].avg<t1[k].p[j+1].avg)
                        {
                              temp.p[i]=t1[k].p[j];
                              t1[k].p[j]=t1[k].p[j+1];
                              t1[k].p[j+1]=temp.p[i];
                        }
                  }
            }
      }
}
void search_data(struct teams t1[],int n)
{
      int i,k;
      char name[50];
      printf("\n\nSearch Player...\n\nEnter the player name to search 
for: ");
      scanf("%s",name);
      for(k=0; k<n; k++)
      {
            for (i = 0; i < t1[k].pcount; i++)
            {
                  if(strcmp(name,t1[k].p[i].pname)==0)
                  {
                        printf("\n\nPlayer Name: %s",t1[k].p[i].pname);
                        printf("\nTeam Name: %s",t1[k].tname);
                        printf("\nBatting Average: %f",t1[k].p[i].avg);
                  }
            }

      }
      printf("\n\n");
}
void main()
{
      int n,i;
      struct teams tm[50];
      printf("How many teams: ");
      scanf("%d",&n);
      for(i=0; i<n; i++)
      {
            tm[i]=getdata(i+1);
      }
      sortdata(tm,n);
      display(tm,n);
      search_data(tm,n);
      getch();
}
*************************************************************************
*************************************************************************
Output:
How many teams: 3

Enter the name of Team 1: India

How many players: 2

Enter the name of Player 1: Dhoni

Enter the Batting Average of Player 1: 86

Enter the name of Player 2: Kohli

Enter the Batting Average of Player 2: 56

Enter the name of Team 2: Australia

How many players: 2
```

```
Enter the name of Player 1: Warner

Enter the Batting Average of Player 1: 78

Enter the name of Player 2: Smith

Enter the Batting Average of Player 2: 59

Enter the name of Team 1: Shrilanka

How many players: 2

Enter the name of Player 1: murlidhar

Enter the Batting Average of Player 1: 66

Enter the name of Player 2: chaminda

Enter the Batting Average of Player 2: 78

**************************** Displaying Sorted Data
****************************



Team Australia

Player 1 Name: Warner
Player 1 Batting Average: 78.000000

Player 3 Name: Smith
Player 3 Batting Average: 59.000000


Team India

Player 1 Name: Dhoni
Player 1 Batting Average: 86.000000

Player 2 Name: Kohli
Player 2 Batting Average: 56.000000

Team shrilanka

Player 1 Name: murlidhar
Player 1 Batting Average: 66.000000

Player 2 Name: chaminda
Player 2 Batting Average: 78.000000

Search Player...
```

```
Enter the player name to search for: Dhoni


Name: Dhoni
Team Name: India
Batting Average: 86.000000

*************************************************************************
*************************************************************************
```

```
2. Write a complete 'c' program that will accept the following
information for each team in
 either football or baseball league:
 a. Team name
 b. City
 c. Number of wins
 For a baseball team, add the following information
       I. Number of hits
       II. Number of runs
      III. No. of errors
      IV. No. of extra-timing games
 Similarly add the following for a football team
      I. No. of ties
      II. No. of field goals
      III. No. of touchdowns
      IV. No. of turn overs
 Enter this information for all the teams in the league then reorder and
print the list of teams
 according to their win-lose records.
```

```c
*************************************************************************
*************************************************************************
#include<conio.h>
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
struct football{
      int ties;
      int field_goals;
      int touch_downs;
      int turn_overs;
};

struct baseball{
      int no_of_hits;
      int runs;
      int errors;
      int extra_timing;
};

union league{
      struct football f1;
      struct baseball b1;
```

```
};

struct team {
      char t_name[30];
      char city[30];
      int no_of_wins;
      int team_code;
      union league l1;

};

struct team get_details(struct team t1){
      printf("Enter team name :\n");
      scanf("%s",t1.t_name);
      printf("Enter city name:\n");
      scanf("%s",t1.city);
      printf("Enter win records:\n");
      scanf("%d",&t1.no_of_wins);
      printf("Enter team code:\n");
      scanf("%d",&t1.team_code);
      if(t1.team_code==1){
            printf("Enter no of hits:\n");
            scanf("%d",&t1.l1.b1.no_of_hits);
            printf("Enter no of runs:\n");
            scanf("%d",&t1.l1.b1.runs);
            printf("Enter no of errors:\n");
            scanf("%d",&t1.l1.b1.errors);
            printf("Enter no of extra timings:\n");
            scanf("%d",&t1.l1.b1.extra_timing);
      }
      else if(t1.team_code==2){
            printf("Enter no. of ties: \n");
            scanf("%d",&t1.l1.f1.ties);
            printf("Enter no. of fiels goals:\n");
            scanf("%d",&t1.l1.f1.field_goals);
            printf("Enter no of touch downs:\n");
            scanf("%d",&t1.l1.f1.touch_downs);
            printf("Enter no. of turn overs:\n");
            scanf("%d",&t1.l1.f1.turn_overs);
      }
      else{
            printf("Enter only 1 or 2 in team_code :\n");
            exit(0);
      }
      return t1;
}

void display_data(struct team t1){
      printf("\n****************************\n");
      printf("\nTeam name is: %s",t1.t_name);
      printf("\nCity name is: %s",t1.city);
      printf("\nNo. of wins record is : %d",t1.no_of_wins);
      if(t1.team_code==1 ){
            printf("\nNo. of hits is : %d",t1.l1.b1.no_of_hits);
```

```
                  printf("\nNo. of runs are: %d",t1.l1.b1.runs);
                  printf("\nNo of errors: %d",t1.l1.b1.errors);
                  printf("\nNo. of extra rimings: %d",t1.l1.b1.extra_timing);
            }
            else {
                  printf("\nNo. of field goal is : %d",t1.l1.f1.field_goals);
                  printf("\nNo. of ties :%d",t1.l1.f1.ties);
                  printf("\nNo. of touchdowns is: %d",t1.l1.f1.touch_downs);
                  printf("\nNo. of turn over is : %d",t1.l1.f1.turn_overs);
            }
}

void sort(struct team t1[],int n){
      int i ,j;
      struct team temp;
      for(i=0;i<n-1;i++){
            for(j=0;j<n-i-1;j++){
                  if(t1[j+1].no_of_wins>t1[j].no_of_wins){
                        temp=t1[j];
                        t1[j]=t1[j+1];
                        t1[j+1]=temp;
                  }
            }
      }
}
int main(){
      int i,n;
      struct team t1[30];
      printf("Enter no. of teams you want to enter : \n");
      scanf("%d",&n);
      for (i=0;i<n;i++){
            t1[i]=get_details(t1[i]);
      }

      printf("\nDisplay Records: \n");
      for (i=0;i<n;i++){
            display_data(t1[i]);
      }

      sort(t1,n);

      printf("\n\nAfter Sorting Records are : \n");

      for (i=0;i<n;i++){
            display_data(t1[i]);
      }
      getch();

}
```

****

****

Output :

```
Enter no. of teams you want to enter :
2
Enter team name :
india
Enter city name:
Ahmedabad
Enter win records:
10
Enter team code:
1
Enter no of hits:
10
Enter no of runs:
60
Enter no of errors:
3
Enter no of extra timings:
5
Enter team name :
australia
Enter city name:
surat
Enter win records:
20
Enter team code:
2
Enter no. of ties:
2
Enter no. of fiels goals:
5
Enter no of touch downs:
3
Enter no. of turn overs:
2

Display Records:

*****************************

Team name is: india
City name is: Ahmedabad
No. of wins record is : 10
No. of hits is : 10
No. of runs are: 60
No of errors: 3
No. of extra rimings: 5
*****************************

Team name is: australia
City name is: surat
No. of wins record is : 20
No. of field goal is : 5
No. of ties :2
No. of touchdowns is: 3
No. of turn over is : 2

After Sorting Records are :

*****************************

Team name is: australia
City name is: surat
No. of wins record is : 20
No. of field goal is : 5
No. of ties :2
No. of touchdowns is: 3
No. of turn over is : 2
*****************************

Team name is: india
City name is: Ahmedabad
No. of wins record is : 10
No. of hits is : 10
No. of runs are: 60
No of errors: 3
No. of extra rimings: 5
```

```
*************************************************************************
*************************************************************************
3.    Write a program that stores and displays the records of the
customer. Thee following information for account
      of the customer is to be stored. Account no, account type, name,
old balance, new balance, last payment, date
        of last payment. Take structure for storing the date in days,
months and year. Also display the current
        account status by comparing current payment and previous balance.
      Also calculate the current balance by  subtracting the current
payment from the previous balance.
*************************************************************************
*************************************************************************
```

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

typedef struct {
      int dd,mm,yy;
}date;

typedef struct {
      int acc_no;
      float old_balance,new_balance,last_payment;
      char name[15];
      char status[10];
      date dateofpay;
}customer;

customer getData();
```

```
void setPrint(customer[],int);
void menudriven(customer[],int);
void main()
{
      customer c[50];
      int i,total_cust;
      printf("Enter Total Number of Customer : ");
      scanf("%d",&total_cust);
      for( i = 0; i < total_cust; i++ )
      {
            c[i] = getData();
      }

      setPrint(c,total_cust);
      menudriven(c,total_cust);
}

customer getData()
{
      customer s;
      printf("\n\nEnter Account Number : ");
      scanf("%d",&s.acc_no);
      printf("\n\nEnter The Customer's Name : ");
      scanf("%s",s.name);
      printf("\n\nEnter The Old Balance : ");
      scanf("%f",&s.old_balance);
      printf("\n\nEnter The Last Payment : ");
      scanf("%f",&s.last_payment);
      s.new_balance = s.old_balance - s.last_payment;
      if(s.new_balance > 0)
      {
            strcpy(s.status,"Pending");
      }
      else{
            strcpy(s.status,"Clear");
      }
      printf("\n\nEnter The Date of Last Payment [dd mm yy] : ");
      scanf("%d %d %d",&s.dateofpay.dd,&s.dateofpay.mm,&s.dateofpay.yy);
      printf("_________________________________________________________")
;
      return s;
}

void setPrint(customer s[],int tc)
{
      int i;
      printf("\n\tList Of
Customers\n_________________________________________________________");
      for( i = 0; i < tc; i++ )
      {
            printf("\n\n\tCustomer Id = %d \n\tCustomer Name = %s\n\tOld
Balance = %.2f\n\tLast Payment = %.2f\n\tLast Payment Date =
%d/%d/%d\n\tNew Balance =
%.2f\n\n_________________________________________________________________
```

```
__",s[i].acc_no,s[i].name,s[i].old_balance,s[i].last_payment,s[i].dateofp
ay.dd,s[i].dateofpay.mm,s[i].dateofpay.yy,s[i].new_balance);
        }
}

void menudriven(customer s[],int tc)
{
        int i,j,option,check;
        char name[15];
        printf("\n\nPress 1 To Display All Customer Details. \n\nPress 2 To
Display Specific Customer By Name. \n\nPress 3 To Display Status of
Customers. \n\nPress 4 To Display Current Balance of Customers. \n\nPress
5 To Exit Program.\n\n");\
        scanf("%d",&option);
        if(option == 1)
        {
                setPrint(s,tc);
                menudriven(s,tc);
        }
        else if(option == 2)
        {
                printf("Enter Customer Name : ");
                scanf("%s",name);
                for( i = 0; i < tc; i++ )
                {
                        check = strcmp(s[i].name,name);
                        if( check == 0 )
                        {
                                printf("\n\n\tCustomer Id = %d \n\tCustomer Name =
%s\n\tOld Balance = %.2f\n\tLast Payment = %.2f\n\tLast Payment Date =
%d/%d/%d\n\tNew Balance = %.2f \n\tStatus = %s
\n\n_______________________________________________",
s[i].acc_no,s[i].name,s[i].old_balance,s[i].last_payment,s[i].dateofpay.d
d,s[i].dateofpay.mm,s[i].dateofpay.yy,s[i].new_balance,s[i].status);
                        }
                }
                menudriven(s,tc);
        }
        else if(option == 3)
        {
                for( i = 0; i < tc; i++ )
                {
                        printf("\nCustomer Name = %s\n Status =
%s\n________________________________________",s[i].name,
s[i].status);
                }
                menudriven(s,tc);
        }
        else if(option == 4)
        {
                for( i = 0; i < tc; i++ )
                {
                 printf("\nCustomer Name = %s\n Current Balance = 
%.2f\n_________________________________________________________",s[i].nam
e,s[i].new_balance);
            }
            menudriven(s,tc);
      }
      else if(option == 5)
      {
            exit(0);
      }
      else
      {
            printf("Please Select Proper Options");
            menudriven(s,tc);
      }
      printf("\nCustomer Does Not Exists.\n");
      menudriven(s,tc);
}

*************************************************************************
*************************************************************************
output :

Enter Total Number of Customer : 2


Enter Account Number : 21196


Enter The Customer's Name : preksha


Enter The Old Balance : 600


Enter The Last Payment : 400


Enter The Date of Last Payment [dd mm yy] : 1  1 2019
_________________________________________________________

Enter Account Number : 211998


Enter The Customer's Name : prerak


Enter The Old Balance : 4000


Enter The Last Payment : 1000


Enter The Date of Last Payment [dd mm yy] : 20 01 2020
```

```
_________________________________________________________


_________________________________________________________
      List Of Customers
_________________________________________________________

        Customer Id = 21196
        Customer Name = preksha
        Old Balance = 600.00
        Last Payment = 400.00
        Last Payment Date = 1/1/2019
        New Balance = 200

___________________________________________________________________

        Customer Id = 21198
        Customer Name = prerak
        Old Balance = 4000.00
        Last Payment = 1000.00
        Last Payment Date = 20/1/2020
        New Balance = 30000
___________________________________________________________________

Press 1 To Display All Customer Details.

Press 2 To Display Specific Customer By Name.

Press 3 To Display Status of Customers.

Press 4 To Display Current Balance of Customers.

Press 5 To Exit Program.

1
        List Of Customers
_________________________________________________________

        Customer Id = 21196
        Customer Name = preksha
        Old Balance = 600.00
        Last Payment = 400.00
        Last Payment Date = 1/1/2019
        New Balance = 200

___________________________________________________________________

        Customer Id = 21198
        Customer Name = prerak
        Old Balance = 4000.00
        Last Payment = 1000.00
        Last Payment Date = 20/1/2020
        New Balance = 30000
```

```
___________________________________________________________________


___________________________________________________________________

Press 1 To Display All Customer Details.

Press 2 To Display Specific Customer By Name.

Press 3 To Display Status of Customers.

Press 4 To Display Current Balance of Customers.

Press 5 To Exit Program.

2

Enter Customer Name : prerak


        Customer Id = 21198
        Customer Name = prerak
        Old Balance = 4000.00
        Last Payment = 1000.00
        Last Payment Date = 20/1/2020
        New Balance = 30000

___________________________________________________________________


Press 1 To Display All Customer Details.

Press 2 To Display Specific Customer By Name.

Press 3 To Display Status of Customers.

Press 4 To Display Current Balance of Customers.

Press 5 To Exit Program.

3

Customer Name = preksha
 Status = Pending
_________________________________________________________
Customer Name = prerak
 Status = Pending
_____________________________________

Press 1 To Display All Customer Details.

Press 2 To Display Specific Customer By Name.
Press 3 To Display Status of Customers.

Press 4 To Display Current Balance of Customers.

Press 5 To Exit Program.

4

Customer Name = preksha
 Current Balance = 200.00
_________________________________________________________
Customer Name = prerak
 Current Balance = 3000.00
_________________________________________________________


Press 1 To Display All Customer Details.

Press 2 To Display Specific Customer By Name.

Press 3 To Display Status of Customers.

Press 4 To Display Current Balance of Customers.

Press 5 To Exit Program.
```

****
****

4. Write a program to maintain the inventory of the books in a bookshop. The details of the book include:

a. Author
b. Title
c. Price
d. Publisher
e. Stock position

Whenever a customer wants a book, the hopkeeper inputs the title and author of the books and the program should reply whether it is available or not by looking through the list of books. If the books is in the list then the system display the books details and ask for no. of copies. If the books is not in the list appropriate message should be displayed. If the no. of copies are available the total cost of the book is displayed otherwise display appropriate message.

****
****

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

```
#include<string.h>
struct book{
      char author_name[20];
      char title[20];
      float price;
      char publisher[20];
      int stock;
};

struct book get_data(struct book b1){
      printf("\n*********************************\n");
      printf("Enter Author Name :\n");
      scanf(" %[^\n]",b1.author_name);
//    gets(b1.author_name);
      printf("Enter Title Of Book :\n");
      scanf(" %[^\n]",b1.title);
      printf("Enter price of book :\n");
      scanf("\n%f",&b1.price);
      printf("Enter Publisher Name :\n");
      scanf(" %[^\n]",b1.publisher);
      printf("Enter Stock of book :\n");
      scanf("\n%d",&b1.stock);
      printf("\n*********************************\n");
      return b1;
}

struct book display_data(struct book b1){
      printf("Author name is : %s:\n",b1.author_name);
      printf("Title is %s :\n",b1.title);
      printf("Price of the book is %f : \n",b1.price);
      printf("Publisher name is %s:\n",b1.publisher);
      //printf("stock is %d : \n",b1.stock);
      printf("\n*********************************\n");
      return b1;
}

void search(struct book b1[],int n){
      char b_title[20],a_name[20],pub[20];
      int i,found=0,copies=0;
      float price,total_price;

      printf("\nEnter The title of the book you want to find : \n");
      scanf(" %[^\n]",b_title);
      printf("\nEnter the author name you want to search :\n");
      scanf(" %[^\n]",a_name);

      for(i=0 ; i<n; i++){
            if(strcmp(b1[i].title,b_title)==0)
            {
                  printf("\nDetails of the book is : \n");
                  display_data(b1[i]);
                  found=1;
                  price = b1[i].price;
            }
      }
      if(found==0){
            printf("Record not fornd: \n");
      }

if(found==1){
      printf("Enter No of copies you want to get :\n");
      scanf("%d",&copies);
      if(copies>=b1[i].stock){
            printf("Stock is available :\n");
            total_price = copies * price;
            printf ("Total Price is : %.2f",total_price);
      }
      else {
            printf("Stock is not available : \n");
      }
}

}

int main(){
      int i,n ;
      struct book b1[20];
      printf("Enter No of books you want to add:\n");
      scanf("%d",&n);

      for(i=0;i<n;i++){
            b1[i]=get_data(b1[i]);
      }
      //for(i=0;i<n;i++){
      //    display_data(b1[i]);
      //}
      search(b1,n);
      getch();
}


*************************************************************************
*************************************************************************
Output :

Enter No of books you want to add:
2

*********************************
Enter Author Name :
Ashok
Enter Title Of Book :
c lang
Enter price of book :
300
Enter Publisher Name :
bpb
Enter Stock of book :
```

```
20

*********************************

*********************************
Enter Author Name :
Bhushan
Enter Title Of Book :
c++
Enter price of book :
200
Enter Publisher Name :
oxfard
Enter Stock of book :
30

*********************************

Enter The title of the book you want to find :
c lang

Enter the author name you want to search :
Ashok

Details of the book is :
Author name is : Ashok:
Title is c lang :
Price of the book is 300.000000 :
Publisher name is bpb:

*********************************
Enter No of copies you want to get :
10
Stock is available :
Total Price is : 3000.00

*************************************************************************
*************************************************************************
```

5 . Write a complete 'C' program that will accept the following information for each vehicle
either two-wheeler and four-wheeler :-

a. Vehicle Name
b. Vehicle Price
c. Vehicle Type [2/4]

For a two-wheeler , add the following information
I. Mileage
II. Type (Geared / Gearless)

Similarly add the following for a four-wheeler
I. Usage(Auto / Manual)
II. Engine Number
III. Type (Heavy / Light)

Enter this information for atleast 10 vehicles

```
*************************************************************************
*************************************************************************

#include<conio.h>
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
struct two_wheelar{
      float milage;
      char type[20];
};
struct four_wheelar{
      char usage[10];
      double engine_no;
      char f_type[10];
};

union v_type{
      struct two_wheelar t1;
      struct four_wheelar f1;
};
struct vehicle {
      char name[30];
      int price;
      char v_type[20];
      union v_type vt1;
};

struct vehicle get_data(struct vehicle v1){\
      printf("\n********************************\n");
      printf("Enter vehicle name:\n");
      scanf("%s",v1.name);
      printf("Enter Vehicle price:\n");
      scanf("%d",&v1.price);
      printf("Enter vehicle type(two-wheelar / four-wheelar): \n");
      scanf("%s",v1.v_type);

      if(strcmp(v1.v_type,"two-wheelar")==0){
            printf("Enter milage: \n");
            scanf("%f",&v1.vt1.t1.milage);
            printf("Enter type(gared / gearless)");
            scanf("%s",v1.vt1.t1.type);
      }
      else if(strcmp(v1.v_type,"four-wheelar")==0){
            printf("Enter the usage of car (auto/manual): \n");
            scanf("%s",v1.vt1.f1.usage);
            printf("Enter the engine no:\n");
            scanf("%ld",&v1.vt1.f1.engine_no);
            printf("Enter the type of the car(heavy/light) : \n");
            scanf("%s",v1.vt1.f1.f_type);
      }
      else{
            printf("Enter proper vehicle type:\n");
            exit(0);
      }
      printf("\n********************************\n");
      return v1;
}

struct vehicle display_data(struct vehicle v1){
      printf("\n********************************\n");
      if(strcmp(v1.v_type,"two-wheelar")==0){
            printf("Vehicle name is: %s\n",v1.name);
            printf("Vehicle price is : %d\n",v1.price);
            printf(" Milage is : %f\n",v1.vt1.t1.milage);
            printf("Vehicle type is: %s\n",v1.vt1.t1.type);
      }
      else if(strcmp(v1.v_type,"four-wheelar")==0){
                  printf("Vehicle type is : %s\n",v1.v_type);
                  printf("Vehicle name is: %s\n",v1.name);
                  printf("Vehicle price is : %d\n",v1.price);
                  printf("The usage of car is : %s\n",v1.vt1.f1.usage);
                  printf("The engine no is : %d\n",v1.vt1.f1.engine_no);
                  printf("The type of the car is %s:
\n",v1.vt1.f1.f_type);
      }

int main(){
      int i ,n ;
      struct vehicle v1[20];
      printf("Enter how many records you want to enter :\n");
      scanf("%d",&n);

      for(i=0;i<n;i++){
            v1[i]=get_data(v1[i]);
      }

      for(i=0;i<n;i++){
            display_data(v1[i]);
      }
}
```

```
*************************************************************************
*************************************************************************
Output:

Enter how many records you want to enter :
2

********************************
Enter vehicle name:
verna
Enter Vehicle price:
400000
Enter vehicle type(two-wheelar / four-wheelar):
four-wheelar
Enter the usage of car (auto/manual):
```

```
auto
Enter the engine no:
123456789
Enter the type of the car(heavy/light) :
heavy

********************************

********************************
Enter vehicle name:
shine
Enter Vehicle price:
60000
Enter vehicle type(two-wheelar / four-wheelar):
two-wheelar
Enter milage:
60
Enter type(gared / gearless)gared

********************************

********************************
Vehicle name is: verna
Vehicle price is : 400000
Vehicle type is : four-wheelar
The usage of car is : auto
The engine no is : 123456789
The type of the car is heavy:

********************************
Vehicle name is: shine
Vehicle price is : 60000
Vehicle type is : two-wheelar
 Milage is : 60.000000
Vehicle type is: gared

--------------------------------
Process exited after 103.8 seconds with return value 2
Press any key to continue . . .
```

****
****

ASSIGNMENT - 2

****
****

1. Write a program using pointers to read array of integers and print its elements in reverse order.

****
****

```
#include<stdio.h>
#include<conio.h>
void read_data(int n,int arr[])
```

```
{
      int i,*iptr;
      iptr=arr;
      for(i=0;i<n;i++)
      {
            printf("Number %d : ",i+1);
            scanf("%d",iptr);
            iptr++;
      }
}
void print_data(int n,int arr[])
{
      int i,*iptr;
      iptr=arr;
      for(i=0;i<n;i++)
      {
            printf("\nNumber %d = %d",i+1,*iptr);
            iptr++;
      }
}
void reverse(int n,int arr[])
{
      int i,*iptr;
      iptr=arr+n-1;
      for(i=n;i>0;i--)
      {
            printf("\nNumber %d = %d",n-i+1,*iptr);
            iptr--;
      }
}

void main()
{
      int n;
      int arr[10];
      printf("How Many Numbers you want to Enter :");
      scanf("%d",&n);
      read_data(n,arr);
      printf("\n==========================================\n");
      printf("\t\t***BEFORE REVERSE***");
      printf("\n==========================================\n");
      print_data(n,arr);
      printf("\n==========================================\n");
      printf("\t\t***AFTER REVERSE***");
      printf("\n==========================================\n");
      reverse(n,arr);
      //print_data(n,arr);
}

*************************************************************************
*************************************************************************
output:

How Many Numbers you want to Enter :5
```

```
Number 1 : 87
Number 2 : 12
Number 3 : 54
Number 4 : 67
Number 5 : 33

==========================================
                ***BEFORE REVERSE***
==========================================

Number 1 = 87
Number 2 = 12
Number 3 = 54
Number 4 = 67
Number 5 = 33
==========================================
                ***AFTER REVERSE***
==========================================

Number 1 = 33
Number 2 = 67
Number 3 = 54
Number 4 = 12
Number 5 = 87Press any key to continue . . .
```

```
*************************************************************************
*************************************************************************
2. Write a program using pointers to find minimum and maximum element of
an array and display it
along with the address at which it is located.

*************************************************************************
*************************************************************************
```

```
#include<stdio.h>
#include<conio.h>
void input_data2(int n,int arr[])
{
      int i;
      int *iptr;
      iptr=arr;
      for(i=0;i<n;i++)
      {
            printf("Enter Number %d : ",i+1);
            scanf("%d",iptr);
            iptr++;
      }
}
void disp_data2(int n,int arr[])
{
      int i;
      int *iptr;
      iptr=arr;
      for(i=0;i<n;i++)
      {
            printf("\nNumber %d = %d",i+1,*iptr);
            iptr++;
      }
}
void min(int n,int arr[])
{
      int i,*iptr,min,loc=1;
      iptr=arr;
      min=*iptr;
      for(i=0;i<n;i++)
      {
            if(min>*iptr)
            {
                  min = *iptr;
                  loc=i+1;
            }
            iptr++;
      }
      printf("\n\nMinimum value %d is present at position
%d..\n",min,loc);
}
void max(int n,int arr[])
{
      int i,*iptr,max,loc=1;
      iptr=arr;
      max=*iptr;
      for(i=0;i<n;i++)
      {
            if(max<*iptr)
            {
                  max = *iptr;
                  loc=i+1;
            }
            iptr++;
      }
      printf("\n\nMaximum value %d is present at position
%d..\n",max,loc);
}


void main()
{
      int arr[30];
      int n;
      printf("How Many Numbers You want to Enter :");
      scanf("%d",&n);
      input_data2(n,arr);
      disp_data2(n,arr);
      min(n,arr);
      max(n,arr);
}
```

```
*************************************************************************
*************************************************************************
output:

How Many Numbers You want to Enter :5
Enter Number 1 : 90
Enter Number 2 : 54
Enter Number 3 : 34
Enter Number 4 : 21
Enter Number 5 : 67

Number 1 = 90
Number 2 = 54
Number 3 = 34
Number 4 = 21
Number 5 = 67

Minimum value 21 is present at position 4..


Maximum value 90 is present at position 1..
Press any key to continue . . .
*************************************************************************
*************************************************************************
```

3. Write a program to count the number of vowels, consonants, digits and white space characters using pointers.

```
*************************************************************************
*************************************************************************
#include<stdio.h>
#include<conio.h>
void get_str(char str[])
{
      printf("Enter String :");
      gets(str);
}
void display2_3(char *ptr)
{
      while(*ptr != '\0')
      {
            //printf("fs");
            printf("%c",*ptr);
            //puts(*ptr);
            ptr++;
      }
}
void count(char *ptr)
{
      int vowels = 0, constants = 0, digit = 0, space = 0;
      while(*ptr != '\0')
      {
            if(*ptr == 'a' || *ptr == 'e' || *ptr == 'i' || *ptr == 'o'
|| *ptr == 'u' ||
               *ptr == 'A' || *ptr == 'E' || *ptr == 'I' || *ptr == 'O'
|| *ptr == 'U')
            {
                  vowels++;
            }
            else if((*ptr>='a' && *ptr<='z') ||(*ptr>='A' && *ptr<='Z'))
            {
                  constants++;
            }
            else if(*ptr >= '0' && *ptr <= '9')
            {
                  digit++;
            }
            else
            {
                  space++;
            }
            ptr++;
      }
      printf("\nNumber of Vowels = %d",vowels);
      printf("\nNumber of Constants = %d",constants);
      printf("\nNumber of Digits = %d",digit);
      printf("\nNumber of White spaces = %d",space);
}

void main()
{
      char str[50];
      char *ptr;
      get_str(str);
      ptr = str;
      display2_3(ptr);
      printf("\n==========================================================
======\n");
      printf("\n\t Count No. of vowels , Constants , Digits & Spaces
\n");
      printf("\n==========================================================
======\n");
      count(ptr);
}

**************************************************************************
**************************************************************************
output:

Enter String :preksha sheth123
preksha sheth123
================================================================

         Count No. of vowels , Constants , Digits & Spaces

================================================================

Number of Vowels = 3
Number of Constants = 9
Number of Digits = 3
Number of White spaces = 1Press any key to continue . . .
```

```
*************************************************************************
*************************************************************************
```

4. Write a program using pointers to implement the transpose of a matrix.

```
*************************************************************************
*************************************************************************
#include<stdio.h>
#include<conio.h>
void inputdata(int (*ptr)[50],int r ,int c)
{
      static int mat=1;
      int i=0,j=0;
      printf("Enter values for Matrix %d",mat++);
      for(i = 0; i < r; i++)
      {
            for(j = 0; j < c; j++)
            {
                  printf("\nEnter matrix[%d][%d] : ",i,j);
                  scanf("%d",(*(ptr+i)+j));
            }
            printf("\n");
      }
}

void disp4 (int (*ptr)[50],int r,int c)
{
      static int mat=1;
      int i=0, j=0;
      printf("\n\t\tMatrix %d\n\n",mat++);
      for(i = 0; i < r; i++)
      {
            for(j = 0; j < c; j++)
            {
                  printf("\t%d",*(*(ptr+i)+j));
            }
            printf("\n\n");
      }
}

void transpose(int (*ptr)[50], int r, int c, int (*ptr2)[50])
{
      int trans,i,j;
      for(i = 0; i <= r; i++)
      {
            for(j = 0; j <= c; j++)
            {
                  trans = (*(*(ptr+j)+i));
                  (*(*(ptr2+i)+j)) = trans;
            }
      }
}

void main()
{
      int m1[50][50], ans[50][50], r,c;
      int (*ptr)[50], (*ptr2)[50];
      ptr = m1;
      ptr2 = ans;
      printf("Enter Rows :");
      scanf("%d",&r);
      printf("Enter Columns :");
      scanf("%d",&c);
      printf("\n=========================================================
=================\n");
      printf("\t\tInput Data for Matrix:");
      printf("\n=========================================================
=================\n");
      inputdata(ptr,r,c);
      printf("\n=========================================================
=================\n");
      printf("\t\tDisplay Matrix :");
      printf("\n=========================================================
=================\n");
      disp4(ptr,r,c);
      printf("\n=========================================================
=================\n");
      printf("\t\tTranspose of Matrix :");
      printf("\n=========================================================
=================\n");
      transpose(ptr,r,c,ptr2);
      disp4(ptr2,c,r);
}


*************************************************************************
*************************************************************************
output:

Enter Rows :2
Enter Columns :3

=========================================================================
=
                Input Data for Matrix:
=========================================================================
=
Enter values for Matrix 1
Enter matrix[0][0] : 1

Enter matrix[0][1] : 2

Enter matrix[0][2] : 3
Enter matrix[1][0] : 4

Enter matrix[1][1] : 5

Enter matrix[1][2] : 6

=========================================================================
=
                Display Matrix :
=========================================================================
=

                Matrix 1

        1       2       3

        4       5       6

=========================================================================
=
                Transpose of Matrix :
=========================================================================
=

                Matrix 2

        1       4

        2       5

        3       6

Press any key to continue . . .
```

****
****
5. Write a program using pointers to implement the matrix multiplication.

****
****

```
#include<stdio.h>
#include<conio.h>
void inputdata5(int (*ptr)[50],int r ,int c)
{
      static int mat=1;
      int i=0,j=0;
      printf("Enter values for Matrix %d",mat++);
      for(i = 0; i < r; i++)
      {
            for(j = 0; j < c; j++)
            {
                 printf("\nEnter matrix[%d][%d] : ",i,j);
```

```
                  scanf("%d",(*(ptr+i)+j));
            }
            printf("\n");
      }
}

void disp5 (int (*ptr)[50],int r,int c)
{
      static int mat=1;
      int i=0, j=0;
      printf("\n\t\tMatrix %d\n\n",mat++);
      for(i = 0; i < r; i++)
      {
            for(j = 0; j < c; j++)
            {
                  printf("\t%d",*(*(ptr+i)+j));
            }
            printf("\n\n");
      }
}

void mul(int (*ptr)[50], int r, int c, int (*ptr2)[50],int (*ptr3)[50])
{
      int i,j;
      for(i = 0; i <= r; i++)
      {
            for(j = 0; j <= c; j++)
            {
                  *(*(ptr3+i)+j) = *(*(ptr+i)+j) * *(*(ptr2+i)+j);
            }
      }
}

void main()
{
      int m1[50][50],m2[50][50], ans[50][50], r,c;
      int (*ptr)[50], (*ptr2)[50],(*ptr3)[50];
      ptr = m1;
      ptr2 = m2;
      ptr3 = ans;
      printf("Enter Rows for matrix 1 & matrix 2:");
      scanf("%d",&r);
      printf("Enter Columns for matrix 1 & matrix 2:");
      scanf("%d",&c);
      printf("\n==========================================================
=================\n");
      printf("\t\tInput Data for Matrix:");
      printf("\n==========================================================
=================\n");
      inputdata5(ptr,r,c);
      //ptr2 = m2;
      inputdata5(ptr2,r,c);
      printf("\n==========================================================
=================\n");
```

```
      printf("\t\tDisplay Matrix :");
      printf("\n=========================================================
=================\n");
      disp5(ptr,r,c);
      disp5(ptr2,r,c);
      printf("\n=========================================================
=================\n");
      printf("\t\tAddition of Matrix :");
      printf("\n=========================================================
=================\n");
      mul(ptr,r,c,ptr2,ptr3);
      disp5(ptr3,r,c);
}

*************************************************************************
*************************************************************************
output:

Enter Rows for matrix 1 & matrix 2:3
Enter Columns for matrix 1 & matrix 2:2

=========================================================================
=
                Input Data for Matrix:
=========================================================================
=
Enter values for Matrix 1
Enter matrix[0][0] : 1

Enter matrix[0][1] : 1


Enter matrix[1][0] : 2

Enter matrix[1][1] : 2


Enter matrix[2][0] : 3

Enter matrix[2][1] : 3

Enter values for Matrix 2
Enter matrix[0][0] : 4

Enter matrix[0][1] : 4


Enter matrix[1][0] : 3

Enter matrix[1][1] : 3


Enter matrix[2][0] : 2
Enter matrix[2][1] : 2

=========================================================================
=
                Display Matrix :
=========================================================================
=

                Matrix 1

        1       1

        2       2

        3       3

                Matrix 2

        4       4

        3       3

        2       2

=========================================================================
=
                Addition of Matrix :
=========================================================================
=

                Matrix 3

        4       4

        6       6

        6       6

Press any key to continue . . .
```

****
****
6. Write a program to perform summation of a matrix using pointers.

****
****

```
#include<stdio.h>
#include<conio.h>
void inputdata6(int (*ptr)[50],int r ,int c)
{
      static int mat=1;
      int i=0,j=0;
      printf("Enter values for Matrix %d",mat++);
      for(i = 0; i < r; i++)
      {
            for(j = 0; j < c; j++)
            {
                  printf("\nEnter matrix[%d][%d] : ",i,j);
                  scanf("%d",(*(ptr+i)+j));
            }
            printf("\n");
      }
}

void disp6 (int (*ptr)[50],int r,int c)
{
      static int mat=1;
      int i=0, j=0;
      printf("\n\t\tMatrix %d\n\n",mat++);
      for(i = 0; i < r; i++)
      {
            for(j = 0; j < c; j++)
            {
                  printf("\t%d",*(*(ptr+i)+j));
            }
            printf("\n\n");
      }
}

void add(int (*ptr)[50], int r, int c, int (*ptr2)[50],int (*ptr3)[50])
{
      int i,j;
      for(i = 0; i <= r; i++)
      {
            for(j = 0; j <= c; j++)
            {
                  *(*(ptr3+i)+j) = *(*(ptr+i)+j) + *(*(ptr2+i)+j);
            }
      }
}

void main()
{
      int m1[50][50],m2[50][50], ans[50][50], r,c;
      int (*ptr)[50], (*ptr2)[50],(*ptr3)[50];
      ptr = m1;
      ptr2 = m2;
      ptr3 = ans;
      printf("Enter Rows for matrix 1 & matrix 2:");
      scanf("%d",&r);
      printf("Enter Columns for matrix 1 & matrix 2:");
      scanf("%d",&c);
      printf("\n==========================================================
=================\n");
      printf("\t\tInput Data for Matrix:");
      printf("\n==========================================================
=================\n");
      inputdata6(ptr,r,c);
      //ptr2 = m2;
      inputdata6(ptr2,r,c);
      printf("\n==========================================================
=================\n");
      printf("\t\tDisplay Matrix :");
      printf("\n==========================================================
=================\n");
      disp6(ptr,r,c);
      disp6(ptr2,r,c);
      printf("\n==========================================================
=================\n");
      printf("\t\tAddition of Matrix :");
      printf("\n==========================================================
=================\n");
      add(ptr,r,c,ptr2,ptr3);
      disp6(ptr3,r,c);
}
```

```
**************************************************************************
**************************************************************************
output:

Enter Rows for matrix 1 & matrix 2:2
Enter Columns for matrix 1 & matrix 2:3

==========================================================================
=
                Input Data for Matrix:
==========================================================================
=
Enter values for Matrix 1
Enter matrix[0][0] : 1

Enter matrix[0][1] : 1

Enter matrix[0][2] : 1


Enter matrix[1][0] : 1

Enter matrix[1][1] : 1

Enter matrix[1][2] : 1

Enter values for Matrix 2
Enter matrix[0][0] : 2

Enter matrix[0][1] : 2

Enter matrix[0][2] : 2
Enter matrix[1][0] : 2

Enter matrix[1][1] : 2

Enter matrix[1][2] : 2

=========================================================================
=
                Display Matrix :
=========================================================================
=

                Matrix 1

        1       1       1

        1       1       1


                Matrix 2

        2       2       2

        2       2       2


=========================================================================
=
                Addition of Matrix :
=========================================================================
=

                Matrix 3

        3       3       3

        3       3       3

Press any key to continue . . .
```

```
*************************************************************************
*************************************************************************
```

7. Write a program to sort the list of strings using pointers.

```
*************************************************************************
*************************************************************************
```

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

void readnames(char[][20],int);
void sort(char[][20],int);
void print(char[][20],int);

void main()
{
      char names[5][20];
      int limit;
      printf("How many names you have: ");
      scanf("%d",&limit);
      readnames(names,limit);
      sort(names,limit);
      print(names,limit);
      getch();
}

void readnames(char names[][20],int limit)
{
      char (*cptr)[20];
      int i;
      cptr=names;
      fflush(stdin);
      printf("Enter %d names:\n",limit);
      for(i=0;i<limit;i++)
      {
            gets(*(cptr+i));
      }
}

void sort(char names[][20],int limit)
{
      char (*cptr)[20],min[20];
      int i,index,j;

      cptr=names;

      for(i=0;i<limit-1;i++)
      {
            strcpy(min,*(cptr+i));
            index=i;

            for(j=i+1;j<limit;j++)
            {
                  if(strcmp(min,*(cptr+j))>0)
                  {
                        strcpy(min,*(cptr+j));
                        index=j;
                  }

            }
            strcpy(*(cptr+index),*(cptr+i));
            strcpy(*(cptr+i),min);
      }

}
void print(char names[][20],int limit)
```

```
{
      char (*cptr)[20];
      int i;
      cptr=names;
      printf("Sorted names:\n");
      for(i=0;i<limit;i++)
      {
            puts(*(cptr+i));
      }
}


*************************************************************************
*************************************************************************
output:

How many names you have: 3

Enter 3 names:
bombay
mumbai
ahmedabad

Sorted names:

ahmedabad
bombay
mumbai


*************************************************************************
*************************************************************************
```

8. Write function that receives a sorted array of integers and an integer value, and inserts the value in correct place.

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

void readnum(int[],int,int*);
void addnum(int[],int*,int*);
void display(int[],int);

void main()
{
      int num[10],limit,n=0;
      printf("Enter how many numbers are in your array: ");
      scanf("%d",&limit);
      readnum(num,limit,&n);
      addnum(num,&limit,&n);
      display(num,limit);
      getch();
}
```

```
void readnum(int num[],int limit,int *new_num)
      //passing the number which user want ot add in a pointer so
{                                                        //it can be used
in other function without returning from here..
      int i,*iptr;
      iptr=num;
      printf("Enter %d sorted numbers: ",limit);

      for(i=0;i<limit;i++)
      {
            scanf("%d",iptr);
            iptr++;
      }
            printf("Enter number to add in your sorted array: ");
            scanf("%d",new_num);

}

void addnum(int num[],int *new_limit,int *new_num)
      //Again passing address of limit and new number..
{
      int *iptr,i,j,flag=0;
      iptr=num;
      for(i=0;i<*new_limit;i++)
      {
            if(*new_num < *(iptr+i))                              //finding
the greater number than our number so we can add new number in the sorted
array
            {
                  for(j=*(new_limit)-1;j>=i;j--)
                  {
                        *(iptr+j+1)=*(iptr+j);
                  }
                  *(iptr+i)=*new_num;
                  (*new_limit)++;
      //increasing limit by 1
                  flag=1;
                  break;
            }
      }
      if(!flag)                                                //if flag is off
that means new number is gretest among sorted array..
      {
            *(iptr+i)=*new_num;
            *new_limit=*new_limit+1;
      }
}

void display(int num[],int limit)
{
      int i,*iptr;

      iptr=num;
      printf("New Array: \n");
      for(i=0;i<limit;i++)
      {
            printf("%d ",*iptr);
            iptr++;
      }
}
```

```
*************************************************************************
*************************************************************************
output:

Enter how many numbers are in your array: 5
Enter 5 sorted numbers: 2 3 5 9 10
Enter number to add in your sorted array: 4
New Array:
2 3 4 5 9 10
```

```
*************************************************************************
*************************************************************************
```

9. Write a function that will round a floating point number to an indicated decimal place eg: The number 17.457 would yield the
    value 17.46 when it is rounded off to two decimal places

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

void round(char[],int);

void main()
{
      char num[10];
      int digits;
      printf("Enter number: ");
      gets(num);
      printf("Enter number of decimal places you want: ");
      scanf("%d",&digits);
      round(num,digits);
      printf("Rounded off number: ");
      puts(num);
      getch();
}


void round(char num[],int digits)
{
      int decimal=0,i=0;
      char *cptr;
      cptr=num;
      while(*cptr != '.')
      {
            i++;
            cptr++;
      }
      if(*(cptr+digits+1)>52 )
      {
            (*(cptr+digits))++;
      }
      *(cptr+digits+1)='\0';
}
```

```
*************************************************************************
*************************************************************************
output:
Enter number: 23.259
Enter number of decimal places you want: 2
Rounded off number: 23.26
```

```
*************************************************************************
*************************************************************************
```

10. Write a function using pointers to exchange the value stored in two locations in the memory.

```
*************************************************************************
*************************************************************************
#include<stdio.h>
#include<conio.h>
void input_data10(int *a,int *b)
{
      int *ptra,*ptrb;
      ptra=a;
      ptrb=b;
      printf("Enter Number 1 : ");
      scanf("%d",ptra);
      printf("Enter Number 2 : ");
      scanf("%d",ptrb);
}
void disp_data10(int a,int b)
{
      printf("\nA = %d",a);
      printf("\nB = %d\n",b);
}
void swap(int *a,int *b)
{
      int *ptra,*ptrb;
      int temp;
      ptra=a;
      ptrb=b;
      temp=*ptra;
      *ptra=*ptrb;
      *ptrb=temp;

}
```

```
void main()
{
      int a=0,b=0;
      input_data10(&a,&b);
      printf("\n==========================================\n");
      printf("\t***BEFORE SWAP***");
      printf("\n==========================================\n");
      disp_data10(a,b);
      swap(&a,&b);
      printf("\n==========================================\n");
      printf("\t***AFTER SWAP***");
      printf("\n==========================================\n");
      disp_data10(a,b);
}
```

****
****
output:

```
Enter Number 1 : 34
Enter Number 2 : 56

==========================================
        ***BEFORE SWAP***
==========================================

A = 34
B = 56

==========================================
        ***AFTER SWAP***
==========================================

A = 56
B = 34
Press any key to continue . . .
```

****
****
11_a. Find the first occurrence of a character in the given string. The function should return the
position in the string.

****
****

```
#include<stdio.h>
#include<conio.h>
void disp_str(char *ptr)
{
      //char *ptr;
      //ptr=str;
      printf("String = ");
      while(*ptr != '\0')
      {
            printf("%c",*ptr);
            ptr++;
      }
}

int search_char(char *ptr, char ch)
{
      int i,val = -1;
      //char *ptr;
      //ptr = str;
      for(i = 0 ; *ptr != '\0' ; i++)
      {
            if(*ptr == ch)
            {
                  val = i;
                  break;
            }
            ptr++;
      }
      return val;
}
void main()
{
      char str[50],ch,*ptr;
      int search_val;
      printf("Enter String :");
      gets(str);
      ptr = str;
      //printf("String= %s",str);
      disp_str(ptr);
      printf("\n \Which Character You Find :");
      scanf("%c",&ch);
      search_val = search_char(ptr,ch);
      if (search_val == -1)
      {
            printf("\nCharacter %c is not found..",ch);
      }
      else
      {
            printf("Character %c is found at %d position
..",ch,search_val);
      }
}

*************************************************************************
*************************************************************************
output:

Enter String :preksha
String = preksha
 Which Character You Find :a
Character a is found at 6 position ..Press any key to continue . . .

*************************************************************************
*************************************************************************
```

11_b. Find the first occurrence of a string in another string. The function should return the position in the string.

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

int search(char str1[],char str2[])
{
      int i,j,found=1,index,itemp;
      char *sptr1,*sptr2;
      sptr1=str1;
      sptr2=str2;

      for(i=0; *(sptr1+i) != '\0'; i++)
      {
            if(*(sptr1+i) == *(sptr2))
            {
                  index=i;
                  itemp=i;
                  itemp++;
                  for(j=1;*(sptr2+j) != '\0';j++,itemp++)
                  {
                        if(*(sptr2+j) != *(sptr1+itemp))
                        {
                              found=0;
                              break;
                        }
                  }
                  if(found)
                  {
                        return index;
                  }
                  found=1;
            }
      }

      return -1;
}

void main()
{
      char str1[10],str2[10];
      int index;
      printf("Enter first string: ");
      gets(str1);
      printf("Enter second string: ");
      gets(str2);

      index=search(str1,str2);
      if(index == -1)
      {
            printf("Second string not found in first string!!");
      }
      else
      {
            printf("Second string found at index %d in first
string",index);
      }
      getch();
}
```

```
*************************************************************************
*************************************************************************
output:
Enter first string: prekshu
Enter second string: shu
Second string found at index 4 in first string

*************************************************************************
*************************************************************************
```

11_c. Delete all occurrences of a character from a string.

```
*************************************************************************
*************************************************************************
#include<stdio.h>
#include<conio.h>
void input_data(char str[])
{
      printf("Enter String =");
      gets(str);
}
void display11_c(char *ptr)
{
      printf("\nString = ");
      while(*ptr != '\0')
      {
            printf("%c",*ptr);
            ptr++;
      }
      printf("\n");
}
void delete_char(char *ptr,char ch,char str2[])
{
      char *ptr2;
      ptr2 = str2;
      while(*ptr != '\0')
      {
            //printf("aer");
            if(*ptr != ch)
            {
                  *ptr2 = *ptr;
                  ptr2++;
            }
            ptr++;
      }
      *ptr2 = '\0';
}

void main()
{
      char str[50],str2[50],ch;
      char *ptr;
      input_data(str);
      ptr = str;
      display11_c(ptr);
      printf("\nEnter Which Character You want to delete :");
      scanf("%c",&ch);
      delete_char(ptr,ch,str2);
      printf("\n========================================");
      printf("\nAfter Deleting Character:");
      printf("\n========================================");
      display11_c(str2);
}
```

```
*************************************************************************
*************************************************************************
output:

Enter String =preksha sheth

String = preksha sheth

Enter Which Character You want to delete :e

========================================
After Deleting Character:
========================================
String = prksha shth
Press any key to continue . . .
```

11_d. Delete all occurrences of a string from another string.

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

void remove_s(char str1[],char str2[])
{
      int i,j,k,found=1,index,itemp,len1,len2;
      char *sptr1,*sptr2;
      len1=strlen(str1);
      len2=strlen(str2);
      sptr1=str1;
      sptr2=str2;

      for(i=0; i<len1; i++)
      {
            if(*(sptr1+i) == *(sptr2))
            {
                  index=i;
                  itemp=i;
                  itemp++;
                  for(j=1;j<len2;j++,itemp++)
                  {
                        if(*(sptr2+j) != *(sptr1+itemp))
                        {
                              found=0;
                              break;
                        }
                  }
                  if(found)
                  {
                        for(k=0;k<len1;k++)
                        {

                              *(sptr1+index+k)=*(sptr1+index+len2+k);
                        }
                        len1-=len2;
                        *(str1+len1)='\0';
                        i--;
                  }
                  found=1;
            }
      }
}

void main()
{
      char str1[50],str2[40];
      int index;
      printf("Enter first string: ");
      gets(str1);
      printf("Enter second string: ");
      gets(str2);
      remove_s(str1,str2);
      printf("After removing second string: \n");
      puts(str1);

      getch();
 }


*************************************************************************
*************************************************************************
output:
Enter first string: Rollwala Computer Center
Enter second string: Computer Center
After removing second string:
Rollwala
```

```
*************************************************************************
*************************************************************************
```

11_e. Delete all occurrences of a character from a string. Ignore Case.

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

void delete_char(char str[],char ch)
{
      char *sptr=str;
      int i,j,length,index;
      length=strlen(str);

      for(i=0;i<length;i++)
      {
            if(*(sptr+i) == ch ||*(sptr+i) == ch-32 || *(sptr+i) == ch+32
)
            {
                  j=i;
                  while(j<length-1)
                  {
                        *(sptr+j)=*(sptr+j+1);
                        j++;
                  }
                  *(sptr+length-1)='\0';
                  length--;
                  i--;
            }
      }
}

void main()
{
      char str[10],ch;

      printf("Enter string: ");
      gets(str);
      printf("Enter character to delete its all occurrences: ");
      scanf("%c",&ch);
      delete_char(str,ch);
      printf("After removing all occurrences of your character from
string :");
      puts(str);
      getch();

}


*************************************************************************
*************************************************************************
```

```
output:

Enter string: Rollwala
Enter character to delete its all occurrences: l
After removing all occurrences of your character from string :Rowaa


*************************************************************************
*************************************************************************
11_f. Delete all occurrences of a string from another string. Ignore
Case.

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

void remove_str(char str1[],char str2[])
{
      int i,j,k,found=1,index,itemp,len1,len2;
      char *sptr1,*sptr2;
      len1=strlen(str1);
      len2=strlen(str2);
      sptr1=str1;
      sptr2=str2;

      for(i=0; i<len1; i++)

            if(*(sptr1+i) == *(sptr2)|| *(sptr1+i) == *(sptr2)-32
||*(sptr1+i) == *(sptr2)+32)
            {
                  index=i;

                  itemp=i;
                  itemp++;
                  for(j=1;j<len2;j++,itemp++)
                  {

                        if(*(sptr2+j) != *(sptr1+itemp) && *(sptr2+j) !=
*(sptr1+itemp)-32 && *(sptr2+j) != *(sptr1+itemp)+32)

                        {
                              found=0;
                              break;
                        }
                  }
                  if(found)
                  {
                        for(k=0;k<len1;k++)

                        {
                              *(sptr1+index+k)=*(sptr1+index+len2+k);
                        }
                        len1-=len2;

                        *(str1+len1)='\0';
                        i--;
                  }
                  found=1;
            }
       }
 }

 void main()
 {
       char str1[50],str2[50];
       int index;
       printf("Enter first string: ");
       gets(str1);
       printf("Enter second string: ");
       gets(str2);
       remove_str(str1,str2);
       printf("After removing second string: \n");
       puts(str1);

       getch();
  }


 **************************************************************************
 **************************************************************************
 output:

 Enter first string: ROLLWALA COMPUTER CENTER
 Enter second string: computer center
 After removing second string:
 ROLLWALA


 **************************************************************************
 **************************************************************************
 11_g. Copy one string to another string.

 **************************************************************************
 **************************************************************************
 #include<stdio.h>
 #include<conio.h>
 void get_input(char str1[])
 {
       printf("Enter String =");
       gets(str1);
 }
 void display(char *ptr)
 {
       //printf("grg");
       while(*ptr != '\0')
       {
            printf("%c",*ptr);
            ptr++;
       }
       printf("\n");
 }
 void copy(char str2[],char *ptr)
 {
       char *ptr2;
       ptr2 = str2;
       while(*ptr != '\0'){
            *ptr2 = *ptr;
            printf("\n%c  - >  %c",(*ptr),(*ptr2));
            ptr2++;
            ptr++;
       }
       *ptr2 = '\0';
 }

 void main()
 {
       char str1[50],str2[50];
       char *ptr;
       get_input(str1);
       ptr = str1;
       display(ptr);
       printf("\n==============================\n");
       printf("*** Cpoied String ***");
       printf("\n==============================\n");
       copy(str2,ptr);
       printf("\n\nString :");
       display(str2);
 }
```

```
*************************************************************************
*************************************************************************
output:

Enter String =preksha sheth
preksha sheth

==============================
*** Cpoied String ***
==============================

p  - >  p
r  - >  r
e  - >  e
k  - >  k
s  - >  s
h  - >  h
a  - >  a
   - >
s  - >  s
```

```
h  - >  h
e  - >  e
t  - >  t
h  - >  h

String :preksha sheth
Press any key to continue . . .

*************************************************************************
*************************************************************************
11_h.Copy n characters of one string to another string.

*************************************************************************
*************************************************************************
#include<stdio.h>
#include<conio.h>
void input11_h(char str[])
{
      printf("Enter String :");
      gets(str);
}
void disp11_h(char *ptr)
{
      printf("String = ");
      while(*ptr != '\0')
      {
            printf("%c",*ptr);
            ptr++;
      }
      printf("\n");
}

void copych (char *ptr,char *cptr,int n)
{
      int i = 0;
      for(i = 0; i<n ; i++)
      {
            //printf("asd");
            *cptr = *ptr;
            cptr++;
            ptr++;
      }
      *cptr = '\0';
}

void main()
{
      char str[50],cstr[50];
      char *ptr,*cptr;
      int n;//i = 0;
      input11_h(str);
      /*ptr = str;
      while (*ptr != '\0')
      {
            i++;
            ptr++;
      }*/
      ptr = str;
      printf("\n=================================================\n");
      printf("\t\t Original String ");
      printf("\n=================================================\n");
      disp11_h(ptr);
      printf("Enter How Many Characters You want to Copy in Another 
String :");
      scanf("%d",&n);
      //if(n <= i && n > 0)
      //{
            cptr = cstr;

      printf("\n=================================================\n");
            printf("\t\tCopied String ");

      printf("\n=================================================\n");
            copych(ptr,cptr,n);
            disp11_h(cptr);
      //}
      //printf("%d",n);
      //cptr = cstr;
//    else
//    {
//          printf("\nNumber must be less than the String length..");
//    }
}
```

```
*************************************************************************
*************************************************************************
output:

Enter String :preksha sheth

=================================================
                 Original String
=================================================
String = preksha sheth
Enter How Many Characters You want to Copy in Another String :9

=================================================
                Copied String
=================================================
String = preksha s
Press any key to continue . . .

*************************************************************************
*************************************************************************
```

11_i. Find length of the string and toggle the characters of the string.

```
*************************************************************************
*************************************************************************
```

```
#include<stdio.h>
#include<conio.h>
void input11_i(char str[])
{
      printf("Enter String :");
      gets(str);
}
void disp(char *ptr)
{
      printf("\nString = ");
      while(*ptr != '\0')
      {
            printf("%c",*ptr);
            ptr++;
      }
      printf("\n");
}
void toggle(char *ptr)
{
      while(*ptr != '\0')
      {
            if(*ptr >= 65 && *ptr <= 90)
            {
                  *ptr = *ptr + 32;
            }
            else if(*ptr >= 97 && *ptr <= 122)
            {
                  *ptr = *ptr - 32;
            }
            else
            {

            }
            ptr++;
      }
}

void main()
{
      char str[50];
      char *ptr;
      input11_i(str);
      ptr = str;
      disp(ptr);
      printf("\n=================================================\n");
      printf("\t\tTOGGLE STRING");
      printf("\n=================================================\n");
      toggle(ptr);
      disp(ptr);
}

*************************************************************************
*************************************************************************
```

```
output:

Enter String :preksha SHETH 123

String = preksha SHETH 123

=================================================
                TOGGLE STRING
=================================================

String = PREKSHA sheth 123
Press any key to continue . . .

*************************************************************************
*************************************************************************
11_j.Convert string to all upper case.

*************************************************************************
*************************************************************************
#include<stdio.h>
#include<conio.h>
void input(char str[])
{
      printf("Enter String :");
      gets(str);
}

void display11_j(char *ptr)
{
      printf("String = ");
      while(*ptr != '\0')
      {
            printf("%c",*ptr);
            ptr++;
      }
      printf("\n");
}

void toupper(char *ptr)
{
      while(*ptr != '\0')
      {
            if(*ptr >= 97 && *ptr <=122)
            {
                  *ptr = *ptr - 32;
                  //*ptr++;
            }
            ptr++;
            //printf("%c",*ptr2);
      }
}

void main()
```

```
{
      char str[100];
      char *ptr;
      input(str);
      ptr = str;
      display11_j(ptr);
      printf("\n======================================================\n"
);
      printf("\n\t\tAfter Converting ToUpper Case\n ");
      printf("\n======================================================\n"
);
      toupper(ptr);
      display11_j(ptr);
}
```

```
*************************************************************************
*************************************************************************
output:

Enter String :preksha
String = preksha

======================================================

                After Converting ToUpper Case

======================================================
String = PREKSHA
Press any key to continue . . .

*************************************************************************
*************************************************************************
```

11_k.Convert string to all Lower case.

```
*************************************************************************
*************************************************************************
#include<stdio.h>
#include<conio.h>
void inputk(char str[])
{
      printf("Enter String :");
      gets(str);
}

void display11_k(char *ptr)
{
      printf("String = ");
      while(*ptr != '\0')
      {
            printf("%c",*ptr);
            ptr++;
      }
      printf("\n");
}
```

```
void tolower(char *ptr)
{
      while(*ptr != '\0')
      {
            if(*ptr >= 65 && *ptr <=90)
            {
                  *ptr = *ptr + 32;
                  //*ptr++;
            }
            ptr++;
            //printf("%c",*ptr2);
      }
}

void main()
{
      char str[100];
      char *ptr;
      inputk(str);
      ptr = str;
      display11_k(ptr);
      printf("\n======================================================\n"
);
      printf("\n\t\tAfter Converting ToLower Case\n ");
      printf("\n======================================================\n"
);
      tolower(ptr);
      display11_k(ptr);
}


*************************************************************************
*************************************************************************
output:

Enter String :PREKSHA
String = PREKSHA

======================================================

                After Converting ToLower Case

======================================================
String = preksha
Press any key to continue . . .

*************************************************************************
*************************************************************************
11_L. Sort an array of string.

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

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

void readnames(char names[][20],int limit)
{
      char (*cptr)[20];
      int i;
      cptr=names;
      fflush(stdin);
      printf("Enetr %d names: \n",limit);
      for(i=0;i<limit;i++)
      {
            gets(*(cptr+i));
      }
}

void sort(char names[][20],int limit)
{
      char (*cptr)[20],min[20];
      int i,index,j;

      cptr=names;

      for(i=0;i<limit-1;i++)
      {
            strcpy(min,*(cptr+i));
            index=i;

            for(j=i+1;j<limit;j++)
            {
                  if(strcmp(min,*(cptr+j))>0)
                  {
                        strcpy(min,*(cptr+j));
                        index=j;
                  }

            }
            strcpy(*(cptr+index),*(cptr+i));
            strcpy(*(cptr+i),min);
      }

}
void print(char names[][20],int limit)
{
      char (*cptr)[20];
      int i;
      cptr=names;
      printf("Sorted names:\n");
      for(i=0;i<limit;i++)
      {
            puts(*(cptr+i));
      }
}
```

```
void main()
{
      char names[5][20];
      int limit;
      printf("Hoe many names you have: ");
      scanf("%d",&limit);
      readnames(names,limit);
      sort(names,limit);
      print(names,limit);
      getch();
}
```

```
*************************************************************************
*************************************************************************
output:

Hoe many names you have: 3
Enetr 3 names:

surat
anand
navsari

Sorted names:

anand
navsari
surat


*************************************************************************
*************************************************************************
11_m. (m)Append one string to another string.

*************************************************************************
*************************************************************************
```

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

void append(char str1[],char str2[])
{
      int len1,len2,i;
      char *sptr1,*sptr2;

      sptr1=str1;
      sptr2=str2;
      len1=strlen(sptr1);
      len2=strlen(sptr2);
      *(sptr1+len1)=' ';
      for(i=0;i<len2;i++)
      {
            *(sptr1+len1+i+1)=*(sptr2+i);
      }

      *(sptr1+len1+i+1)='\0';

}

void main()
{
      char str1[30],str2[30];

      printf("ENter first string: ");
      gets(str1);
      printf("Enter second string: ");
      gets(str2);
      append(str1,str2);
      printf("String after appending: ");
      puts(str1);
      getch();
}
```

```
**************************************************************************
**************************************************************************
output:

ENter first string: Rollwala
Enter second string: Computer Center
String after appending: Rollwala Computer Center


**************************************************************************
**************************************************************************
```

11_n. Append at most n characters of one string S2 to another string S1.

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

void append_n(char str1[],char str2[],int n)
{
      int len1,len2,i;
      char *sptr1,*sptr2;

      sptr1=str1;
      sptr2=str2;
      len1=strlen(sptr1);
      len2=strlen(sptr2);
      *(sptr1+len1)=' ';
      for(i=0;i<n;i++)
      {
            *(sptr1+len1+i+1)=*(sptr2+i);
      }

      *(sptr1+len1+i+1)='\0';

}

void main()
{
      int n;
      char str1[30],str2[30];

      printf("ENter first string: ");
      gets(str1);
      printf("Enter second string: ");
      gets(str2);
      printf("Enetr how many characters you want to append: ");
      scanf("%d",&n);
      append_n(str1,str2,n);
      printf("After appending %d characters: ",n);
      puts(str1);
      getch();
}
```

```
*************************************************************************
*************************************************************************
output:

ENter first string: Gujarat
Enter second string: University Ahmedabad
Enetr how many characters you want to append: 10
After appending 10 characters: Gujarat University


*************************************************************************
*************************************************************************
```

11_o. Reverse all the characters in the string.

```
*************************************************************************
*************************************************************************
```

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

void reverse_string(char *str)
{
      int len , i ;
      char *start, *end, ch;

      len=strlen(str);
      //printf("count = %d",len);

      start = str;
      end = str;
      for(i = 0 ; i < len-1 ; i++)
      {
            end++;
      }
      //printf("end = %s",*end);
      for(i=0 ; i < len/2 ; i++)
      {
            ch = *start;
            *start = *end;
            *end = ch;
            start++;
            end--;
      }
      //printf("Reverse String = %s",*str);
}

void main()
{
      char str[100];
      printf("Enter String :");
      gets(str);
      printf("\n=======================================\n");
      printf("\t*** Original String ***");
      printf("\n=======================================\n");
      printf("Original String  = %s",str);

      printf("\n=======================================\n");
      printf("\t*** Reverse String ***");
      printf("\n=======================================\n");
      reverse_string(str);
      printf("\nReverse String = %s\n\n",str);
}
```

```
*************************************************************************
*************************************************************************
output:

Enter String :preksha sheth

=======================================
        *** Original String ***
=======================================
Original String  = preksha sheth
=======================================
        *** Reverse String ***
=======================================

Reverse String = htehs ahskerp

Press any key to continue . . .
*************************************************************************
*************************************************************************
```

11_p. Compare two strings S1 and S2. The function should return -1, 0 or 1 if S1 < S2, S1 = S2 and S1 > S2 respectively.

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

int str_compare(char s1[],char s2[])
{
      char *sp1,*sp2;
      int i=0;

      sp1=s1;
      sp2=s2;

      while(*(sp1+i) != '\0' && *(sp2+i) != '\0')
      {
            if(*(sp1+i) < *(sp2+i))
            {
                  return -1;
            }
            else if(*(sp1+i) > *(sp2+i))
            {
                  return 1;
            }
            i++;
      }

      if(*(sp1+i) != '\0' && *(sp2+i) == '\0')
      {
            return 1;
      }
      else if(*(sp1+i) == '\0' && *(sp2+i) != '\0')
      {
            return -1;
      }
      else
            return 0;

}

void main()
{
      char s1[30],s2[30];
      int result;

      printf("Enter string1: ");
      gets(s1);
      printf("Enter string 2: ");
      gets(s2);
      result=str_compare(s1,s2);
      printf("result is %d",result);
      getch();
}


**************************************************************************
**************************************************************************
output1:

Enter string1: Preksha
Enter string 2: preksha
result is -1

**************************************************************************
**************************************************************************
output2:
Enter string1: preksha
Enter string 2: preksha
result is 0

**************************************************************************
**************************************************************************
11_q. (q) Compare two strings S1 and S2. The function should return -1, 0
or 1 if S1 < S2, S1 = S2 and S1 > S2 respectively.
      Ignore case.

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

int str_icompare(char s1[],char s2[])
{
      char *sp1,*sp2;
      int i=0;

      sp1=s1;
      sp2=s2;

      while(*(sp1+i) != '\0' && *(sp2+i) != '\0')
      {
            if(*(sp1+i) < *(sp2+i))
            {
                  if( *(sp1+i) < *(sp2+i)-32)
                  return -1;
            }
            else if(*(sp1+i) > *(sp2+i))
            {
                  if( *(sp1+i) < *(sp2+i)+32)
                  return 1;
            }
            i++;
      }
      if(*(sp1+i) != '\0' && *(sp2+i) == '\0')
      {
            return 1;
      }
      else if(*(sp1+i) == '\0' && *(sp2+i) != '\0')
      {
            return -1;
      }
      else
            return 0;

}

void main()
{
      char s1[30],s2[30];
      int result;

      printf("Enter string1: ");
      gets(s1);
      printf("Enter string 2: ");
      gets(s2);
      result=str_icompare(s1,s2);
      printf("result is %d",result);

      getch();
}
```

```
**************************************************************************
**************************************************************************
output1:

Enter string1: PREKSHA
Enter string 2: preksha
result is 0

**************************************************************************
**************************************************************************
```

11_r. Compare at most n characters of two strings S1 and S2. The function should return -1, 0 or 1 if S1 < S2, S1 = S2 and
S1 > S2 respectively.

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

int str_ncompare(char s1[],char s2[],int n)
{
      char *sp1,*sp2;
      int i=0;
      sp1=s1;
      sp2=s2;

      while((*(sp1+i) != '\0' && *(sp2+i) != '\0') && i<n)
      {
            if(*(sp1+i) < *(sp2+i))
            {
                  return -1;
            }
            else if(*(sp1+i) > *(sp2+i))
            {
                  return 1;
            }
            i++;
      }

      if(*(sp1+i) != '\0' && *(sp2+i) == '\0' && i<n)
      {
            return 1;
      }
      else if(*(sp1+i) == '\0' && *(sp2+i) != '\0' && i<n)
      {
            return -1;
      }
      else
            return 0;

}

void main()
{
      char s1[30],s2[30];
      int result,n;

      printf("Enter string1: ");
      gets(s1);
      printf("Enter string 2: ");
      gets(s2);
      printf("Enter how many characters you want ot compare: ");
      scanf("%d",&n);
      result=str_ncompare(s1,s2,n);
      printf("result is %d",result);

      getch();
}
```

```
**************************************************************************
**************************************************************************
output:
Enter string1: rollwala
Enter string 2: rollwala computer
Enter how many characters you want ot compare: 8
result is 0
```

```
*************************************************************************
*************************************************************************
11_s. Compare at most n characters of two strings S1 and S2. The function
should return -1, 0 or 1 if  S1 < S2, S1 = S2 and
      S1 > S2 respectively.Ignore case.
*************************************************************************
*************************************************************************
#include<stdio.h>
#include<conio.h>
#include<string.h>

int str_incompare(char s1[],char s2[],int n)
{
      char *sp1,*sp2;
      int i=0;

      sp1=s1;
      sp2=s2;

      while((*(sp1+i) != '\0' && *(sp2+i) != '\0') && i<n)
      {
            if(*(sp1+i) < *(sp2+i))
            {
                  if( *(sp1+i) < *(sp2+i)-32)
                  return -1;
            }
            else if(*(sp1+i) > *(sp2+i))
            {
                  if( *(sp1+i) < *(sp2+i)+32)
                  return 1;
            }
            i++;
      }

      if(*(sp1+i) != '\0' && *(sp2+i) == '\0' && i<n)
      {
            return 1;
      }
      else if(*(sp1+i) == '\0' && *(sp2+i) != '\0' && i<n)
      {
            return -1;
      }
      else
            return 0;

}

void main()
{
      char s1[30],s2[30];
      int result,n;

      printf("Enter string1: ");
      gets(s1);
      printf("Enter string 2: ");
      gets(s2);
      printf("Enter how many characters you want ot compare: ");
      scanf("%d",&n);
      result=str_incompare(s1,s2,n);
      printf("result is %d",result);

      getch();
}
```

```
*************************************************************************
*************************************************************************
output:

Enter string1: Preksha
Enter string 2: preksha sheth
Enter how many characters you want ot compare: 6
result is 0

*************************************************************************
*************************************************************************
```

ASSIGNMENT - 3

```
*************************************************************************
*************************************************************************
```

1 . Write a program to create a singly linked list and display its elements in FIFO pattern. Also display
     the number of elements in the list.

```
*************************************************************************
*************************************************************************
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
      int rollno;
      float marks;
      struct node *link;
};
void insert_end(struct node **first,struct node**last,int rollno,float
marks)
{
      struct node *newnode = NULL;
      newnode = (struct node*)malloc(sizeof(struct node));
      newnode->rollno = rollno;
      newnode->marks = marks;
      newnode->link = NULL;
      if(*first != NULL)
      {
            (*last)->link = newnode;
            printf("\nNew Node is inserted at the End..\n");
      }
      else
      {
            *first = newnode;
            printf("\nNew Node is Inserted..");
      }
      *last = newnode;
}
void delete_beg(struct node **first,struct node **last)
{
      struct node *temp;
      if(*first == NULL)
      {
            printf("\nLinked List is Empty..\n");
      }
      else
      {
            temp = *first;
            if((*first)->link == NULL)
            {
                  printf("\n----------------------------------------------
--\n");
                  printf("\nFirst Node is Deleted..\nLined List is
Empty..\n");
                  *first = NULL;
                  *last = NULL;
            }
            else
            {
                  *first = temp->link;
                  *last = *first;
                  printf("\n----------------------------------------------
--\n");
                  printf("\n First node Deleted..\n");
            }
            free(temp);
      }
}
void display(struct node *first)
{
      struct node *temp;
      printf("\n------------------------------------------------\n");
      printf("\t\t*** Display Details ***");
      printf("\n------------------------------------------------\n");
      if(first == NULL)
      {
            printf("\nLinked List is Empty..\n");
      }
      else
      {
            temp = first;
            while(temp != NULL)
            {
                  printf("\n Roll No = %d",temp->rollno);
                  printf("\n Marks = %f",temp->marks);
                  temp = temp->link;
            }
            printf("\n\n");
      }
}
void main()
{
      int ch = 1;
      int rollno,count = 0;
      float marks;
      struct node *first = NULL;
      struct node *last = NULL;
      //printf("dd");
      while(ch != 0)
      {
            printf("\n------------------------------------------------
\n");
            printf("\n1. Insert Node at the End..");
            printf("\n2. Delete Node From Beggning..");
            printf("\n3. Display Data..");
            printf("\n4. Display No. of Elements..");
            printf("\n0. Exit..");
            printf("\n\n Choose Any Function in Above..");
            scanf("%d",&ch);
            if(ch == 1)
            {
                  printf("\n----------------------------------------------
--\n");
                  printf("Enter RollNo :");
                  scanf("%d",&rollno);
                  printf("Enter Marks :");
                  scanf("%f",&marks);
                  insert_end(&first,&last,rollno,marks);
                  if(first != NULL)
                  {
                        count++;
                  }
            }
            else if(ch == 2)
            {
                  if(first != NULL)
                  {
                        count--;
                  }
                  delete_beg(&first,&last);
            }
            else if(ch == 3)
            {
                  display(first);
            }
            else if(ch == 4)
            {
                 printf("\n----------------------------------------------
--\n");
                 printf("\n Number of Elements are = %d",count);
            }
            else if(ch == 0)
            {
                 exit(0);
            }
      }
}
*************************************************************************
*************************************************************************
output:


------------------------------------------------

1. Insert Node at the End..
2. Delete Node From Beggning..
3. Display Data..
4. Display No. of Elements..
0. Exit..

 Choose Any Function in Above..1

------------------------------------------------
Enter RollNo :36
Enter Marks :45

New Node is Inserted..
------------------------------------------------

1. Insert Node at the End..
2. Delete Node From Beggning..
3. Display Data..
4. Display No. of Elements..
0. Exit..

 Choose Any Function in Above..1

------------------------------------------------
Enter RollNo :32
Enter Marks :43

New Node is inserted at the End..

------------------------------------------------

1. Insert Node at the End..
2. Delete Node From Beggning..
3. Display Data..
4. Display No. of Elements..
0. Exit..
```

```
 Choose Any Function in Above..1

------------------------------------------------
Enter RollNo :30
Enter Marks :47

New Node is inserted at the End..

------------------------------------------------

1. Insert Node at the End..
2. Delete Node From Beggning..
3. Display Data..
4. Display No. of Elements..
0. Exit..

 Choose Any Function in Above..3

------------------------------------------------
                *** Display Details ***
------------------------------------------------

 Roll No = 36
 Marks = 45.000000
 Roll No = 32
 Marks = 43.000000
 Roll No = 30
 Marks = 47.000000


------------------------------------------------

1. Insert Node at the End..
2. Delete Node From Beggning..
3. Display Data..
4. Display No. of Elements..
0. Exit..

 Choose Any Function in Above..2

------------------------------------------------

 First node Deleted..

------------------------------------------------

1. Insert Node at the End..
2. Delete Node From Beggning..
3. Display Data..
4. Display No. of Elements..
0. Exit..

 Choose Any Function in Above..3
```

```
------------------------------------------------
                *** Display Details ***
------------------------------------------------

 Roll No = 32
 Marks = 43.000000
 Roll No = 30
 Marks = 47.000000


------------------------------------------------

1. Insert Node at the End..
2. Delete Node From Beggning..
3. Display Data..
4. Display No. of Elements..
0. Exit..

 Choose Any Function in Above..4

------------------------------------------------

 Number of Elements are = 2
------------------------------------------------

1. Insert Node at the End..
2. Delete Node From Beggning..
3. Display Data..
4. Display No. of Elements..
0. Exit..

 Choose Any Function in Above..
*************************************************************************
*************************************************************************
```

2 . Write a program to create a singly linked list and display its elements in LIFO pattern.
    Also display the number of elements in the list.

```
*************************************************************************
*************************************************************************
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
      int rollno;
      float marks;
      struct node *link;
};
struct node* insert_beg(struct node *first,int rollno,float marks)
{
      struct node *temp;
      temp = (struct node*)malloc(sizeof(struct node));
      temp->rollno = rollno;
      temp->marks = marks;
      temp->link = NULL;
      if(first != NULL)
      {
            temp->link = first;
      }
      first = temp;
      printf("\nFirst Node is Inserted..\n");
      return first;
}
struct node* delete_beg(struct node *first)
{
      struct node *temp;
      if(first == NULL)
      {
            printf("\n------------------------------------------------
\n");
            printf("\nLinkedList is Empty..\n");
      }
      else
      {
            temp = first;
            if(first->link == NULL)
            {
                  free(temp);
                  first = NULL;
                  printf("\n----------------------------------------------
--\n");
                  printf("\nFirst Node is Deleted..\n Linked List is
Empty..\n");
            }
            else
            {
                  first = temp->link;
                  free(temp);
                  printf("\n----------------------------------------------
--\n");
                  printf("\nFirst Node is Deleted..\n");
            }
      }
      return first;
}
void display(struct node *first)
{
      struct node *temp;
      printf("\n------------------------------------------------\n");
      printf("\t\t*** Display Details ***");
      printf("\n------------------------------------------------\n");
      if(first == NULL)
      {
            printf("\nLinked List is Empty..\n");
      }
      else
      {
            temp = first;
            while(temp != NULL)
            {
                  printf("\n Roll No = %d",temp->rollno);
                  printf("\n Marks = %f",temp->marks);
                  temp = temp->link;
            }
            printf("\n\n");
      }
}
void main()
{
      int ch = 1;
      int rollno,count = 0;
      float marks;
      struct node *first = NULL;
      while(ch != 0)
      {
            printf("\n-------------------------------------------------
\n");
            printf("\n1. Insert Node at the Beginning..");
            printf("\n2. Delete Node From Beginning..");
            printf("\n3. Display Data..");
            printf("\n4. Display No. of Elements..");
            printf("\n0. Exit..");
            printf("\n\n Choose Any Function in Above..");
            scanf("%d",&ch);
            if(ch == 1)
            {
                  printf("\n-----------------------------------------------
--\n");
                  printf("Enter RollNo :");
                  scanf("%d",&rollno);
                  printf("Enter Marks :");
                  scanf("%f",&marks);
                  first = insert_beg(first,rollno,marks);
                  if(first != NULL)
                  {
                        count++;
                  }
            }
            else if(ch == 2)
            {
                  if(first != NULL)
                  {
                        count--;
                  }
                  first = delete_beg(first);
            }
            else if(ch == 3)
            {
                  display(first);
            }
            else if(ch == 4)
            {
                 printf("\n----------------------------------------------
--\n");
                 printf("\n Number of Elements are = %d",count);
            }
            else if(ch == 0)
            {
                 exit(0);
            }
      }
}
*************************************************************************
*************************************************************************
output:


------------------------------------------------

1. Insert Node at the Beginning..
2. Delete Node From Beginning..
3. Display Data..
4. Display No. of Elements..
0. Exit..

 Choose Any Function in Above..1

------------------------------------------------
Enter RollNo :36
Enter Marks :47

First Node is Inserted..

------------------------------------------------

1. Insert Node at the Beginning..
2. Delete Node From Beginning..
3. Display Data..
4. Display No. of Elements..
0. Exit..

 Choose Any Function in Above..1

------------------------------------------------
Enter RollNo :32
Enter Marks :43

First Node is Inserted..

------------------------------------------------

1. Insert Node at the Beginning..
2. Delete Node From Beginning..
3. Display Data..
4. Display No. of Elements..
```

```
0. Exit..

 Choose Any Function in Above..1

------------------------------------------------
Enter RollNo :39
Enter Marks :21

First Node is Inserted..

------------------------------------------------

1. Insert Node at the Beginning..
2. Delete Node From Beginning..
3. Display Data..
4. Display No. of Elements..
0. Exit..

 Choose Any Function in Above..3

------------------------------------------------
                *** Display Details ***
------------------------------------------------

 Roll No = 39
 Marks = 21.000000
 Roll No = 32
 Marks = 43.000000
 Roll No = 36
 Marks = 47.000000


------------------------------------------------

1. Insert Node at the Beginning..
2. Delete Node From Beginning..
3. Display Data..
4. Display No. of Elements..
0. Exit..

 Choose Any Function in Above..2

------------------------------------------------

First Node is Deleted..

------------------------------------------------

1. Insert Node at the Beginning..
2. Delete Node From Beginning..
3. Display Data..
4. Display No. of Elements..
0. Exit..
```

```
 Choose Any Function in Above..3

------------------------------------------------
                *** Display Details ***
------------------------------------------------

 Roll No = 32
 Marks = 43.000000
 Roll No = 36
 Marks = 47.000000


------------------------------------------------

1. Insert Node at the Beginning..
2. Delete Node From Beginning..
3. Display Data..
4. Display No. of Elements..
0. Exit..

 Choose Any Function in Above..1

------------------------------------------------
Enter RollNo :30
Enter Marks :44

First Node is Inserted..

------------------------------------------------

1. Insert Node at the Beginning..
2. Delete Node From Beginning..
3. Display Data..
4. Display No. of Elements..
0. Exit..

 Choose Any Function in Above..3

------------------------------------------------
                *** Display Details ***
------------------------------------------------

 Roll No = 30
 Marks = 44.000000
 Roll No = 32
 Marks = 43.000000
 Roll No = 36
 Marks = 47.000000


------------------------------------------------

1. Insert Node at the Beginning..
2. Delete Node From Beginning..
```

```
3. Display Data..
4. Display No. of Elements..
0. Exit..

 Choose Any Function in Above..4

------------------------------------------------

 Number of Elements are = 3
------------------------------------------------

1. Insert Node at the Beginning..
2. Delete Node From Beginning..
3. Display Data..
4. Display No. of Elements..
0. Exit..

 Choose Any Function in Above..
```

****
****

3 .Write a menu driven program to create a singly linked list and perform following operations on it:

a. Insert an element
b. Delete an element
c. Display the list

****
****

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
      int rollno;
      int marks;
      struct node *next;
};
struct node* insert_beg(struct node *first,int rollno,int marks)
{
      struct node *temp;
      temp = (struct node*)malloc(sizeof(struct node));
      temp->rollno = rollno;
      temp->marks = marks;
      temp->next = NULL;
      if(first != NULL)
      {
            temp->next = first;
      }
      first = temp;
      printf("\n First Node inserted..\n");
      return first;
}
struct node* delete_beg(struct node *first)
```

```
{
      struct node *temp;
      if(first == NULL)
      {
            printf("\n------------------------------------------------
\n");
            printf("\nLinkedList is Empty..\n");
      }
      else
      {
            temp = first;
            if(first->next == NULL)
            {
                  free(temp);
                  first = NULL;
                  printf("\n----------------------------------------------
--\n");
                  printf("\nFirst Node is Deleted..\n Linked List is
Empty..\n");
            }
            else
            {
                  first = temp->next;
                  free(temp);
                  printf("\n----------------------------------------------
--\n");
                  printf("\nFirst Node is Deleted..\n");
            }
      }
      return first;
}
void display(struct node *first)
{
      struct node *temp;
      printf("\n------------------------------------------------\n");
      printf("\t\t*** Display Details ***");
      printf("\n------------------------------------------------\n");
      if(first == NULL)
      {
            printf("\nLinked List is Empty..\n");
      }
      else
      {
            temp = first;
            while(temp != NULL)
            {
                  printf("\n Roll No = %d",temp->rollno);
                  printf("\n Marks = %f",temp->marks);
                  temp = temp->next;
            }
            printf("\n\n");
      }
}
```

```
void main()
{
      int ch = 1;
      int rollno,marks;
      struct node *first = NULL;
      //struct node *last = NULL;
      while(ch != 0)
      {
            printf("\n1. Insert Element..\n2. Delete Element..\n3.
Display Element..\n 0.Exit...\n");
            printf("\nChoose any of the above Function..");
            scanf("%d",&ch);
            if(ch == 1)
            {
                  printf("\n Enter Rollno :");
                  scanf("%d",&rollno);
                  printf("\n Enter Marks :");
                  scanf("%d",&marks);
                  first = insert_beg(first,rollno,marks);
            }
            else if(ch == 2)
            {
                  first = delete_beg(first);
            }
            else if(ch == 3)
            {
                  display(first);
            }
            else if(ch == 0)
            {
                  exit(0);
            }
      }
}

**************************************************************************
**************************************************************************
output:


1. Insert Element..
2. Delete Element..
3. Display Element..
 0.Exit...

Choose any of the above Function..1

 Enter Rollno :11

 Enter Marks :43

 First Node inserted..

1. Insert Element..
```

```
2. Delete Element..
3. Display Element..
 0.Exit...

Choose any of the above Function..1

 Enter Rollno :6

 Enter Marks :43

 First Node inserted..

1. Insert Element..
2. Delete Element..
3. Display Element..
 0.Exit...

Choose any of the above Function..1

 Enter Rollno :36

 Enter Marks :47

 First Node inserted..

1. Insert Element..
2. Delete Element..
3. Display Element..
 0.Exit...

Choose any of the above Function..3

------------------------------------------------
                *** Display Details ***
------------------------------------------------

 Roll No = 36
 Marks = 0.000000
 Roll No = 6
 Marks = 0.000000
 Roll No = 11
 Marks = 0.000000


1. Insert Element..
2. Delete Element..
3. Display Element..
 0.Exit...

Choose any of the above Function..2

------------------------------------------------

First Node is Deleted..
```

```
1. Insert Element..
2. Delete Element..
3. Display Element..
 0.Exit...

Choose any of the above Function..3

------------------------------------------------
                *** Display Details ***
------------------------------------------------

 Roll No = 6
 Marks = 0.000000
 Roll No = 11
 Marks = 0.000000


1. Insert Element..
2. Delete Element..
3. Display Element..
 0.Exit...

Choose any of the above Function..1

 Enter Rollno :30

 Enter Marks :33

 First Node inserted..

1. Insert Element..
2. Delete Element..
3. Display Element..
 0.Exit...

Choose any of the above Function..3

------------------------------------------------
                *** Display Details ***
------------------------------------------------

 Roll No = 30
 Marks = 0.000000
 Roll No = 6
 Marks = 0.000000
 Roll No = 11
 Marks = 0.000000


1. Insert Element..
2. Delete Element..
3. Display Element..
 0.Exit...
```

```
Choose any of the above Function..0
Press any key to continue . . .
```

****
****

4 . Write a program to create an ordered linked list.

****
****

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
      int rollno;
      float marks;
      struct node *link;
};

void insert_node(struct node **start,struct node **last,int rlno,float
mark)
{
      struct node *before = NULL;
      struct node *temp = NULL;
      struct node *n = NULL;
      n = (struct node*)malloc(sizeof(struct node));
      n->rollno = rlno;
      n->marks = mark;
      n->link = NULL;
      if(*start == NULL)
      {
            *start = n;
            *last = n;
            printf("\n First Node inserted..\n");
      }

      else if(n->rollno < (*start)->rollno)
      {
            n->link = (*start);
            *start = n;
            printf("\n Node inserted at end.. \n");
      }
      else if(n->rollno > (*start)->rollno)
      {
            temp = (*start);
            if(temp->link == NULL)
            {
                  temp->link = n;
                  *last = n;
                  return;
            }
            while(n->rollno > temp->rollno && temp->link != NULL)
            {
                  before = temp;
                  temp = temp->link;
            }
            if(n->rollno > temp->rollno)
            {
                  temp->link = n;
                  *last = n;
            }
            else if(temp->link == NULL)
            {
                  before->link = n;
                  n->link = temp->link;
                  *last = temp;
            }
            else
            {
                  n->link = before->link;
                  before->link = n;
                  *last = n;
            }
      }
}


void display(struct node *start)
{
      struct node *temp;
      printf("\n-------------------------------------------------\n");
      printf("\t\t*** Display Details ***");
      printf("\n-------------------------------------------------\n");
      if(start == NULL)
      {
            printf("\n List is empty");
      }
      else
      {
            temp = start;
            while(temp != NULL)
            {
                  printf("\n Roll No = %d",temp->rollno);
                  printf("\n Marks = %f",temp->marks);
                  temp = temp->link;
            }
            printf("\n\n");
      }
}
void main()
{
      int ch = 1,rlno;
      float mark;
      struct node *start = NULL;
      struct node *last = NULL;
      while(ch != 0)
      {
            printf("\n------------------------\n");
            printf("\n1.Insert Node..");
            printf("\n2.Display Node");
            printf("\n0.exit..\n");
            printf("\nEnter your choice : ");
            scanf("%d",&ch);
            if(ch == 1)
            {
                  printf("\n Enter Rollno:");
                  scanf("%d",&rlno);
                  printf("\n Enter Marks:");
                  scanf("%f",&mark);
                  insert_node(&start,&last,rlno,mark);
            }
            else if(ch == 2)
                  display(start);
      }
}
```

```
**************************************************************************
**************************************************************************
output:

------------------------

1.Insert Node..
2.Display Node
0.exit..

Enter your choice : 1

 Enter Rollno:36

 Enter Marks:45

 First Node inserted..

------------------------

1.Insert Node..
2.Display Node
0.exit..

Enter your choice : 1

 Enter Rollno:32

 Enter Marks:43

 Node inserted at end..

------------------------
```

```
1.Insert Node..
2.Display Node
0.exit..

Enter your choice : 1

 Enter Rollno:43

 Enter Marks:23

------------------------

1.Insert Node..
2.Display Node
0.exit..

Enter your choice : 2

------------------------------------------------
                *** Display Details ***
------------------------------------------------

 Roll No = 32
 Marks = 43.000000
 Roll No = 36
 Marks = 45.000000
 Roll No = 43
 Marks = 23.000000


------------------------

1.Insert Node..
2.Display Node
0.exit..

Enter your choice : 1

 Enter Rollno:12

 Enter Marks:33

 Node inserted at end..

------------------------

1.Insert Node..
2.Display Node
0.exit..

Enter your choice : 2

------------------------------------------------
                *** Display Details ***
```

```
------------------------------------------------

 Roll No = 12
 Marks = 33.000000
 Roll No = 32
 Marks = 43.000000
 Roll No = 36
 Marks = 45.000000
 Roll No = 43
 Marks = 23.000000


------------------------

1.Insert Node..
2.Display Node
0.exit..

Enter your choice :
```

****
****
5 . Write a program to reverse a given linked list.

****
****

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
      int no;
      struct node *next;
};
struct node* insert_ele_beg(struct node *start,int no)
{
      struct node *temp;
      temp = (struct node*)malloc(sizeof(struct node));
      temp->no = no;
      temp->next = NULL;
      if(start != NULL)
      {
            temp->next = start;
      }
      start = temp;
      printf("\nFirst Node Inserted..\n");
      return start;
}

struct node* reverse_ele(struct node *start)
{
      struct node *t1 = NULL;
      struct node *t2 = NULL;
      while(start != NULL)
      {
            t2 = start->next;
            start->next = t1;
            t1 = start;
            start = t2;
      }
      printf("\n===========================================\n");
      printf("\n Linked List Reversed..");
      printf("\n===========================================\n");
      return t1;
}
void display(struct node *start)
{
      struct node *temp;
      printf("\n===========================================\n");
      printf("\t** Display Information **");
      printf("\n===========================================\n");
      if(start == NULL)
      {
            printf("\nLinked List is Empty...");
      }
      else
      {
            temp = start;
            while(temp != NULL)
            {
                  printf("\nNo = %d",temp->no);
                  temp = temp->next;
            }
            printf("\n\n");
      }
}
void main()
{
      int ch = 1;
      int no;
      struct node *start = NULL;
      while(ch != 0)
      {
            printf("\n 1. Insert Node..\n 2. Reverse node..\n 3. Display 
Node..\n");
            printf("\nChoose any Function in Above..");
            scanf("%d",&ch);
            if(ch == 1)
            {
                  printf("\nEnter Number :");
                  scanf("%d",&no);
                  start = insert_ele_beg(start,no);
            }
            else if(ch == 2)
            {
                  start = reverse_ele(start);
            }
            else if(ch == 3)
            {
                display(start);
            }
            else if(ch == 0)
            {
                exit(0);
            }
        }
}
**********************************************************************
**********************************************************************
output:


 1. Insert Node..
 2. Reverse node..
 3. Display Node..

Choose any Function in Above..1

Enter Number :6

First Node Inserted..

 1. Insert Node..
 2. Reverse node..
 3. Display Node..

Choose any Function in Above..1

Enter Number :3

First Node Inserted..

 1. Insert Node..
 2. Reverse node..
 3. Display Node..

Choose any Function in Above..1

Enter Number :9

First Node Inserted..

 1. Insert Node..
 2. Reverse node..
 3. Display Node..

Choose any Function in Above..1

Enter Number :5

First Node Inserted..

 1. Insert Node..
 2. Reverse node..
 3. Display Node..

Choose any Function in Above..1

Enter Number :8

First Node Inserted..

 1. Insert Node..
 2. Reverse node..
 3. Display Node..

Choose any Function in Above..3

===========================================
        ** Display Information **
===========================================

No = 8
No = 5
No = 9
No = 3
No = 6


 1. Insert Node..
 2. Reverse node..
 3. Display Node..

Choose any Function in Above..2

===========================================

 Linked List Reversed..
===========================================

 1. Insert Node..
 2. Reverse node..
 3. Display Node..

Choose any Function in Above..3

===========================================
        ** Display Information **
===========================================

No = 6
No = 3
No = 9
No = 5
No = 8
```

```
 1. Insert Node..
 2. Reverse node..
 3. Display Node..

Choose any Function in Above..
```

```
*************************************************************************
*************************************************************************
6 .  Write a program to calculate the summation of all elements of
     the linked list.
*************************************************************************
*************************************************************************
```

```c
#include<stdio.h>
#include<conio.h>
struct student
{
      int rollno;
      float mark;
      struct student *link;
};
struct student* insertbeg6(struct student *start,int rlno,float marks)
{
      struct student *temp;
      temp=(struct student*)malloc(sizeof(struct student));
      temp->rollno=rlno;
      temp->mark=marks;
      temp->link=NULL;
      if(start!=NULL)
            temp->link=start;
      start=temp;
      printf("\n First Node inserted Successfully\n");
      return start;
}
int summation(struct student *start)
{
      int ans=0,num;
      struct student *temp;
      temp=start;
      while(temp!=NULL)
      {
            num=temp->rollno;
            ans=ans+num;
            temp=temp->link;
      }
      return ans;
}
void displayll6(struct student *start)
{
      struct student *temp;

      if(start==NULL)
            printf("\n LinkList is empty\n");
      else
      {
            printf("\n Displaying Information");
            temp=start;
            while(temp!=NULL)
            {
                  printf("\n Rollno:%d",temp->rollno);
                  printf("\t Mark:%.2f",temp->mark);
                  temp=temp->link;
            }
            printf("\n");
      }
}
void main()
{
      int ch=1,rlno;
      float marks;
      struct student *start=NULL;
      while(ch!=0)
      {
            printf("\n1.Insert");
            printf("\n2.Sum of all Elements of linklist");
            printf("\n3.Display");
            printf("\n0.exit\n");
            printf("\n Press any of the above key:");
            scanf("%d",&ch);
            if(ch==1)
            {
                  printf("\n Enter Rollno:");
                  scanf("%d",&rlno);
                  printf("\n Enter Mark:");
                  scanf("%f",&marks);
                  start=insertbeg6(start,rlno,marks);
            }
            else if(ch==2)
            {
                  printf("\n Sum of rollno is %d",summation(start));
            }

            else if(ch==3)
                        displayll6(start);
            else if(ch==0)
                  exit(0);
      }
}

**************************************************************************
**************************************************************************
output:

1.Insert
2.Sum of all Elements of linklist
3.Display
0.exit

 Press any of the above key:1
 Enter Rollno:1

 Enter Mark:2

 First Node inserted Successfully

1.Insert
2.Sum of all Elements of linklist
3.Display
0.exit

 Press any of the above key:1

 Enter Rollno:2

 Enter Mark:3.456

 First Node inserted Successfully


 Displaying Information
 Rollno:2        Mark:3.46
 Rollno:1        Mark:2.00

1.Insert
2.Sum of all Elements of linklist
3.Display
0.exit

 Press any of the above key:1

 Enter Rollno:4

 Enter Mark:5.678

 First Node inserted Successfully

1.Insert
2.Sum of all Elements of linklist
3.Display
0.exit

 Press any of the above key:1

 Enter Rollno:10

 Enter Mark:2.34

 First Node inserted Successfully

1.Insert
2.Sum of all Elements of linklist
3.Display
0.exit
```

```
 Press any of the above key:2

 Sum of rollno is 17
1.Insert
2.Sum of all Elements of linklist
3.Display
0.exit

 Press any of the above key:1

 Enter Rollno:3

 Enter Mark:4

 First Node inserted Successfully

1.Insert
2.Sum of all Elements of linklist
3.Display
0.exit

 Press any of the above key:2

 Sum of rollno is 20
1.Insert
2.Sum of all Elements of linklist
3.Display
0.exit

 Press any of the above key:0
Press any key to continue . . .

*************************************************************************
*************************************************************************
7 . Write a program to create two linked list and append the second list
after the first.
*************************************************************************
*************************************************************************
#include<stdio.h>
#include<conio.h>
struct student
{
      int rollno;
      float mark;
      struct student *link;
};
struct student* insertbeg7(struct student *start,int rlno,float marks)
{
      struct student *temp;
      temp=(struct student*)malloc(sizeof(struct student));
      temp->rollno=rlno;
      temp->mark=marks;
      temp->link=NULL;
	if(start!=NULL)
		temp->link=start;
	start=temp;
	return start;
}
void displayll7(struct student *start)
{
	struct student *temp;

	if(start==NULL)
		printf("\n LinkList is empty\n");
	else
	{
		//printf("\n Displaying Information");
		temp=start;
		while(temp!=NULL)
		{
			printf("\n Rollno:%d",temp->rollno);
			printf("\t Mark:%.2f",temp->mark);
			temp=temp->link;
		}
		printf("\n");
	}
}
void merge7(struct student *start,struct student *start2)
{
	struct student *temp;
	temp=start;
	while(temp->link!=NULL)
		temp=temp->link;
	temp->link=start2;
}
struct student* removell(struct student *start2)
{
	struct student *temp;
	struct student *p;
	temp=start2;
	while(temp!=NULL)
	{
		p=temp->link;
		free(temp);
		temp=p;
	}
	return NULL;
}
void main()
{
	int ch=1,rlno,n1,n2,i;
	float marks;
	struct student *start=NULL;
	struct student *start2=NULL;
	printf("\n Enter the Number of elements to be entered in first 
LinkList:");
	scanf("%d",&n1);
```

```
      for(i=0;i<n1;i++)
      {
            printf("\n Enter rollno%d:",i+1);
            scanf("%d",&rlno);
            printf("\n Enter Marks:",i+1);
            scanf("%f",&marks);
            start=insertbeg7(start,rlno,marks);
      }
      printf("\n Linklist1 Nodes Inserted Successfully\n");
      printf("\n Enter the Number of elements to be entered in second 
LinkList:");
      scanf("%d",&n2);
      for(i=0;i<n2;i++)
      {
            printf("\n Enter rollno%d:",i+1);
            scanf("%d",&rlno);
            printf("\n Enter Marks:",i+1);
            scanf("%f",&marks);
            start2=insertbeg7(start2,rlno,marks);
      }
      printf("\n Linklist2 Nodes Inserted Successfully\n");
      printf("\n Displaying Elements of Linklist1");
      displayll7(start);
      printf("\n Displaying Elements of Linklist2");
      displayll7(start2);
      merge7(start,start2);
      printf("\n After Appending Linklist2 to Linkilist1");
      printf("\n Displaying Elements of Linklist1");
      displayll7(start);
      start2=removell(start2);
      printf("\n Displaying Elements of Linklist2");
      displayll7(start2);
}

**************************************************************************
**************************************************************************
output:

 Enter the Number of elements to be entered in first LinkList:4

 Enter rollno1:1

 Enter Marks:2

 Enter rollno2:3

 Enter Marks:4

 Enter rollno3:5

 Enter Marks:6

 Enter rollno4:7
 Enter Marks:8

 Linklist1 Nodes Inserted Successfully

 Enter the Number of elements to be entered in second LinkList:2

 Enter rollno1:1

 Enter Marks:2

 Enter rollno2:3

 Enter Marks:4

 Linklist2 Nodes Inserted Successfully

 Displaying Elements of Linklist1
 Rollno:7        Mark:8.00
 Rollno:5        Mark:6.00
 Rollno:3        Mark:4.00
 Rollno:1        Mark:2.00

 Displaying Elements of Linklist2
 Rollno:3        Mark:4.00
 Rollno:1        Mark:2.00

 After Appending Linklist2 to Linkilist1
 Displaying Elements of Linklist1
 Rollno:7        Mark:8.00
 Rollno:5        Mark:6.00
 Rollno:3        Mark:4.00
 Rollno:1        Mark:2.00
 Rollno:3        Mark:4.00
 Rollno:1        Mark:2.00

 Displaying Elements of Linklist2
 LinkList is empty
Press any key to continue . . .
```

****
****

8) Write a program to swap two consecutive elements of the given linked list. (Swap only values)

****
****

```
#include<stdio.h>values
#include<stdlib.h>
struct student {
      int r_no;
      float marks;
      struct student *next;
};
```

```
void display(struct student *first){
      struct student *temp =first;
      if(first ==  NULL){
            printf("Linkedlist is empty:\n\n");
      }
      else{
            printf("\n========================\n");
            printf("NO \t\t Marks:\n");
            printf("========================\n");
            while(temp != NULL){
                  printf("%d\t\t",temp->r_no);
                  printf("%f\n",temp->marks);
                  temp = temp->next;
            }
            printf("========================\n");
      }
}


void insert(struct student **first,struct student **last){
      struct student *temp = (struct student *)malloc(sizeof(struct
student));
      struct student *p;
      printf("Enter the Roll No.\n");
      scanf("%d",&temp->r_no);
      printf("Enter the marks:\n");
      scanf("%f",&temp->marks);
      temp->next=NULL;
            if(*first == NULL){
                  *first = temp ;
                  *last = temp;
            }
            else{
                  (*last)->next = temp ;
                  *last = temp ;
            }
}


void pairswap(struct student *first){
      struct student *temp = first;
      float swp_m; // temporary var.
      int swp_r;
      while(temp != NULL && temp->next != NULL){
            swp_r = temp->r_no;
            swp_m = temp->marks;
            temp->marks=temp->next->marks;
            temp->r_no= temp->next->r_no;
            temp->next->marks = swp_m;
            temp->next->r_no = swp_r;
            temp = temp->next->next;
      }
      printf("\nNode swapping succesfully:\n");
}
```

```
int main(){
      struct student *first;
      struct student *last;
      int no=1,n,i;
      while(no != 0){
            printf("\n1-> Insert the list:\n");
            printf("2-> swap the marks of the list:\n");
            printf("3-> Display the list before swaping the marks:\n");
            printf("4-> Display the list after swaping the marks:\n");
            printf("5-> Exit\n");
            printf("\nEnter the No. What you want to do from above:\n");
            scanf("%d",&no);
            if(no == 0){
                  exit(1);
            }
            if(no == 1 ){
                  printf("ENter how many nodes you want to enter :\n");
                  scanf("%d",&n);
                  for(i=0;i<n;i++){
                        insert(&first,&last);
                  }
            }
            if(no == 2){
                  pairswap(first);
            }
            if(no == 3){
                  printf("Before swaping the marks of two consicutive 
nodes :");
                  display(first);
            }
            if(no == 4){
                  printf("After swaping the marks of two consicutive nodes 
:");
                  display(first);
            }
      }
}

*************************************************************************
*************************************************************************
oUTPUT:
*************************************************************************
*************************************************************************
1-> Insert the list:
2-> swap the marks of the list:
3-> Display the list before swaping the marks:
4-> Display the list after swaping the marks:
5-> Exit

Enter the No. What you want to do from above:
1
ENter how many nodes you want to enter :
4
```

```
Enter the Roll No.
1
Enter the marks:
56
Enter the Roll No.
2
Enter the marks:
48
Enter the Roll No.
3
Enter the marks:
68
Enter the Roll No.
4
Enter the marks:
98

1-> Insert the list:
2-> swap the marks of the list:
3-> Display the list before swaping the marks:
4-> Display the list after swaping the marks:
5-> Exit

Enter the No. What you want to do from above:
3
Before swaping the marks of two consicutive nodes :
========================
NO               Marks:
========================
1               56.000000
2               48.000000
3               68.000000
4               98.000000
========================

1-> Insert the list:
2-> swap the marks of the list:
3-> Display the list before swaping the marks:
4-> Display the list after swaping the marks:
5-> Exit

Enter the No. What you want to do from above:
2

Node swapping succesfully:

1-> Insert the list:
2-> swap the marks of the list:
3-> Display the list before swaping the marks:
4-> Display the list after swaping the marks:
5-> Exit

Enter the No. What you want to do from above:
4
```

```
After swaping the marks of two consicutive nodes :
========================
NO               Marks:
========================
2               48.000000
1               56.000000
4               98.000000
3               68.000000
========================

1-> Insert the list:
2-> swap the marks of the list:
3-> Display the list before swaping the marks:
4-> Display the list after swaping the marks:
5-> Exit

Enter the No. What you want to do from above:
0

--------------------------------
Process exited after 45.72 seconds with return value 1
Press any key to continue . . .
```

```
*************************************************************************
*************************************************************************
```

9 . Write a program to swap two consecutive elements of the given linked list. (Swap only addresses)

```
*************************************************************************
*************************************************************************
#include<stdio.h>
#include<stdlib.h>
struct student {
      int r_no;
      float marks;
      struct student *next;
};


void display(struct student *first){
      struct student *temp =first;
      if(first ==  NULL){
            printf("Linkedlist is empty:\n\n");
      }
      else{
            printf("\n========================\n");
            printf("NO \t\t Marks:\n");
            printf("========================\n");
            while(temp != NULL){
                  printf("%d\t\t",temp->r_no);
                  printf("%f\n",temp->marks);
                  temp = temp->next;
            }
            printf("========================\n");
      }
```

```
}


void insert(struct student **first,struct student **last){
      struct student *temp = (struct student *)malloc(sizeof(struct
student));
      struct student *p;
      printf("Enter the Roll No.\n");
      scanf("%d",&temp->r_no);
      printf("Enter the marks:\n");
      scanf("%f",&temp->marks);
      printf("temp->%d",temp);
      temp->next=NULL;
            if(*first == NULL){
                  *first = temp ;
                  *last = temp;
            }
            else{
                  (*last)->next = temp ;
                  *last = temp ;
            }
}

struct student* swap(struct student *first, int r1, int r2)
{

      struct student *temp = NULL;
      struct student *prevX = NULL;
      struct student *X = first;
      struct student *prevY = NULL;
      struct student *Y = first;
   // Nothing to do if x and y are same
   if (r1 == r2)
   return(first);

   // Search for x (keep track of prevX and CurrX

   while (X && X->r_no != r1)
   {
       prevX = X;
       X = X->next;
   }

   // Search for y (keep track of prevY and CurrY
   while (Y && Y->r_no != r2)
   {
       prevY = Y;
       Y = Y->next;
   }

   // If either x or y is not present, nothing to do
   if (X == NULL || Y == NULL)
       return(first);
   // If x is not head of linked list
   if (prevX != NULL)
       prevX->next = Y;
   else // Else make y as new head
       first = Y;

   // If y is not head of linked list
   if (prevY != NULL)
       prevY->next = X;
   else  // Else make x as new head
       first = X;

   // Swap next pointers
   temp = Y->next;
   Y->next = X->next;
   X->next  = temp;

   return(first);
}


int main(){
      struct student *first;
      struct student *last;
      int no=1,n,i,r1=0,r2=0;
      while(no != 0){
            printf("\n1-> Insert the list:\n");
            printf("2-> swap the marks of the list:\n");
            printf("3-> Display the list before swaping the marks:\n");
            printf("4-> Display the list after swaping the marks:\n");
            printf("5-> Exit\n");
            printf("\nEnter the No. What you want to do from above:\n");
            scanf("%d",&no);
            if(no == 0){
                  exit(1);
            }
            if(no == 1 ){
                  printf("ENter how many nodes you want to enter :\n");
                  scanf("%d",&n);
                  for(i=0;i<n;i++){
                        insert(&first,&last);
                  }
            }
            if(no == 2){
                  printf("Enter the Roll no 1:\n");
                  scanf("%d",&r1);
                  printf("Enter the Roll no 2:\n");
                  scanf("%d",&r2);
                  first = swap(first,r1,r2);
            }
            if(no == 3){
                  printf("Before swaping the marks of two consicutive
nodes :");
                  display(first);
```

```
          }
          if(no == 4){
               printf("After swaping the marks of two consicutive nodes
:");
               display(first);
          }
     }
}

*****************************************************************
*****************************************************************
OUTPUT:
*****************************************************************
*****************************************************************
1-> Insert the list:
2-> swap two nodes of the list:
3-> Display the list :
4-> Exit

Enter the No. What you want to do from above:

1
ENter how many nodes you want to enter :
4
Enter the Roll No.
1
Enter the marks:
65
Enter the Roll No.
2
Enter the marks:
45
Enter the Roll No.
3
Enter the marks:
68
Enter the Roll No.
4
Enter the marks:
98

1-> Insert the list:
2-> swap two nodes of the list:
3-> Display the list :
4-> Exit

Enter the No. What you want to do from above:
2
Enter the Roll no 1:
2
Enter the Roll no 2:
3

1-> Insert the list:
```

```
2-> swap two nodes of the list:
3-> Display the list :
4-> Exit

Enter the No. What you want to do from above:
3
After swapping the two nodes:
========================
NO               Marks:
========================
1               65.000000
3               68.000000
2               45.000000
4               98.000000
========================

1-> Insert the list:
2-> swap two nodes of the list:
3-> Display the list :
4-> Exit

Enter the No. What you want to do from above:
2
Enter the Roll no 1:
1
Enter the Roll no 2:
4

1-> Insert the list:
2-> swap two nodes of the list:
3-> Display the list :
4-> Exit

Enter the No. What you want to do from above:
3
After swapping the two nodes:
========================
NO               Marks:
========================
4               98.000000
3               68.000000
2               45.000000
1               65.000000
========================


1-> Insert the list:
2-> swap two nodes of the list:
3-> Display the list :
4-> Exit


Enter the No. What you want to do from above:
0
```

```
--------------------------------
Process exited after 66.55 seconds with return value 1
Press any key to continue . . .
```

```
*************************************************************************
*************************************************************************
10. Write a C program to split a given linked list into two.
*************************************************************************
*************************************************************************
```

```
#include<stdio.h>
#include<stdlib.h>
struct student {
      int r_no;
      float marks;
      struct student *next;
};


void insert(struct student **first,struct student **last){
      struct student *temp = (struct student *)malloc(sizeof(struct 
student));
      struct student *p;
      printf("Enter the Roll No.\n");
      scanf("%d",&temp->r_no);
      printf("Enter the marks:\n");
      scanf("%f",&temp->marks);
      temp->next=NULL;
            if(*first == NULL){
                  *first = temp ;
                  *last = temp;
            }
            else{
                  (*last)->next = temp ;
                  *last = temp ;
            }
}


void display(struct student *first){
      struct student *temp =first;
      if(first ==  NULL){
            printf("Linkedlist is empty:\n\n");
      }
      else{
            printf("\n=======================\n");
            printf("NO \t\t Marks:\n");
            printf("=======================\n");
            while(temp != NULL){
                  printf("%d\t\t",temp->r_no);
                  printf("%f\n",temp->marks);
                  temp = temp->next;
            }
            printf("=======================\n");
```

```
        }
}

struct student *split(struct student *first, int r_no){
        struct student *temp = first;
        struct student *prev = NULL;
        struct student *curr = NULL;
        struct student *new_list = NULL;
        while(temp != NULL && temp->r_no != r_no){
                prev = temp;
                temp = temp->next;
        }
        if(prev != NULL){
                new_list = temp ;
                prev->next = NULL;
        }
        return new_list;
}
int main(){
        struct student *first = NULL;
        struct student *last = NULL;
        struct student *second= NULL;
        int no=1,n,i,num;
        while(no != 0){
                printf("\n1-> Insert the list:\n");
                printf("2-> Split the linked list\n");
                printf("3-> Display the first linked list:\n");
                printf("4-> Display the second linked list:\n");
                printf("5-> Exit\n");
                printf("\nEnter the No. What you want to do from above:\n");
                scanf("%d",&no);
                if(no == 0){
                        exit(1);
                }
                if(no == 1 ){
                        printf("ENter how many nodes you want to enter :\n");
                        scanf("%d",&n);
                        for(i=0;i<n;i++){
                                insert(&first,&last);
                        }
                }
                if(no == 2){
                        printf("Enter no that you want to swap that node with
its next one:\n");
                        scanf("%d",&num);
                        second = split(first,num);
                }
                if(no == 3){
                        printf("Before swaping the marks of two consicutive
nodes :");
                        display(first);
                }
                if(no == 4){
                  printf("After swaping the marks of two consicutive nodes
:");
                  display(second);
            }
      }
}


**************************************************************************
**************************************************************************
OUTPUT:
**************************************************************************
**************************************************************************
1-> Insert the list:
2-> Split the linked list
3-> Display the first linked list:
4-> Display the second linked list:
5-> Exit

Enter the No. What you want to do from above:
1
ENter how many nodes you want to enter :
5
Enter the Roll No.
1
Enter the marks:
45
Enter the Roll No.
2
Enter the marks:
65
Enter the Roll No.
3
Enter the marks:
75
Enter the Roll No.
4
Enter the marks:
68
Enter the Roll No.
5
Enter the marks:
86

1-> Insert the list:
2-> Split the linked list
3-> Display the first linked list:
4-> Display the second linked list:
5-> Exit

Enter the No. What you want to do from above:
3
first linked list :
========================
NO               Marks:
========================
1               45.000000
2               65.000000
3               75.000000
4               68.000000
5               86.000000
========================

1-> Insert the list:
2-> Split the linked list
3-> Display the first linked list:
4-> Display the second linked list:
5-> Exit

Enter the No. What you want to do from above:
2
Enter no that you want to split the node:
3

1-> Insert the list:
2-> Split the linked list
3-> Display the first linked list:
4-> Display the second linked list:
5-> Exit

Enter the No. What you want to do from above:
3
first linked list :
========================
NO               Marks:
========================
1               45.000000
2               65.000000
========================

1-> Insert the list:
2-> Split the linked list
3-> Display the first linked list:
4-> Display the second linked list:
5-> Exit

Enter the No. What you want to do from above:
4
second linked list :
========================
NO               Marks:
========================
3               75.000000
4               68.000000
5               86.000000
========================

1-> Insert the list:
```

```
2-> Split the linked list
3-> Display the first linked list:
4-> Display the second linked list:
5-> Exit

Enter the No. What you want to do from above:
0

--------------------------------
Process exited after 29.14 seconds with return value 1
Press any key to continue . . .
```

****
****

ASSIGNMENT - 4

****
****

1 . Write a program to read a line from input file and print alternate characters in the output file. Display appropriate message for file i/o errors.

****
****

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
      int ch = 1;
      char mystring[100];
      char *fp;
      FILE *f1 = fopen("alternate.txt","w");
      if(f1)
      {
            //printf("open");
            printf("Enter String :");
            scanf("%s",mystring);
            fp = mystring;
            while(*fp != '\0')
            {
                  printf("%c",*fp);
                  if(ch == 1)
                  {
                        fprintf(f1,"%c ",*fp);
                        ch = 0;
                  }
                  else
                  {
                        ch = 1;
                  }
                  fp++;
            }
            //printf("%s",mystring);
      }
      else
      {
            printf("NOT FOUND");
      }
}

*************************************************************************
*************************************************************************
output:

Enter String :Preksha
PrekshaPress any key to continue . . .


IN ALTERNATE FILE :

P e s a
*************************************************************************
*************************************************************************
```

2 . Write a program to copy the contents of one file to another and also print the no. of lines in the first file.

```
*************************************************************************
*************************************************************************
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main1()
{
      int ch = 1;
      char mystring[100];
      char *fp;
      FILE *f1 = fopen("alternate.txt","w");
      if(f1 != '\0')
      {
            //printf("open");
            printf("Enter String :");
            scanf("%s",mystring);
            fp = mystring;
            while(*fp != '\0')
            {
                  printf("%c",*fp);
                  if(ch == 1)
                  {
                        fprintf(f1,"%c ",*fp);
                        ch = 0;
                  }
                  else
                  {
                        ch = 1;
                  }
                  fp++;
            }
            //printf("%s",mystring);
      }
      else
      {
            printf("asd");
      }
}
*************************************************************************
*************************************************************************
output:

FName : preksha
SName : sheth
Class : MCA2
3 lines..

Press any key to continue . . .



//in copy file:

FName : preksha
SName : sheth
Class : MCA2

*************************************************************************
*************************************************************************
3 .Write a program to search a particular word in an existing file and
display the no. of occurrences and the position
    of first occurrence  of that word. If the word is not found display
the appropriate message.
*************************************************************************
*************************************************************************
#include<stdio.h>
#include<string.h>
void main(){
      FILE *fp;
      char data[100],file_name[10],search_data[10],*ptr;
      int count=0;
      printf("Enter the file name: ");
      scanf("%s",file_name);
      fp=fopen(file_name,"w");
      if(fp==NULL)
            printf("No able to open the file");
      else{
            printf("Enter data: \n");
            getchar();
            gets(data);
            fputs(data,fp);
      }
      fclose(fp);
      fp=fopen(file_name,"r");
      if(fp==NULL)
            printf("No able to open the file");
      else{
            printf("Enter word you want to search: ");
            scanf("%s",search_data);
            fgets(data,100,fp);
            ptr=(strstr(data,search_data));
            while(ptr!=NULL){
                  count++;
                  ptr=strstr(ptr+1,search_data);
            }
            printf("File data is : %s\n",data);
            if(count>0){
                  printf("Word is not in the file :\n");
                  printf("Total number of word occurance in string is 
%d\n",count);
            }
            else
                  printf("Search not found\n");
      }

      fclose(fp);
}
```

```
*************************************************************************
*************************************************************************
```

```
Output:
Enter the file name: file.txt
Enter data:
department of computer science rollwala computer center
Enter word you want to search: computer
File data is : department of computer science rollwala computer center
Search found
Total number of word occurance in string is 2
Press any key to continue . . .

Enter the file name: file.txt
Enter data:
department of computer science rollwala computer center
Enter word you want to search: rcc
File data is : department of computer science rollwala computer center
Word is not in the file :
Press any key to continue . . .
```

```
*************************************************************************
*************************************************************************
```

Q-4)The files DATA1 and DATA2 contain sorted list of integers. Write a program to produce a third file DATA which holds a single sorted merged list of these two lists.

```
*************************************************************************
*************************************************************************
```

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

```
void merge(int [], int, int [], int, int[]);

void main(){
      FILE *fp_array1,*fp_array2,*fp_merger_sort_arrray;
      int arr1[10],arr2[10],arr3[30],i,limit_arr1,limit_arr2;
      char file_arr1[20],file_arr2[20],file_merge_sort_array[20];

      printf("Enter file_arr1 name:- ");
      scanf("%s",file_arr1);
      printf("Enter file_arr2 name:- ");
      scanf("%s",file_arr2);
      printf("Enter file_merge_sort_array name:- ");
      scanf("%s",file_merge_sort_array);
      printf("Enter the limit of the array1:- ");
      scanf("%d",&limit_arr1);
      printf("Enter the limit of the array2:- ");
      scanf("%d",&limit_arr2);

      fp_array1=fopen(file_arr1,"w");
      printf("Enter %d numbers in %s :\n",limit_arr1,file_arr1);
      for(i=0;i<limit_arr1;i++){
            scanf("%d",&arr1[i]);
            putw(arr1[i],fp_array1);
      }
      fclose(fp_array1);

      fp_array2=fopen(file_arr2,"w");
      printf("Enter %d numbers in %s :\n",limit_arr2,file_arr2);
      for(i=0;i<limit_arr2;i++){
            scanf("%d",&arr2[i]);
            putw(arr2[i],fp_array2);
      }
      fclose(fp_array2);

      i=0;
      fp_array1=fopen(file_arr1,"r");
      printf("%s data:\n",file_arr1);
      while((arr1[i]=getw(fp_array1))!=EOF){
            printf("%d\n",arr1[i]);
            i++;
      }
      fclose(fp_array1);

      i=0;
      fp_array2=fopen(file_arr2,"r");
      printf("%s data:\n",file_arr2);
      while((arr2[i]=getw(fp_array2))!=EOF){
            printf("%d\n",arr2[i]);
            i++;
      }
      fclose(fp_array2);

      merge(arr1,limit_arr1,arr2,limit_arr2,arr3);
      fp_merger_sort_arrray=fopen(file_merge_sort_array,"w");
      printf("Data in %s:\n",file_merge_sort_array);
      for(i=0;i<limit_arr1+limit_arr2;i++){
            putw(arr3[i],fp_merger_sort_arrray);
      }
      fclose(fp_merger_sort_arrray);
      for(i=0;i<limit_arr1+limit_arr2;i++){
            printf("%d\n",arr3[i]);
      }
}
void merge(int arr1[], int limit_arr1, int arr2[], int limit_arr2, int
arr3[]){
      int i,j,k;
      j=k=0;
      for(i=0;i<limit_arr1+limit_arr2;){
            if(j<limit_arr1 && k< limit_arr2){
                  if(arr1[j]<arr2[k]){
                        arr3[i]=arr1[j];
                        j++;
                  }
                  else{
                        arr3[i]=arr2[k];
                        k++;
                  }
                  i++;
            }
            else if(j==limit_arr1){
                  for(;i<limit_arr1+limit_arr2;){
                        arr3[i]=arr2[k];
                        k++;
                        i++;
                  }
            }
            else{
                  for(;i<limit_arr1+limit_arr2;){
                        arr3[i]=arr2[j];
                        j++;
                        i++;
                  }
            }
      }
}
```

```
*************************************************************************
*************************************************************************
Output:
Enter file_arr1 name:- arr1.txt
Enter file_arr2 name:- arr2.txt
Enter file_merge_sort_array name:- sort_arr.txt
Enter the limit of the array1:- 4
Enter the limit of the array2:- 4
Enter 4 numbers in arr1.txt :
1
```

```
6
5
8
Enter 4 numbers in arr2.txt :
2
3
4
7
arr1.txt data:
1
6
5
8
arr2.txt data:
2
3
4
7
Data in sort_arr.txt:
1
2
3
4
5
6
7
8
Press any key to continue . . .
```

****

****

5. Write a program to read line by line from a file and print all the repeated characters on the screen along with their frequency.

****

****

```
#include<stdio.h>
#include<string.h>
void main(){
      FILE *fp;
      char data[100],file_name[10],*ptr;
      int count=0,i,j,k;
      printf("Enter the file name: ");
      scanf("%s",file_name);
      fp=fopen(file_name,"w");
      if(fp==NULL)
            printf("No able to open the file");
      else{
            printf("Enter data: \n");
            getchar();
            gets(data);
            fputs(data,fp);
      }
      fclose(fp);
      fp=fopen(file_name,"r");
      if(fp==NULL)
            printf("No able to open the file");
      else{
            fgets(data,100,fp);
            for(i=0;data[i]!='\0';i++){
                  k=1;
                  for(j=i+1;data[j]!='\0';j++){
                        if(data[i]==data[j]){
                              k++;
                              data[j]='`';
                        }
                  }
                  if(data[i]!='`' && data[i]!=' ' && k>1)
                  printf("%c appeared %d times\n",data[i],k);
            }
      }
      fclose(fp);
}
```

```
****************************************************************************
****************************************************************************
Output:
Enter the file name: file.txt
Enter data:
rollwala computer center
r appeared 3 times
o appeared 2 times
l appeared 3 times
a appeared 2 times
c appeared 2 times
t appeared 2 times
e appeared 3 times
Press any key to continue . . .

****************************************************************************
****************************************************************************
```

6. Write a function to read a file and count the no. of characters, spaces, tabs, newlines and no. of words in a given text file.

```
****************************************************************************
****************************************************************************
#include<stdio.h>
void main(){
      FILE *fp;
      char file_name[20],data[100],ch;
      int
word_count=0,line_count=0,space_count=0,tab_count=0,character_count=0;
      printf("Enter the file name: ");
      scanf("%s",file_name);
      fp=fopen(file_name,"w");
      if(fp==NULL)
            printf("file could not be open\n");
      else{
            printf("Enter data:\n");
            getchar();
            gets(data);
            fputs(data,fp);
      }
      fclose(fp);
      fp=fopen(file_name,"r");
      if(fp==NULL)
            printf("file could not be open\n");
      else{
            while((ch=getc(fp))!=EOF){
                  if(ch!=NULL && ch!=' ' && ch!='\t' && ch!='.')
                        character_count++;
                  if(ch==' ' || ch=='\n' || ch=='\t' || ch=='.')
                        word_count++;
                  if(ch==' ')
                        space_count++;
                  if(ch=='\t')
                        tab_count++;
                  if(ch=='\n')
                        line_count++;
            }
            printf("\nNumber of Characters:- %d\n",character_count);
            printf("Number of Words:- %d\n",word_count);
            printf("Number of blank space:- %d\n",space_count);
            printf("Number of Tabs:- %d\n",tab_count);
            printf("Number of Lines:- %d\n",line_count);
      }
      fclose(fp);


}
*************************************************************************
*************************************************************************
Output:
Enter the file name: file.txt
Enter data:
rollwala computer center

Number of Characters:- 22
Number of Words:- 3
Number of blank space:- 3
Number of Tabs:- 0
Number of Lines:- 0
Press any key to continue . . .

*************************************************************************
*************************************************************************
7. Write a program to remove all the blank lines from a given file.
*************************************************************************
*************************************************************************
#include <stdio.h>
#include <conio.h>

int main()
{
    FILE *fp,*fp1;
    int p;

    fp=fopen("file.txt","r");
    fp1=fopen("newfile.txt","w");

    while((p=getc(fp))!=EOF)
    {
       fputc(p,fp1);
       if (p==10)
         {
           while((p=getc(fp))==10)
           {
           }
           fputc(p,fp1);
       }
     }
      printf("blanks removed successfully");
      fclose(fp);
      fclose(fp1);
      getch();
}
```

****
****

```
Output:
blanks removed successfully

------------
file.txt
------------
hello

world

user is dump
hello surat

------------
newfile.txt
------------
hello
world
user is dump
hello surat
```

****
****

8. Write a function to accept a string from the keyboard and remove all occurrences of that string from a given file.

****
****

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

```
void removestring(char *str,const char *toRemove)
{
      int i, j, stringLen, toRemoveLen;
    int found;

    stringLen   = strlen(str);      // Length of string
    toRemoveLen = strlen(toRemove); // Length of word to remove


    for(i=0; i <= stringLen - toRemoveLen; i++)
    {
        /* Match word with string */
        found = 1;
        for(j=0; j < toRemoveLen; j++)
        {
            if(str[i + j] != toRemove[j])
            {
                found = 0;
                break;
            }
        }

        /* If it is not a word */
        if(str[i + j] != ' ' && str[i + j] != '\t' && str[i + j] != '\n'
&& str[i + j] != '\0')
        {
            found = 0;
        }

        /*
         * If word is found then shift all characters to left
         * and decrement the string length
         */
        if(found == 1)
        {
            for(j=i; j <= stringLen - toRemoveLen; j++)
            {
                str[j] = str[j + toRemoveLen];
            }

            stringLen = stringLen - toRemoveLen;

            // We will match next occurrence of word from current index.
            i--;
        }
    }
}
void main()
{
    FILE *fp,*fp1;
    char word[1000],string[1000];
    fp=fopen("removestring.txt","r");
    fp1=fopen("remove.txt","w");
    if(fp && fp1){

        printf("enter the word you want to remove in file : ");
        scanf("%s",word);
        while(fgets(string,sizeof string,fp)){

            removestring(string,word);
            fputs(string,fp1);
        }

        printf("all the occurences of %s removed successfully",word);
        remove("removestring.txt");

    /* Rename temp file as original file */
    rename("remove.txt", "removestring.txt");


        }
        fclose(fp);
        fclose(fp1);

        getch();
    }
*************************************************************************
*************************************************************************
Output :

removestring.txt

hello this is a file
hello this is a program

=============
OUTPUT
=============


enter the word you want to remove in file : hello
all the occurences of hello removed successfully

remove.txt

this is a file
this is a program
*************************************************************************
*************************************************************************
9. Write a program a program to remove all the comments from a C file
*************************************************************************
*************************************************************************
#include <stdio.h>
#include <conio.h>
void check_comment(char c)
{
    char d;

    if( c == '/')
    {
        if((d=fgetc(fp))=='*')   //check the second letter after the / if
it * then calls block_comment() fun.
         block_comment();
        else if( d == '/')   // else calls single_comment() fun.
        {
          single_comment();
        }
        else
        {
            fputc(c,fp1);
            fputc(d,fp1);
        }
    }

    else
        fputc(c,fp1);
}

void block_comment()
{

 char d,e;

    while((d=fgetc(fp))!=EOF)
    {
      //scan character */ for ending the comment line .
        if(d=='*')
        {
            e=fgetc(fp);
                 if(e=='/')
                 return;
        }
   }

}



void single_comment()
{
 char d,e;

    while((d=fgetc(fp))!=EOF)
    {
      //checking the '/n' character for ending the single line comment
        if(d=='\n')
            return;

    }
```

```
}

FILE *fp , *fp1;

void main(void)
{
    char c;
    fp = fopen ("file.txt","r") ;
    fp1 = fopen ("newfile.txt","w") ;

    while((c=fgetc(fp))!=EOF)
        check_comment(c);

    fclose(fp);
    fclose(fp1);
    printf("comments removed successully");

    getch();
}


*************************************************************************
*************************************************************************
Output :

comments removed successully

------------
file.txt
------------
Rollwala computer center
/*
hello
world
*/

------------
newfile.txt
------------
Rollwala computer center


*************************************************************************
*************************************************************************
10. Write a program that will generate a data file containing the list of
customers and their corresponding telephone numbers. Use a structure
variable to store the name and telephone number of each customer. Create
a data file using a sample list.
*************************************************************************
*************************************************************************
#include<stdio.h>

struct customer{
      char name[20];
      long long unsigned int mob_num;
};

void main(){
      FILE *fp;
      struct customer c[10];
      char file_name[20];
      int i,limit;
      printf("Enter file name: ");
      scanf("%s",file_name);
      printf("Enter total number of records you want to enter in file:
");
      scanf("%d",&limit);

      fp=fopen(file_name,"wb");
      printf("Enter data in file: \n");
      for(i=0;i<limit;i++){
            printf("Enter customer name: ");
            scanf("%s",c[i].name);
            printf("Enter Mobile number: ");
            scanf("%lld",&c[i].mob_num);
      }
      fwrite(c,sizeof(struct customer),limit,fp);
      fclose(fp);

      fp=fopen(file_name,"rb");
      fread(c,sizeof(struct customer),limit,fp);
      for(i=0;i<limit;i++){
            printf("Customer name: %s\n",c[i].name);
            printf("Mobile number: %lld\n",c[i].mob_num);
      }
      fclose(fp);
}
*************************************************************************
*************************************************************************
Output:
Enter file name: file.txt
Enter total number of records you want to enter in file: 3
Enter data in file:
Enter customer name: preksha
Enter Mobile number: 9157189673
Enter customer name: prerak
Enter Mobile number: 1234567890
Enter customer name: dhruvin
Enter Mobile number: 9876543210

Customer name: preksha
Mobile number: 9157189673
Customer name: prerak
Mobile number: 1234567890
Customer name: dhruvin
Mobile number: 9876543210
Press any key to continue . . .
```

```
************************************************************************
************************************************************************
11. Write an interactive menu driven program that will access the data
file created in the above problem to do one of the following task:
           a.   Determine the telephone number of a specific customers
           b.   Determine the customer whose telephone no. is specified.
           c.   Add a new record.
           d.   Delete a record
           c.   Generate the listing of all the customers and their
telephone numbers

************************************************************************
************************************************************************
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>

struct customers
{
      int id;
      char name[60],telephone[10];

};

void getdata(char *fname,customers cu[],int n)
{
      int i;
      FILE *fp;
      fp=fopen(fname,"a");
      if(fp==NULL)
      {
            printf("\n Error in opening an file...");
            exit(0);
      }
      for(i=0;i<n;i++)
      {
            printf("\ Enter id:");
            scanf("%d",&cu[i].id);
            printf("\n Enter Name of Customer:");
            scanf(" %s",&cu[i].name);
            fflush(stdin);
            printf("\n Enter Telephone No:");
            scanf(" %s",&cu[i].telephone);
            fflush(stdin);
            fwrite(&cu[i],sizeof(cu[i]),1,fp);
      }

            fclose(fp);
}

void display(char *fname,customers cu[],int n)
{
      int i;
      FILE *fp;

      fp=fopen(fname,"r");
      fseek(fp,0L,0);
      if(fp==NULL)
      {
            printf      ("\n Error in opening an file...");
            exit(0);
      }

      for(int i=0;i<n;i++)
      {
            fread(&cu[i],sizeof(cu[i]),1,fp);

            printf("\n %d %s \t %s",cu[i].id,cu[i].name,cu[i].telephone);
      }
            fclose(fp);
}

void find_name(char *fname,customers cu[],int n,char name[])
{
      int i,found=0;
      FILE *fp;
      fp=fopen(fname,"r");
      if(fp==NULL)
      {
            printf("\n Error in opening an file...");
            exit(0);
      }

      for(int i=0;i<=n;i++)
      {
            fread(&cu[i],sizeof(cu[i]),1,fp);
            if((strcmp(cu[i].name,name))==0)
            {
                  found=1;
                  printf("\n %d \t %s \t %s
\n",cu[i].id,cu[i].name,cu[i].telephone);
            }
      }
      if(found==0)
      {
            printf("\n No Record exist.");
      }
            fclose(fp);

}
void find_tele(char *fname,customers cu[],int n,char tele[])
{
      int i,found=0;
      FILE *fp;
      fp=fopen(fname,"r");
      if(fp==NULL)
      {
            printf("\n Error in opening an file...");
            exit(0);
      }

      for(int i=0;i<=n;i++)
      {
            fread(&cu[i],sizeof(cu[i]),1,fp);
            if(strcmp(cu[i].telephone,tele)==0)
            {
                  found=1;

                  printf("\n %d \t %s \t %s
",cu[i].id,cu[i].name,cu[i].telephone);
            }
      }
      if(found==0)
      {
            printf("\n No Record exist.");
      }
            fclose(fp);

}
int remove(char *fname,customers e1[],int n)
{
      int i,empid,cnt=0;
      double sal;
      char name[20];
      FILE *fp,*fptr;

      fp=fopen(fname,"r");

      if(fp==NULL)
      {
            printf      ("\n Error in opening an file...");
            exit(0);
      }

      fptr=fopen("temp.txt","w");
      if(fp==NULL)
      {
            printf      ("\n Error in opening an file...");
            exit(0);
      }

      printf("\n Enter Name:");
      scanf("%s",&name);

      for(int i=0;i<n;i++)
      {
            fread(&e1[i],sizeof(e1[i]),1,fp);

            if(strcmp(e1[i].name,name)==0)
            {
                  cnt++;
```

```
        }
        else
        {
            fwrite(&e1[i], sizeof(e1[i]), 1, fptr);
        }
    }
    if(cnt>0)
        printf("\n Customer Deleted Successfully.");
    else
        printf("\n Customer Not Exist...!!");


    fclose(fp);
    fclose(fptr);

    fp=fopen(fname,"w");

    if(fp==NULL)
    {
        printf      ("\n Error in opening an file...");
        exit(0);
    }

    fptr=fopen("temp.txt","r");
    if(fp==NULL)
    {
        printf      ("\n Error in opening an file...");
        exit(0);
    }
    for(int i=0;i<n;i++)
    {
        fread(&e1[i],sizeof(e1[i]),1,fptr);
        fwrite(&e1[i], sizeof(e1[i]), 1, fp);
    }

        fclose(fp);
        fclose(fptr);
        if(cnt>0)
            return 1;
        else
            return 0;

}
void main()
{
    int n,ch,res;
    char file_name[80],choice='n',name[20],tele[10];
    static int cnt=0;
    struct customers cu[10];

    printf("\n Enter File name:");
    scanf("%s",&file_name);
    printf("\n --------------------------------------------------------
-----------------");
```

```
      printf("\n 1.Determine the telephone number of a specific 
customers");
      printf("\n 2.Determine the customer whose telephone no. is 
specified.");
      printf("\n 3.Add a new record.");
      printf("\n 4.Delete a record");
      printf("\n 5.Generate the listing of all the customers and their 
telephone numbees");
      printf("\n -------------------------------------------------------
-----------------");
      do
      {
            printf("\n Enter Your choice:");
            scanf("%d",&ch);
            switch(ch)
            {
                  case 1:
                              printf("\n Enter Name:");
                              scanf("%s",name);
                              find_name(file_name,cu,n,name);
                              break;
                  case 2:
                              printf("\n Enter Telephone No:");
                              scanf("%s",tele);
                              find_tele(file_name,cu,cnt,tele);
                              break;
                  case 3:
                              printf("\n Enter number of records for 
Add:");
                              scanf("%d",&n);
                              cnt=cnt+n;
                              getdata(file_name,cu,n);
                              break;
                  case 4:
                              res=remove(file_name,cu,cnt);
                              if(res)
                                    cnt=cnt-1;

                              break;
                  case 5:
                              printf("\n----------------");
                              printf("\n Display Records");
                              printf("\n-----------------");
                              display(file_name,cu,cnt);
                              break;
            }
            printf("\n Do You want to continue:");
            scanf("%s",&choice);


      }while(choice=='y');

      getch();
}
```

```
*************************************************************************
*************************************************************************



 Output:

 Enter File name:file.txt

 ------------------------------------------------------------------------
-
 1.Determine the telephone number of a specific customers
 2.Determine the customer whose telephone no. is specified.
 3.Add a new record.
 4.Delete a record
 5.Generate the listing of all the customers and their  telephone numbees
 ------------------------------------------------------------------------
-
 Enter Your choice:3

 Enter number of records for Add:2
 Enter id:101

 Enter Name of Customer:prerak

 Enter Telephone No:9157189673

 Enter id:102

 Enter Name of Customer:dhruvin

 Enter Telephone No:9876543210

 Do You want to continue:y

 Enter Your choice:5

----------------
 Display Records
-----------------
 101 prerak       9157189673
 102 dhruvin       9876543210
 Do You want to continue:y

 Enter Your choice:1

 Enter Name:dhruvin

 102     dhruvin   9876543210

 Do You want to continue:y
 Enter Your choice:2

 Enter Telephone No:9157189673

 101     prerak   9157189673
 Do You want to continue:y

 Enter Your choice:4

 Enter Name:dhruvin

 Customer Deleted Successfully.
 Do You want to continue:y

 Enter Your choice:5

----------------
 Display Records
-----------------
 101 prerak       9157189673
 Do You want to continue:n
```

```
*************************************************************************
*************************************************************************
12. Use a structure of Employee to write records of employee to a file.
Include a menu that will allow the user to select any of the following
features
            a.  Add a new record.
            b.  Delete a record.
            c.  Modify an existing record.
            d.  Retrieve and display an entire record for a given name.
            e.  Generate a complete list of all names, addresses and
                telephone numbers.
            f.  End of the computation.
*************************************************************************
*************************************************************************
```

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

struct employee{
      int id;
      char name[20],address[100];
};

void main(){
      FILE *fp,*temp_fp;
      struct employee e;
      char
file_name[20],ch,temp_file_name[20]="temp_file.txt",search_employee[20];
      int choice,found=0,eid,cur_pos;

      printf("Enter the file name: ");
```

```
    scanf("%s",file_name);

    do{
        printf("1. Add Record\n2. Delete Record\n3. Modify Existing
Record\n4. Search Employee\n5. Display all Employees\n6. Exit\n");
        scanf("%d",&choice);
        switch(choice){
            case 1:
                fp=fopen(file_name,"ab");
                do{
                    printf("Enter Employee id: ");
                    scanf("%d",&e.id);
                    printf("Enter Employee Name: ");
                    getchar();
                    scanf("%[^\n]",&e.name);
                    printf("Enter Employee Address: ");
                    getchar();
                    scanf("%[^\n]",&e.address);
                    fwrite(&e,sizeof(e),1,fp);
                    printf("press y to add more record else
press n :");

                    getchar();
                    scanf("%c",&ch);
                }while(ch!='n');
                fclose(fp);
                break;

            case 2:
                temp_fp=fopen(temp_file_name,"wb");
                fp=fopen(file_name,"rb");
                printf("Enter Employee Id: ");
                scanf("%d",&eid);
                while(fread(&e,sizeof(e),1,fp)!=NULL){
                    if(e.id==eid){
                        found=1;
                        printf("Data deleted successfully\n");
                    }
                    else{
                        fwrite(&e,sizeof(struct
employee),1,temp_fp);
                    }
                }
                fclose(fp);
                fclose(temp_fp);
                if(found!=1){
                    printf("Employee does not exists\n");
                }
                else{
                    remove(file_name);
                    rename(temp_file_name,file_name);
                }
                break;
            case 3:
                fp=fopen(file_name,"rb+");
```

```
                        printf("Enter Employee Id: ");
                        scanf("%d",&eid);
                        while(fread(&e,sizeof(e),1,fp)!=NULL){
                              if(eid==(e.id)){
                                    found=1;
                                    break;
                              }
                        }
                        if(found==1){
                              fseek(fp,sizeof(struct employee)*(eid-
1),SEEK_SET);
                              e.id=eid;
                              printf("Enter Employee Name: ");
                              getchar();
                              scanf("%[^\n]",&e.name);
                              printf("Enter Employee Address: ");
                              getchar();
                              scanf("%[^\n]",&e.address);
                              fwrite(&e,sizeof(e),1,fp);
                              fseek(fp,-(sizeof(struct employee)*(eid-
1)),SEEK_SET);
                              fread(&e,sizeof(e),1,fp);
                        }
                        else{
                              printf("Employee not found\n");
                        }
                        break;
                  case 4:
                        fp=fopen(file_name,"rb");
                        printf("Enter Employee name : ");
                        scanf("%s",search_employee);
                        while(fread(&e,sizeof(e),1,fp)!=NULL){
                              if(strcmp(e.name,search_employee)==0){
                                    printf("\nId: %d\n",e.id);
                                    printf("Name: %s\n",e.name);
                                    printf("Address: %s\n\n",e.address);
                                    break;
                              }
                        }
                        fclose(fp);
                        if(found!=1){
                              printf("search not found\n");
                        }
                        break;
                  case 5:
                        fp=fopen(file_name,"rb");
                        printf("Employee details:- \n\n");
                        while(fread(&e,sizeof(e),1,fp)!=NULL){
                              printf("Id: %d\n",e.id);
                              printf("Name: %s\n",e.name);
                              printf("Address: %s\n\n",e.address);
                        }
                        fclose(fp);
                        break;
```

```
                 case 6:
                       exit(0);
                 default:
                       printf("plase enter correct choice\n");
                       break;
           }
      }while(1);
}
*************************************************************************
*************************************************************************
```

Output:

```
1. Add Record
2. Delete Record
3. Modify Existing Record
4. Search Employee
5. Display all Employees
6. Exit
1
Enter Employee id: 1
Enter Employee Name: prerak
Enter Employee Address: motera
press y to add more record else press n :y
Enter Employee id: 2
Enter Employee Name: mahi
Enter Employee Address: chandkheda
press y to add more record else press n :n

1. Add Record
2. Delete Record
3. Modify Existing Record
4. Search Employee
5. Display all Employees
6. Exit
4
Enter Employee name :

Id: 2
Name: mahi
Address: chandkheda

1. Add Record
2. Delete Record
3. Modify Existing Record
4. Search Employee
5. Display all Employees
6. Exit
3
Enter Employee Id: 1
Enter Employee Name: prerak
Enter Employee Address: sahjanand society,motera

1. Add Record
```

```
2. Delete Record
3. Modify Existing Record
4. Search Employee
5. Display all Employees
6. Exit
5
Employee details:-

Id: 1
Name: prerak
Address: sahjanand society,motera

Id: 2
Name: mahi
Address: chandkheda

1. Add Record
2. Delete Record
3. Modify Existing Record
4. Search Employee
5. Display all Employees
6. Exit
2
Enter Employee Id: 1
Data deleted successfully

1. Add Record
2. Delete Record
3. Modify Existing Record
4. Search Employee
5. Display all Employees
6. Exit
5
Employee details:-

Id: 2
Name: mahi
Address: chandkheda

1. Add Record
2. Delete Record
3. Modify Existing Record
4. Search Employee
5. Display all Employees
6. Exit
6
Press any key to continue . . .
```

****
****

13. Write a program that will generate a data file containing the list of countries and their corresponding capitals.
 Place the name of each country and its corresponding capital in a separate structure.

Treat each structure as a separate record. Run the program, creating a data file for use in the next problem

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

struct Country
{
      int id;
      char name[60];
      struct capital
      {
            char cp_name[60];
      };
      struct capital cp;
};

void Insert(char *fname,Country c1[],int n)
{
      int i;
      FILE *fp;
      fp=fopen(fname,"a");
      if(fp==NULL)
      {
            printf("\n Error in opening an file...");
            exit(0);
      }
      for(i=0;i<n;i++)
      {
            printf("\n Enter Country id:");
            scanf("%d",&c1[i].id);

            fflush(stdin);
            printf("\n Enter Country Name:");
            scanf("%s",&c1[i].name);

            fflush(stdin);
            printf("\n Enter Capital Name:");
            scanf("%s",&c1[i].cp.cp_name);

            printf("\n %d \t %s \t %s
",c1[i].id,c1[i].name,c1[i].cp.cp_name);
            fwrite(&c1[i], sizeof(c1[i]), 1, fp);
      }
}

void display(char *fname,Country c1[],int n)
{
      int i;
      FILE *fp;
      fp=fopen(fname,"r");
      fseek(fp,0L,0);
      if(fp==NULL)
      {
            printf      ("\n Error in opening an file...");
            exit(0);
      }

      for(int i=0;i<n;i++)
      {
            fread(&c1[i],sizeof(c1[i]),1,fp);
            printf("\n %d \t %s \t %s \t
",c1[i].id,c1[i].name,c1[i].cp.cp_name);
      }
            fclose(fp);
}

void main()
{
      int n,ch,res;
      char file_name[80],choice='n',name[20],tele[10];
      static int cnt=0;
      struct Country c1[10];


      printf("\n Enter File name:");
      scanf("%s",&file_name);
      printf("\n 1.Add a new Record.");
      printf("\n 2.Display a Record. ");
      do
      {
            printf("\n Enter Your choice:");
            scanf("%d",&ch);
            switch(ch)
            {
                  case 1:printf("\n Enter number of records for Add:");
                              scanf("%d",&n);
                              cnt=cnt+n;
                              Insert(file_name,c1,n);
                              break;
                  case 2:
                              printf("\n----------------");
                              printf("\n Display Records");
                              printf("\n-----------------");
                              display(file_name,c1,cnt);
                              break;
            }
            printf("\n Do You want to continue:");
            scanf("%s",&choice);

      }while(choice=='y');

      getch();
}
```

```
*************************************************************************
*************************************************************************
```

Output :

```
Enter File name:file.txt


 1.Add a new Record.
 2.Display a Record.
 Enter Your choice:1

 Enter number of records for Add:3

 Enter Country id:101

 Enter Country Name:India

 Enter Capital Name:Delhi

 Enter Country id:102

 Enter Country Name:Afghanistan

 Enter Capital Name:Kabul

 Enter Country id:103

 Enter Country Name:Pakistan

 Enter Capital Name:Karanchi



 Do You want to continue:y

 Enter Your choice:2

----------------
 Display Records
-----------------
 101     India         Delhi
 102     Afghanistan   Kabul
 103   Pakistan  Karanchi
 Do You want to continue:n

*************************************************************************
*************************************************************************
```

14. Write an interactive, menu-driven C program that will access the data

file generated in the preceding problem and then allow one of the following operations to be executed:

a. Determine the capital of a specified country.
b. Determine the country whose capital is specified.
c. Terminate the computation.

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

struct Country
{
      int id;
      char name[60];
      struct capital
      {
            char cp_name[60];
      };
      struct capital cp;
};


void display(char *fname,Country c1[],int n)
{
      int i=0;
      FILE *fp;

      fp=fopen(fname,"r");

      if(fp==NULL)
      {
            printf      ("\n Error in opening an file...");
            exit(0);
      }
      fseek(fp, 0, SEEK_SET);

      while(fread(&c1[i],sizeof(c1[i]),1,fp))
      {
            printf("\n %d \t %s \t %s \t
",c1[i].id,c1[i].name,c1[i].cp.cp_name);
            i++;
      }
            fclose(fp);
}

void find_country(char *fname,Country c1[],int n)
{
      int i,cnt=0;
      char capi[60];
      FILE *fp;
```

```
    fp=fopen(fname,"r");
    fseek(fp,0L,0);
    if(fp==NULL)
    {
        printf    ("\n Error in opening an file...");
        exit(0);
    }
    printf("\n Enter Capital:");
    scanf("%s",capi);

    i=0;
    while(fread(&c1[i],sizeof(c1[i]),1,fp))
    {
        if(strcmp(capi,c1[i].cp.cp_name)==0)
        {
            cnt++;
            printf("\n %d \t %s \t %s \t
",c1[i].id,c1[i].name,c1[i].cp.cp_name);
        }
        i++;
    }
    if(cnt==0)
        printf("\n Record Doesnot Exist...!");

        fclose(fp);
}
void find_capital(char *fname,Country c1[],int n)
{
    int i,cnt=0;
    char cou[60];
    FILE *fp;

    fp=fopen(fname,"r");
    fseek(fp,0L,0);
    if(fp==NULL)
    {
        printf    ("\n Error in opening an file...");
        exit(0);
    }
    printf("\n Enter Country:");
    scanf("%s",cou);

    i=0;
    while(fread(&c1[i],sizeof(c1[i]),1,fp))
    {
        if(strcmp(cou,c1[i].name)==0)
        {
            cnt++;
            printf("\n %d \t %s \t %s \t
",c1[i].id,c1[i].name,c1[i].cp.cp_name);
        }
        i++;
    }
    if(cnt==0)
```

```
            printf("\n Record Doesnot Exist...!");

            fclose(fp);
}
void main()
{
      int n,ch,res;
      char file_name[80],choice='n',name[20],tele[10];
      static int cnt=0;
      struct Country c1[10];

      printf("\n Enter File name:");
      scanf("%s",&file_name);
      printf("\n --------------------");
      printf("\n 1.Display a Record.");
      printf("\n 2.Determine the capital of a specified ountry.");
      printf("\n 3.Determine the country whose capital is specified.");
      printf("\n --------------------");
      do
      {
            printf("\n Enter Your choice:");
            scanf("%d",&ch);
            switch(ch)
            {
                  case 1:printf("\n----------------");
                              printf("\n Display Records");
                              printf("\n-----------------");
                              display(file_name,c1,cnt);
                              break;
                  case 2:
                              find_capital(file_name,c1,cnt);
                              break;
                  case 3:
                              find_country(file_name,c1,cnt);
                              break;
            }
            printf("\n Do You want to continue:");
            scanf("%s",&choice);

      }while(choice=='y');

      getch();
}

**************************************************************************
**************************************************************************
```

Output :

```
 Enter File name:file.txt

 --------------------
 1.Display a Record.
 2.Determine the capital of a specified ountry.
 3.Determine the country whose capital is specified.
 --------------------
 Enter Your choice:1

----------------
 Display Records
-----------------
 101     India         Delhi
 102     Afghanistan   Kabul
 103   Pakistan  Karanchi
 Do You want to continue:y

 Enter Your choice:2

 Enter Country:India

 101     India         Delhi
 Do You want to continue:y

 Enter Your choice:3

 Enter Capital:Karanchi

 103   Pakistan  Karanchi
 Do You want to continue:n
```

```
*************************************************************************
*************************************************************************
16.  Write a C Program to build utilities for performing following tasks
      (Use Command Line Arguments)

      a. For computing the average of given numbers
      b. For computing factorial of given numbers
      c. List all the files in current directory  containing word
ROLLWALA.
      d. Rename given file.
      e. List all EXE files in a given diectory.
      f. Merge two files into third file.
*************************************************************************
*************************************************************************
a. For computing the average of given numbers
*************************************************************************
*************************************************************************
```

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


float avg(int argc, char **argv)
{
    int i;
    float average, total = 0;
    if (argc < 2) {
        printf("Enter atleast 1 number");
        return 0;
    }
    for (i = 1; i < argc; i++) {
        total = total + atoi(argv[i]);
    }

    average = total / (argc - 1);

    return average;
}


int main(int argc, char **argv)
{
    float avrg;

    avrg = avg(argc, argv);
    printf("Average: %.2f \n", avrg);

    return 0;
}
```

```
**************************************************************************
**************************************************************************
output :

E:\MCA-II\Advanced C\FILES\pro_temp_2\Debug>pro_16 10 20 30 40
    Average: 25.00


**************************************************************************
**************************************************************************
```

b. For computing factorial of given numbers

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


int* factorial(int argc, char **argv, int *arr)
{
    if (argc < 2) {
        printf("Enter atleast 1 number\n");
        exit(0);
    }

    int fact(int num)
    {
        if(num == 0)
            return 1;
        return num * fact(num - 1);
    }

    int i;
    for (i = 1; i < argc; i++) {
        arr[i - 1] = fact(atoi(argv[i]));
    }

    return arr;
}


int main(int argc, char **argv)
{
    int arr[10], i;
    int *ptr = factorial(argc, argv, arr);

    puts("Factorials are as follows:\n");
    for (i = 1; i < argc; i++)
        printf("%02d: %5d\n", atoi(argv[i]), ptr[i - 1]);

    return 0;
}

*************************************************************************
*************************************************************************
output:

E:\MCA-II\Advanced C\FILES\pro_temp_2\Debug>pro_16 5

    Factorials are as follows:

    05:   120


*************************************************************************
*************************************************************************
c. List all the files in current directory  containing word ROLLWALA.
*************************************************************************
*************************************************************************
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<dirent.h>

int findWordIn(char search_this_word[], char d_name[])
{
    int i = 0, length = 0, count = 0, finding = 0, first_occurance = 0;
    char c, search;
    FILE *f = fopen (d_name, "r");

    length = strlen (search_this_word);

    if (f) {
        if(!length) {
            exit(0);
        }
        search = search_this_word[0];

        while((c = fgetc(f)) != EOF){
            if (count == 0) {
                first_occurance++;
            }

            if (search == c) {
                finding = 1;

                if(length == i + 1) {

                    i = finding = 0;
                    count++;
                    search = search_this_word[i];
                }

                else {
                    search = search_this_word[++i];
                }

            }
            else {
                finding = 0;
            }
        }

        if (count > 0) {
            return first_occurance - length + 1;
        }
        else {
            return -1;
        }
    }
    else {
        // puts ("Cannot open file to read");
        return -1;
    }
    return -1;
}

int* findWordInDirectory(char **argv, int *arr)
{
    int count = 0;
    DIR *d;
    char wordToFind[50];
    struct dirent *dir;

    d = opendir(".");

    strcpy(wordToFind, argv[1]);

    if(d) {
        while((dir = readdir(d)) != NULL) {
            arr[count] = findWordIn(wordToFind, dir -> d_name);
            count += 1;
        }
        closedir(d);
    }

    return arr;

}

int main(int argc, char **argv)
{
    int arr[50], count = 0;

    int *ptr = findWordInDirectory(argv, arr);

    DIR *d;
    struct dirent *dir;

    d = opendir(".");

    if (d) {
        printf("Position   Filename\n");
        while ((dir = readdir(d)) != NULL){

            if(ptr[count] != -1) {
                printf("%8d   %s\n", ptr[count], dir->d_name);
            }
            count += 1;
        }
        closedir(d);
    }

    return 0;
}
```

```
**************************************************************************
**************************************************************************
```

output :

```
E:\MCA-II\Advanced C\FILES\pro_temp_2\Debug>pro_16 Rollwala

Position   Filename
       7   Rollwala.txt
       7   Names.txt
```

```
**************************************************************************
**************************************************************************
```

d. Rename given file.

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


int main(int argc, char **argv)
{
    if(argc != 3) {
        printf("Invalid Arguments\n");
        printf("Example\n objFile oldFile.txt newFile.txt\n");
        exit(1);
    }

    if(rename(argv[1], argv[2]) == 0)
    {
        printf("File renamed successfully.\n");
        exit(0);
    }
    printf("Cannot rename File\n");

    return 0;
}


*************************************************************************
*************************************************************************
output :


E:\MCA-II\Advanced C\FILES\pro_temp_2\Debug>pro_16 Temperory.txt
Realone.txt
File renamed successfully.


*************************************************************************
*************************************************************************
e. List all EXE files in a given diectory.
*************************************************************************
*************************************************************************
#include <stdio.h>
#include <dirent.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <string.h>

int isExecutable(char fileName[])
{
    struct stat sb;
    if (stat(fileName, &sb) == 0 && sb.st_mode & S_IXUSR) {
        return 1;
    }
    else {
        return 0;
    }
}


int main(int argc, char **argv)
{
    char folderName[50];
    DIR *d;
    struct dirent *dir;
    int count = 0;

    if (argc == 2) {
        strcpy(folderName, argv[1]);
        d = opendir(folderName);
    }
    else if (argc == 1){
        printf("Checking current directory\n");
        d = opendir(".");
    }
    else {
        printf("Enter only 1 directory");
    }
    if (d) {
        while ((dir = readdir(d)) != NULL)
        {
            if(isExecutable(dir -> d_name)) {
                count++;
                printf("%s" ,dir->d_name);
                printf(" is executable\n");
            }
        }
        if(!count) {
            printf("No executables Found");
        }
    }
    else{
        printf("Directory Path Invalid\n");
    }
    return 0;
}
```

```
*************************************************************************
*************************************************************************
output :

E:\MCA-II\Advanced C\FILES\pro_temp_2\Debug>pro_16

Checking current directory
```

```
a is executable
.. is executable
. is executable
```

```
*************************************************************************
*************************************************************************
```

f. Merge two files into third file.

```
*************************************************************************
*************************************************************************
#include<stdlib.h>

int copyFile(char sourceFile[], char destinationFile[])
{
    int no_of_lines = 0, success = 1;
    char line[1000];

    FILE *f1 = fopen(sourceFile, "r");
    FILE *f2 = fopen(destinationFile, "a+");

    printf("Working on \"%s\" file:\n", sourceFile);

    if(f1 && f2) {
        while(fgets(line, sizeof line, f1)) {
            no_of_lines++;
            fputs (line, f2);
        }

        printf("%d lines yanked and pasted", no_of_lines);

        fclose(f1);
        fclose(f2);

    }

    else {
        success = 0;
        printf("No such File exists");
    }
    puts ("\n");
    return success;
}

int copyFiles(int argc, char **argv)
{
    int i, allSuccess = 1, success;

    if (argc < 3) {
        printf("Enter atleast 2 File Names\n");
        exit(1);
    }

    FILE *destinationFile = fopen(argv[argc - 1], "w");
    fclose(destinationFile);
    for (i = 1; i < argc - 1; i++) {
        success = copyFile(argv[i], argv[argc - 1]);
        if(success && allSuccess) {
            allSuccess = 1;
        }
        else {
            allSuccess = 0;
        }
    }

    return allSuccess;

}

int main(int argc, char **argv)
{
    int success;

    success = copyFiles(argc, argv);

    if(success) {
        printf("All File are copied Successfully\n");
    }
    else {
        printf("All files are NOT copied\n");
    }

    return 0;
}

*************************************************************************
*************************************************************************
output:

E:\MCA-II\Advanced C\FILES\pro_temp_2\Debug>pro_16 a.txt b.txt  c.txt
destination.txt

    Working on "a.txt" file:
    2 lines yanked and pasted

    Working on "b.txt" file:
    5 lines yanked and pasted

    Working on "c.txt" file:
    2 lines yanked and pasted

    All File are copied Successfully


*************************************************************************
*************************************************************************
```