

# Assignment 1

```

/*****
*****
*****
*****/

```

NAME : Pradip S Karmakar

ROLL NO : 10

CLASS : MCA (SEM-2)

SUBJECT : Advance Programing (AP)

```

*****
*****
*****
*****/

```

```

/*****
*****
*****
*****/

```

DEFINE :

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>

#include<stdlib.h>


struct player{
    char Player_name[15];
    float Batting_average;
}players;


struct teams{
    int total_players;
    char Team_name[15];
    struct player players[50];
};


int getnplayer();
struct teams getData();
void getSort(struct teams[],int);
void setPrint(struct teams[],int);
void menudriven(struct teams[],int);
void main()
{
    int i,j,team_count;
    struct teams s[10];

    team_count = getnteam();
    for( i = 0; i < team_count; i++ )
    {
        s[i] = getData();
    }
}

```

```

printf("\n\nBefore Sorting List \n");
setPrint(s,team_count);
getSort(s,team_count);
printf("\n\nAfter Sorting List \n");
setPrint(s,team_count);
menudriven(s,team_count);
getch();
}

```

```

struct teams getData()
{
    int i,player_count;
    struct teams g;
    printf("\nEnter Team Name : ");
    scanf("%s",g.Team_name);
    player_count = getnplayer();
    g.total_players = player_count;
    for( i = 0; i < player_count; i++)
    {
        printf("\nEnter Player Name : ");
        scanf("%s",g.players[i].Player_name);
        printf("\nEnter Batting Average : ");
        scanf("%f",&g.players[i].Batting_average);
    }
    printf("\n_____ \n");
    return g;
}

```

```
int getnteam()
```

```
{
```

```
    int x;
```

```
    printf("\nEnter The Number of Teams : ");
```

```
    scanf("%d",&x);
```

```
    return x;
```

```
}
```

```
int getnplayer()
```

```
{
```

```
    int y;
```

```
    printf("\nEnter The Number of Players : ");
```

```
    scanf("%d",&y);
```

```
    return y;
```

```
}
```

```
void getSort(struct teams s[],int n)
```

```
{
```

```
    int i,j,k,l,Check;
```

```
    struct teams temp;
```

```
    struct player p;
```

```
    for( i = 0; i < n; i++ )
```

```
    {
```

```
        for( j = 0; j < s[i].total_players; j++)
```

```
        {
```

```
            for( k = 0; k < s[i].total_players - j -1; k++)
```

```
            {
```

```
                if(s[i].players[k].Batting_average < s[i].players[k+1].Batting_average
```

```
            )
```

```
            {
```

```
                p = s[i].players[k];
```

```

                                s[i].players[k] = s[i].players[k+1];
                                s[i].players[k+1] = p;
                                }
                            }
                        }
                    }
                for( i = 0; i < n; i++ )
                {
                    for( j = 0; j < n - i - 1; j++ )
                    {
                        Check = strcmp(s[j].Team_name,s[j+1].Team_name);
                        if(Check > 0)
                        {
                            temp = s[j];
                            s[j] = s[j+1];
                            s[j+1] = temp;
                        }
                    }
                }
            }
        }
    }

```

```

void setPrint(struct teams p[],int n)
{
    int i,j;
    for( i = 0; i < n; i++ )
    {
        printf("\nTEAM NAME =
%s\n*****\n",p[i].Team_name);

        for( j = 0; j < p[i].total_players; j++ )
        {

```

```

                printf("\n\tPLAYER NAME = %s \n\tBATTING AVERAGE =
%.2f\n*****\n",p[i].players[j].Player_name,p[i].players[j].Bat
ting_average);
            }
        }
    }

```

```

void menudriven(struct teams p[],int n)

```

```

{
    int val,i,j,Check;
    char player[15];
    printf("\n\nPress 1 To Search Player Detail ");
    printf("\n\nPress 2 To Exit The Program\n");
    scanf("%d",&val);
    if(val == 1)
    {
        printf("\n\nEnter Player Name : ");
        scanf("%s",player);

        for( i = 0; i < n; i++ )
        {
            for( j = 0; j < p[i].total_players; j++ )
            {
                Check = strcmp(p[i].players[j].Player_name,player);
                if(Check == 0)
                {
                    printf("\n\nPLAYER NAME = %s \nTEAM NAME = %s
\nBATTING AVERAGE =
%.2f\n*****\n",p[i].players[j].Player_name,p[i].Team_name,
p[i].players[j].Batting_average);

                    menudriven(p,n);
                }
            }
        }
    }
}

```

```

        }
    }
    else if(val == 2)
    {
        exit(0);
    }
    else{
        printf("\n\nPlease Select Proper Options.\n");
        menudriven(p,n);
    }
    printf("\n\nPlayer Detail Not Found\n");
    menudriven(p,n);
}

```

```

/*****
*****
*****

```

OUTPUT :

Enter The Number of Teams : 3

Enter Team Name : XLOAD

Enter The Number of Players : 3

Enter Player Name : Pradip

Enter Batting Average : 52.4

Enter Player Name : Monil

Enter Batting Average : 60.8

Enter Player Name : Sudip

Enter Batting Average : 40.6

---

Enter Team Name : UNOFFICIAL

Enter The Number of Players : 2

Enter Player Name : Nirav

Enter Batting Average : 34.6

Enter Player Name : Ajinkya

Enter Batting Average : 12.6

---

Enter Team Name : REBELS

Enter The Number of Players : 4

Enter Player Name : Vijay

Enter Batting Average : 69.8



Enter Player Name : Kohli

Enter Batting Average : 89.3

Enter Player Name : Dhoni

Enter Batting Average : 92.5

Enter Player Name : Rohit

Enter Batting Average : 95.8

---

Before Sorting List

TEAM NAME = XLOAD

\*\*\*\*\*

PLAYER NAME = Pradip

BATTING AVERAGE = 52.40

\*\*\*\*\*

PLAYER NAME = Monil

BATTING AVERAGE = 60.80

\*\*\*\*\*

PLAYER NAME = Sudip

BATTING AVERAGE = 40.60

\*\*\*\*\*

TEAM NAME = UNOFFICIAL

\*\*\*\*\*

PLAYER NAME = Nirav

BATTING AVERAGE = 34.60

\*\*\*\*\*

PLAYER NAME = Ajinkya

BATTING AVERAGE = 12.60

\*\*\*\*\*

TEAM NAME = REBELS

\*\*\*\*\*

PLAYER NAME = Vijay

BATTING AVERAGE = 69.80

\*\*\*\*\*

PLAYER NAME = Kohli

BATTING AVERAGE = 89.30

\*\*\*\*\*

PLAYER NAME = Dhoni

BATTING AVERAGE = 92.50

\*\*\*\*\*

PLAYER NAME = Rohit

BATTING AVERAGE = 95.80

\*\*\*\*\*

After Sorting List

TEAM NAME = REBELS

\*\*\*\*\*

PLAYER NAME = Rohit

BATTING AVERAGE = 95.80

\*\*\*\*\*

PLAYER NAME = Dhoni

BATTING AVERAGE = 92.50

\*\*\*\*\*

PLAYER NAME = Kohli

BATTING AVERAGE = 89.30

\*\*\*\*\*

PLAYER NAME = Vijay

BATTING AVERAGE = 69.80

\*\*\*\*\*

TEAM NAME = UNOFFICIAL

\*\*\*\*\*

PLAYER NAME = Nirav

BATTING AVERAGE = 34.60

\*\*\*\*\*

PLAYER NAME = Ajinkya

BATTING AVERAGE = 12.60

\*\*\*\*\*

TEAM NAME = XLOAD

\*\*\*\*\*

PLAYER NAME = Monil

BATTING AVERAGE = 60.80

\*\*\*\*\*

PLAYER NAME = Pradip

BATTING AVERAGE = 52.40

\*\*\*\*\*

PLAYER NAME = Sudip

BATTING AVERAGE = 40.60

\*\*\*\*\*

Press 1 To Search Player Detail

Press 2 To Exit The Program

1

Enter Player Name : Rohit

PLAYER NAME = Rohit

TEAM NAME = REBELS

BATTING AVERAGE = 95.80

\*\*\*\*\*

Press 1 To Search Player Detail

Press 2 To Exit The Program

1

Enter Player Name : Unknown

Player Detail Not Found

Press 1 To Search Player Detail

Press 2 To Exit The Program

3

Please Select Proper Options.

Press 1 To Search Player Detail

Press 2 To Exit The Program

```

*****
*****
*****//*****
*****
*****

```

NAME : Pradip S Karmakar

ROLL NO : 10

CLASS : MCA (SEM-2)

SUBJECT : Advance Programing (AP)

```

*****
*****
*****/

```

```

/*****
*****
*****

```

DEFINE :

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.

```
*****  
*****  
*****/
```

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

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

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

```
union Game{
```

```
    struct football{
```

```
        char coachname[15];
```

```
        int played,wins,lose,draw,goals;
```

```
    }football;
```

```
    struct baseball{
```

```
        char coachname[15];
```

```
        int played,wins,lose,draw,homeruns;
```

```
    }baseball;
```

```
}games;
```

```
struct league{
```

```
    int Game_Id;
```

```
    char Team_Name[20];
```

```
    char city[15];  
    union Game games;  
};
```

```
int getData(struct league[]);  
void setPrint(struct league[],int);  
void menudriven(struct league[],int);
```

```
void main()  
{  
    struct league game[15];  
    int i,j,n;  
    n = getData(game);  
    /*printf("")*/  
    setPrint(game,n);  
    menudriven(game,n);  
}
```

```
int getData(struct league p[])  
{  
    int i,j,total_league;  
    printf("Enter Total Leagues : ");  
    scanf("%d",&total_league);  
    for( i = 0;i < total_league; i++ )  
    {  
        printf("\n\nPress 1 For Football \n\nPress 2 For Baseball\n\n");  
        scanf("%d",&p[i].Game_Id);  
  
        if(p[i].Game_Id == 1)  
        {  
            printf("\nEnter Team Name : ");
```



```

scanf("%s",p[i].Team_Name);

printf("\nEnter City : ");

scanf("%s",p[i].city);

printf("\nEnter Coach Name : ");

scanf("%s",p[i].games.football.coachname);

printf("\nEnter Total Numbers of Matches Played : ");

scanf("%d",&p[i].games.football.played);

printf("\nEnter Total Numbers of Wins : ");

scanf("%d",&p[i].games.football.wins);

printf("\nEnter Total Numbers of Loses : ");

scanf("%d",&p[i].games.football.lose);

printf("\nEnter Total Numbers of Goals : ");

scanf("%d",&p[i].games.football.goals);

p[i].games.football.draw = p[i].games.football.played-
p[i].games.football.wins-p[i].games.football.lose;

printf("_____");

}

else if(p[i].Game_Id == 2)

{

printf("\nEnter Team Name : ");

scanf("%s",p[i].Team_Name);

printf("\nEnter City : ");

scanf("%s",p[i].city);

printf("\nEnter Coach Name : ");

scanf("%s",p[i].games.baseball.coachname);

printf("\nEnter Total Numbers of Matches Played : ");

scanf("%d",&p[i].games.baseball.played);

printf("\nEnter Total Numbers of Wins : ");

scanf("%d",&p[i].games.baseball.wins);

printf("\nEnter Total Numbers of Loses : ");

scanf("%d",&p[i].games.baseball.lose);

```

```

        printf("\nEnter Total Numbers of Homeruns : ");

        scanf("%d",&p[i].games.baseball.homeruns);

        p[i].games.baseball.draw = p[i].games.baseball.played-
p[i].games.baseball.wins-p[i].games.baseball.lose;

        printf("_____");

        }

        else

        {

                printf("\nPlease Select Proper Options\n");

                --i;

        }

    }

    return total_league;

}

```

```

void setPrint(struct league p[], int n)

```

```

{

    int i;

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

    {

        if(p[i].Game_Id == 1)

        {

                printf("\n\n\tLeague : Football\n\tTeam Name : %s\n\tCity : %s\n\tCoach :
%s\n\tMatch Played : %d\n\tWins : %d\n\tLoses : %d\n\tDraws : %d\n\tTotal Goals :
%d\n\n_____",p[i].Team_Name,p[i].city,p[i].games.football.coachname,p[i].games.football.played,p[i].games.f
ootball.wins,p[i].games.football.lose,p[i].games.football.draw,p[i].games.football.goals);

        }

        else{

                printf("\n\n\tLeague : Baseball\n\tTeam Name : %s\n\tCity : %s\n\tCoach :
%s\n\tMatch Played : %d\n\tWins : %d\n\tLoses : %d\n\tDraws : %d\n\tTotal Goals :
%d\n\n_____",p[i].Team_Name,p[i].city,p[i].games.baseball.coachname,p[i].games.baseball.played,p[i].games.b
aseball.wins,p[i].games.baseball.lose,p[i].games.baseball.draw,p[i].games.baseball.goals);

        }

    }

}

```

```

__",p[i].Team_Name,p[i].city,p[i].games.baseball.coachname,p[i].games.baseball.played,p[i].games.
baseball.wins,p[i].games.baseball.lose,p[i].games.baseball.draw,p[i].games.baseball.homeruns);
    }
}
}

```

```

void menudriven(struct league p[],int n)

```

```

{
    int i,j,option = 0;
    struct league temp;
    printf("\n\nPress 1 To Display Data in Win Wise. \n\nPress 2 To Display Data in Lose Wise.
\n\nPress 3 To Display Data in Draw Wise. \n\nPress 4 To Exit Program.\n\n");\
    scanf("%d",&option);
    if(option == 1)
    {
        for( i = 0; i < n; i++ )
        {
            for( j = 0; j < n-i-1; j++ )
            {
                if(p[j].games.football.wins < p[j+1].games.football.wins ||
p[j].games.football.wins < p[j+1].games.baseball.wins || p[j].games.baseball.wins <
p[j+1].games.baseball.wins || p[j].games.baseball.wins < p[j+1].games.football.wins)
                {
                    temp = p[j];
                    p[j] = p[j+1];
                    p[j+1] = temp;
                }
            }
        }
        setPrint(p,n);
        menudriven(p,n);
    }
}

```

```

else if(option == 2)
{
    for( i = 0; i < n; i++ )
    {
        for( j = 0; j < n-i-1; j++ )
        {
            if(p[j].games.football.lose < p[j+1].games.football.lose ||
p[j].games.football.lose < p[j+1].games.baseball.lose || p[j].games.baseball.lose <
p[j+1].games.baseball.lose || p[j].games.baseball.lose < p[j+1].games.football.lose)
            {
                temp = p[j];
                p[j] = p[j+1];
                p[j+1] = temp;
            }
        }
    }
    setPrint(p,n);
    menudriven(p,n);
}

else if(option == 3)
{
    for( i = 0; i < n; i++ )
    {
        for( j = 0; j < n-i-1; j++ )
        {
            if(p[j].games.football.draw < p[j+1].games.football.draw ||
p[j].games.football.draw < p[j+1].games.baseball.draw || p[j].games.baseball.draw <
p[j+1].games.baseball.draw || p[j].games.baseball.draw < p[j+1].games.football.draw)
            {
                temp = p[j];
                p[j] = p[j+1];
                p[j+1] = temp;
            }
        }
    }
}

```

```

        }
    }
}
setPrint(p,n);
menudriven(p,n);
}
else if(option == 4)
{
    exit(0);
}
else
{
    printf("Please Select Proper Option.");
    menudriven(p,n);
}
}

```

```

/*****
*****
*****

```

OUTPUT :

Enter Total Leagues : 5

Press 1 For Football

Press 2 For Baseball

1

Enter Team Name : Xload

Enter City : Navsari

Enter Coach Name : Pradip

Enter Total Numbers of Matches Played : 34

Enter Total Numbers of Wins : 30

Enter Total Numbers of Loses : 2

Enter Total Numbers of Goals : 90

---

Press 1 For Football

Press 2 For Baseball

1

Enter Team Name : Unofficial

Enter City : Kutch

Enter Coach Name : Nirav

Enter Total Numbers of Matches Played : 40

Enter Total Numbers of Wins : 32

Enter Total Numbers of Loses : 3

Enter Total Numbers of Goals : 72

---

Press 1 For Football

Press 2 For Baseball

2

Enter Team Name : Loafer

Enter City : Ahmedabad

Enter Coach Name : Monil

Enter Total Numbers of Matches Played : 27

Enter Total Numbers of Wins : 12

Enter Total Numbers of Loses : 5

Enter Total Numbers of Homeruns : 56

---

Press 1 For Football

Press 2 For Baseball

2

Enter Team Name : Bittu

Enter City : Mundra

Enter Coach Name : Lakshya

Enter Total Numbers of Matches Played : 70

Enter Total Numbers of Wins : 37

Enter Total Numbers of Loses : 12

Enter Total Numbers of Homeruns : 102

---

Press 1 For Football

Press 2 For Baseball

1

Enter Team Name : Google

Enter City : Surat

Enter Coach Name : Ajinkya



Enter Total Numbers of Matches Played : 103

Enter Total Numbers of Wins : 20

Enter Total Numbers of Loses : 49

Enter Total Numbers of Goals : 60

---

League : Football

Team Name : Xload

City : Navsari

Coach : Pradip

Match Played : 34

Wins : 30

Loses : 2

Draws : 2

Total Goals : 90

---

League : Football

Team Name : Unofficial

City : Kutch

Coach : Nirav

Match Played : 40

Wins : 32

Loses : 3

Draws : 5

Total Goals : 72

---

League : Baseball

Team Name : Loafer

City : Ahmedabad

Coach : Monil

Match Played : 27

Wins : 12

Loses : 5

Draws : 10

Total Goals : 56

---

League : Baseball

Team Name : Bittu

City : Mundra

Coach : Lakshya

Match Played : 70

Wins : 37

Loses : 12

Draws : 21

Total Goals : 102

---

League : Football

Team Name : Google

City : Surat

Coach : Ajinkya

Match Played : 103

Wins : 20

Loses : 49

Draws : 34

Total Goals : 60

---

Press 1 To Display Data in Win Wise.

Press 2 To Display Data in Lose Wise.

Press 3 To Display Data in Draw Wise.

Press 4 To Exit Program.

1

League : Baseball

Team Name : Bittu

City : Mundra

Coach : Lakshya

Match Played : 70

Wins : 37

Loses : 12

Draws : 21

Total Goals : 102

---

League : Football

Team Name : Unofficial

City : Kutch

Coach : Nirav

Match Played : 40

Wins : 32

Loses : 3

Draws : 5

Total Goals : 72

---

League : Football

Team Name : Xload

City : Navsari

Coach : Pradip

Match Played : 34

Wins : 30

Loses : 2

Draws : 2

Total Goals : 90

---

League : Football

Team Name : Google

City : Surat

Coach : Ajinkya

Match Played : 103

Wins : 20

Loses : 49

Draws : 34

Total Goals : 60

---

League : Baseball

Team Name : Loafer

City : Ahmedabad

Coach : Monil

Match Played : 27

Wins : 12

Loses : 5

Draws : 10

Total Goals : 56

---

Press 1 To Display Data in Win Wise.

Press 2 To Display Data in Lose Wise.

Press 3 To Display Data in Draw Wise.

Press 4 To Exit Program.

2

League : Football

Team Name : Google

City : Surat

Coach : Ajinkya

Match Played : 103

Wins : 20

Loses : 49

Draws : 34

Total Goals : 60

---

League : Baseball

Team Name : Bittu

City : Mundra

Coach : Lakshya

Match Played : 70

Wins : 37

Loses : 12

Draws : 21

Total Goals : 102

---

League : Baseball

Team Name : Loafer

City : Ahmedabad

Coach : Monil

Match Played : 27

Wins : 12

Loses : 5

Draws : 10

Total Goals : 56

---

League : Football

Team Name : Unofficial

City : Kutch

Coach : Nirav

Match Played : 40

Wins : 32

Loses : 3

Draws : 5

Total Goals : 72

---

League : Football

Team Name : Xload

City : Navsari

Coach : Pradip

Match Played : 34

Wins : 30

Loses : 2

Draws : 2

Total Goals : 90

---

Press 1 To Display Data in Win Wise.

Press 2 To Display Data in Lose Wise.

Press 3 To Display Data in Draw Wise.

Press 4 To Exit Program.

3

League : Football

Team Name : Google

City : Surat

Coach : Ajinkya

Match Played : 103

Wins : 20

Loses : 49

Draws : 34

Total Goals : 60

---

League : Baseball

Team Name : Bittu

City : Mundra

Coach : Lakshya

Match Played : 70

Wins : 37

Loses : 12

Draws : 21

Total Goals : 102

---

League : Baseball

Team Name : Loafer



City : Ahmedabad

Coach : Monil

Match Played : 27

Wins : 12

Loses : 5

Draws : 10

Total Goals : 56

---

League : Football

Team Name : Unofficial

City : Kutch

Coach : Nirav

Match Played : 40

Wins : 32

Loses : 3

Draws : 5

Total Goals : 72

---

League : Football

Team Name : Xload

City : Navsari

Coach : Pradip

Match Played : 34

Wins : 30

Loses : 2

Draws : 2

Total Goals : 90

---

Press 1 To Display Data in Win Wise.

Press 2 To Display Data in Lose Wise.

Press 3 To Display Data in Draw Wise.

Press 4 To Exit Program.

```
*****
*****
*****//*****
*****
*****
```

NAME : Pradip S Karmakar

ROLL NO : 10

CLASS : MCA (SEM-2)

SUBJECT : Advance Programing (AP)

```
*****
*****
*****/
```

```
/*****
*****
*****
```

DEFINE :

3. Write a program that stores and displays the records of the customer. The 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].dateofpay.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.dd,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].name,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 : 4

Enter Account Number : 4285

Enter The Customer's Name : Pradip

Enter The Old Balance : 43030

Enter The Last Payment : 20892

Enter The Date of Last Payment [dd mm yy] : 3 01 20

---

Enter Account Number : 9812

Enter The Customer's Name : Nirav

Enter The Old Balance : 39002

Enter The Last Payment : 39000

Enter The Date of Last Payment [dd mm yy] : 14 12 19

---

Enter Account Number : 3209



Enter The Customer's Name : Lakshya

Enter The Old Balance : 5000

Enter The Last Payment : 5000

Enter The Date of Last Payment [dd mm yy] : 26 01 20

---

Enter Account Number : 9122

Enter The Customer's Name : Ajinkya

Enter The Old Balance : 50000

Enter The Last Payment : 2380

Enter The Date of Last Payment [dd mm yy] : 20 10 19

---

List Of Customers

---

Customer Id = 4285

Customer Name = Pradip

Old Balance = 43030.00

Last Payment = 20892.00

Last Payment Date = 3/1/20

New Balance = 22138.00

---

Customer Id = 9812

Customer Name = Nirav

Old Balance = 39002.00

Last Payment = 39000.00

Last Payment Date = 14/12/19

New Balance = 2.00

---

Customer Id = 3209

Customer Name = Lakshya

Old Balance = 5000.00

Last Payment = 5000.00

Last Payment Date = 26/1/20

New Balance = 0.00

---

Customer Id = 9122

Customer Name = Ajinkya

Old Balance = 50000.00

Last Payment = 2380.00

Last Payment Date = 20/10/19

New Balance = 47620.00

---

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.

1

List Of Customers

---

Customer Id = 4285

Customer Name = Pradip

Old Balance = 43030.00

Last Payment = 20892.00

Last Payment Date = 3/1/20

New Balance = 22138.00

---

Customer Id = 9812

Customer Name = Nirav

Old Balance = 39002.00

Last Payment = 39000.00

Last Payment Date = 14/12/19

New Balance = 2.00

---

Customer Id = 3209

Customer Name = Lakshya

Old Balance = 5000.00

Last Payment = 5000.00

Last Payment Date = 26/1/20

New Balance = 0.00

---

Customer Id = 9122

Customer Name = Ajinkya

Old Balance = 50000.00

Last Payment = 2380.00

Last Payment Date = 20/10/19

New Balance = 47620.00

---

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.

2

Enter Customer Name : Pradip

Customer Id = 4285

Customer Name = Pradip

Old Balance = 43030.00

Last Payment = 20892.00

Last Payment Date = 3/1/20

New Balance = 22138.00

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.

3

Customer Name = Pradip

Status = Pending

---

Customer Name = Nirav

Status = Pending

---

Customer Name = Lakshya

Status = Clear

---

Customer Name = Ajinkya

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 = Pradip

Current Balance = 22138.00

---

Customer Name = Nirav

Current Balance = 2.00

---

Customer Name = Lakshya

Current Balance = 0.00

---

Customer Name = Ajinkya

Current Balance = 47620.00

---

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.

```
*****
*****
*****//*****
*****
*****
```

NAME : Pradip S Karmakar

ROLL NO : 10

CLASS : MCA (SEM-2)

SUBJECT : Advance Programing (AP)

```
*****
*****
*****/
```

```
/*****
*****
*****
```

DEFINE :

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.

```
*****
*****
*****/
```

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

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

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



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

```
typedef struct{  
    int stock;  
    char author[15],title[15],publisher[15];  
    float price;  
}books;
```

```
books getData();
```

```
void menudriven(books[],int);
```

```
void main()  
{  
    books b[50];  
    int i,total_books;  
    printf("Enter Total Number of Books : ");  
    scanf("%d",&total_books);  
    for( i = 0; i < total_books; i++ )  
    {  
        b[i] = getData();  
    }  
  
    menudriven(b,total_books);  
}
```

```
books getData()  
{  
    books b;  
    printf("\nEnter Book's Author Name : ");  
    scanf("%s",b.author);
```

```

printf("\nEnter Book's Title : ");
scanf("%s",b.title);
printf("\nEnter Book's Publisher Name : ");
scanf("%s",b.publisher);
printf("\nEnter Book Price : ");
scanf("%f",&b.price);
printf("\nEnter Book Stock : ");
scanf("%d",&b.stock);
printf("_____");
return b;
}

```

```

void menudriven(books b[],int tb)
{
    int i,j,option,check1,check2,copies;
    char auth[15],title[15];
    float amt;
    printf("\n\n\tPress 1 For Book Availbility \n\n\tPress 2 For Exit\n");
    scanf("%d",&option);
    if(option == 1)
    {
        printf("\nEnter Book Title : ");
        scanf("%s",title);
        printf("\nEnter Author Name : ");
        scanf("%s",auth);
        for( i = 0; i < tb; i++ )
        {
            check1 = strcmp(b[i].title,title);
            check2 = strcmp(b[i].author,auth);
            if(check1 == 0 && check2 == 0)
            {

```

```

        printf("\nEnter Number of Copies You Want : ");
        scanf("%d",&copies);
        if(b[i].stock >= copies)
        {
            amt = b[i].price * copies;
            printf("Total Payable Amount For %d Copies Will Be : Rs
%.2f",copies,amt);

            menudriven(b,tb);
        }
        else
        {
            printf("Only %d Copies Are Available",b[i].stock);
            menudriven(b,tb);
        }
    }

    printf("Book Not Found");
    menudriven(b,tb);
}

else if(option == 2)
{
    exit(0);
}

else
{
    printf("Please Select Proper Options");
    menudriven(b,tb);
}
}

```

/\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

OUTPUT :

Enter Total Number of Books : 3

Enter Book's Author Name : Pradip

Enter Book's Title : C

Enter Book's Publisher Name : Mccgrow

Enter Book Price : 490

Enter Book Stock : 4

---

Enter Book's Author Name : Nirav

Enter Book's Title : DBMS

Enter Book's Publisher Name : TATA

Enter Book Price : 910

Enter Book Stock : 6

---

Enter Book's Author Name : Ajinkya

Enter Book's Title : C

Enter Book's Publisher Name : TATA

Enter Book Price : 549

Enter Book Stock : 2

---

Press 1 For Book Availability

Press 2 For Exit

1

Enter Book Title : C

Enter Author Name : Ajinkya

Enter Number of Copies You Want : 3

Only 2 Copies Are Available

Press 1 For Book Availability

Press 2 For Exit

1

Enter Book Title : C

Enter Author Name : Pradip

Enter Number of Copies You Want : 3

Total Payable Amount For 3 Copies Will Be : Rs 1470.00

Press 1 For Book Availability

Press 2 For Exit

3

Please Select Proper Options

Press 1 For Book Availability

Press 2 For Exit

```
*****
*****
*****//*****
*****
*****
```

NAME : Pradip S Karmakar

ROLL NO : 10

CLASS : MCA (SEM-2)

SUBJECT : Advance Programing (AP)

```
*****
*****
*****/
```

```
/*****
*****
*****
```

DEFINE :

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 and print list of 2 & 4 wheeler vehicles accordingly.

```
*****  
*****  
*****/
```

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

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

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

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

```
typedef struct{
```

```
    float mileage;
```

```
    char type[10];
```

```
}two;
```

```
typedef struct{
    float mileage;
    char transmission[10],type[10];
    int Eng_no;
}four;
```

```
typedef struct{
    int vehi_type;
    char vehi_name[15];
    float price;
    union{
        two t;
        four f;
    };
}vehicle;
```

```
vehicle getData();
```

```
void setPrint(vehicle[],int,int);
```

```
void menudriven(vehicle[],int);
```

```
void main()
{
    vehicle v[50];
    int i,total_vehicles;
    printf("Enter Total Number of Vehicles : ");
    scanf("%d",&total_vehicles);
    for( i = 0; i < total_vehicles; i++ )
    {
        v[i] = getData();
    }
}
```



```

    }

    menudriven(v,total_vehicles);
}

vehicle getData()
{
    vehicle b;

    printf("\nPress 1 for 2 Wheelers \nPress 2 for 4 Wheelers\n");
    scanf("%d",&b.vehi_type);
    if(b.vehi_type == 1)
    {
        printf("\nEnter Vehicle Name : ");
        scanf("%s",b.vehi_name);
        printf("\nEnter Vehicle Price : ");
        scanf("%f",&b.price);
        printf("\nEnter Vehicle Mileage : ");
        scanf("%f",&b.t.mileage);
        printf("\nEnter Vehicle Type(Gear/Gearless) : ");
        scanf("%s",b.t.type);
    }
    else if(b.vehi_type == 2){
        printf("\nEnter Vehicle Name : ");
        scanf("%s",b.vehi_name);
        printf("\nEnter Vehicle Price : ");
        scanf("%f",&b.price);
        printf("\nEnter Vehicle Engine Number : ");
        scanf("%d",&b.f.Eng_no);
        printf("\nEnter Vehicle Mileage : ");
        scanf("%f",&b.f.mileage);
        printf("\nEnter Vehicle Transmission Type : ");
    }
}

```

```

        scanf("%s",b.f.transmission);

        printf("\nEnter Vehicle Type (Heavy/Light) : ");

        scanf("%s",b.f.type);

    }

    return b;
}

void setPrint(vehicle b[], int tv, int v_type)
{
    int i;

    for( i = 0; i < tv; i++ )
    {
        if( b[i].vehi_type == v_type )
        {
            if( v_type == 1 )
            {
                printf("\n\n\tVehicle Type = Two Wheeler \n\tVehicle Name = %s
\n\tVehicle Price = %.2f \n\tMileage = %.2f \n\tGear Type =
%s\n",b[i].vehi_name,b[i].price,
b[i].t.mileage,b[i].t.type);
            }
            else{
                printf("\n\n\tVehicle Type = Four Wheeler \n\tVehicle Name = %s
\n\tVehicle Price = %.2f \n\tMileage = %.2f \n\tTransmission Type = %s \n\t Type = %s \n\tEngine No
=
%d\n",b[i].vehi_name,b[i].price,
b[i].f.mileage,b[i].f.transmission,b[i].f.type,b[i].f.Eng_no);
            }
        }
    }
}

void menudriven(vehicle b[],int tv)

```

```

{

    int i,j,option;

    printf("\n\n\tPress 1 To Display All Vehicles. \n\n\tPress 2 To Display Two Wheelers
\n\n\tPress 3 To Display Four Wheelers \n\n\tPress 4 For Exit\n");

    scanf("%d",&option);

    if(option == 1)
    {

        printf("\n+++++Two Wheelers+++++");
        setPrint(b,tv,1);
        printf("\n+++++Four Wheelers+++++");
        setPrint(b,tv,2);
        menudriven(b,tv);
    }
    else if(option == 2)
    {

        printf("\n+++++Two Wheelers+++++");
        setPrint(b,tv,1);
        menudriven(b,tv);
    }
    else if(option == 3)
    {

        printf("\n+++++Four Wheelers+++++");
        setPrint(b,tv,2);
        menudriven(b,tv);
    }
    else if(option == 4)
    {

        exit(0);
    }
    else

```

```

    {
        printf("\nPlease Select Proper Options\n");
        menudriven(b,tv);
    }
}

/*****
*****
*****
*****/

```

OUTPUT :

Enter Total Number of Vehicles : 10

Press 1 for 2 Wheelers

Press 2 for 4 Wheelers

1

Enter Vehicle Name : Activa

Enter Vehicle Price : 62000

Enter Vehicle Mileage : 60

Enter Vehicle Type(Gear/Gearless) : Gearless

Press 1 for 2 Wheelers

Press 2 for 4 Wheelers

2

Enter Vehicle Name : Scorpio

Enter Vehicle Price : 450000

Enter Vehicle Engine Number : 20395420

Enter Vehicle Mileage : 32

Enter Vehicle Transmission Type : Manual

Enter Vehicle Type (Heavy/Light) : Heavy

Press 1 for 2 Wheelers

Press 2 for 4 Wheelers

2

Enter Vehicle Name : Nano

Enter Vehicle Price : 100000

Enter Vehicle Engine Number : 23947832

Enter Vehicle Mileage : 40

Enter Vehicle Transmission Type : Manual

Enter Vehicle Type (Heavy/Light) : Light

Press 1 for 2 Wheelers

Press 2 for 4 Wheelers

1

Enter Vehicle Name : PassionPro

Enter Vehicle Price : 59000

Enter Vehicle Mileage : 70

Enter Vehicle Type(Gear/Gearless) : Gear

Press 1 for 2 Wheelers

Press 2 for 4 Wheelers

1

Enter Vehicle Name : Pleasure

Enter Vehicle Price : 65000

Enter Vehicle Mileage : 62

Enter Vehicle Type(Gear/Gearless) : Gearless

Press 1 for 2 Wheelers

Press 2 for 4 Wheelers

2

Enter Vehicle Name : Fortuner

Enter Vehicle Price : 1200000

Enter Vehicle Engine Number : 239847328

Enter Vehicle Mileage : 28

Enter Vehicle Transmission Type : Auto

Enter Vehicle Type (Heavy/Light) : Heavy

Press 1 for 2 Wheelers

Press 2 for 4 Wheelers

2

Enter Vehicle Name : i10

Enter Vehicle Price : 460000

Enter Vehicle Engine Number : 23986432

Enter Vehicle Mileage : 30

Enter Vehicle Transmission Type : Manual

Enter Vehicle Type (Heavy/Light) : Light

Press 1 for 2 Wheelers

Press 2 for 4 Wheelers

2

Enter Vehicle Name : Rapid

Enter Vehicle Price : 2100000

Enter Vehicle Engine Number : 298374833

Enter Vehicle Mileage : 25

Enter Vehicle Transmission Type : Auto

Enter Vehicle Type (Heavy/Light) : Light

Press 1 for 2 Wheelers

Press 2 for 4 Wheelers

1

Enter Vehicle Name : R15

Enter Vehicle Price : 210000

Enter Vehicle Mileage : 35

Enter Vehicle Type(Gear/Gearless) : Gear

Press 1 for 2 Wheelers

Press 2 for 4 Wheelers

1

Enter Vehicle Name : Bullet

Enter Vehicle Price : 250000

Enter Vehicle Mileage : 40

Enter Vehicle Type(Gear/Gearless) : Gear



Press 1 To Display All Vehicles.

Press 2 To Display Two Wheelers

Press 3 To Display Four Wheelers

Press 4 For Exit

1

+++++Two Wheelers+++++

Vehicle Type = Two Wheeler

Vehicle Name = Activa

Vehicle Price = 62000.00

Mileage = 60.00

Gear Type = Gearless

---

Vehicle Type = Two Wheeler

Vehicle Name = PassionPro

Vehicle Price = 59000.00

Mileage = 70.00

Gear Type = Gear

---

Vehicle Type = Two Wheeler

Vehicle Name = Pleasure

Vehicle Price = 65000.00

Mileage = 62.00

Gear Type = Gearless

---

Vehicle Type = Two Wheeler

Vehicle Name = R15

Vehicle Price = 210000.00

Mileage = 35.00

Gear Type = Gear

---

Vehicle Type = Two Wheeler

Vehicle Name = Bullet

Vehicle Price = 250000.00

Mileage = 40.00

Gear Type = Gear

---

+++++Four Wheelers+++++

Vehicle Type = Four Wheeler

Vehicle Name = Scorpio

Vehicle Price = 450000.00

Mileage = 32.00

Transmission Type = Manual

Type = Heavy

Engine No = 20395420

---

Vehicle Type = Four Wheeler

Vehicle Name = Nano

Vehicle Price = 100000.00

Mileage = 40.00

Transmission Type = Manual

Type = Light

Engine No = 23947832

---

Vehicle Type = Four Wheeler

Vehicle Name = Fortuner

Vehicle Price = 1200000.00

Mileage = 28.00

Transmission Type = Auto

Type = Heavy

Engine No = 239847328

---

Vehicle Type = Four Wheeler

Vehicle Name = i10

Vehicle Price = 460000.00

Mileage = 30.00

Transmission Type = Manual

Type = Light

Engine No = 23986432

---

Vehicle Type = Four Wheeler

Vehicle Name = Rapid

Vehicle Price = 2100000.00

Mileage = 25.00

Transmission Type = Auto

Type = Light

Engine No = 298374833

---

Press 1 To Display All Vehicles.

Press 2 To Display Two Wheelers

Press 3 To Display Four Wheelers

Press 4 For Exit

2

+++++Two Wheelers+++++

Vehicle Type = Two Wheeler

Vehicle Name = Activa

Vehicle Price = 62000.00

Mileage = 60.00

Gear Type = Gearless

---

Vehicle Type = Two Wheeler

Vehicle Name = PassionPro

Vehicle Price = 59000.00

Mileage = 70.00

Gear Type = Gear

---

Vehicle Type = Two Wheeler

Vehicle Name = Pleasure

Vehicle Price = 65000.00

Mileage = 62.00

Gear Type = Gearless

---

Vehicle Type = Two Wheeler

Vehicle Name = R15

Vehicle Price = 210000.00

Mileage = 35.00

Gear Type = Gear

---

Vehicle Type = Two Wheeler

Vehicle Name = Bullet

Vehicle Price = 250000.00

Mileage = 40.00

Gear Type = Gear

---

Press 1 To Display All Vehicles.

Press 2 To Display Two Wheelers

Press 3 To Display Four Wheelers

Press 4 For Exit

3

+++++Four Wheelers+++++

Vehicle Type = Four Wheeler

Vehicle Name = Scorpio

Vehicle Price = 450000.00

Mileage = 32.00

Transmission Type = Manual

Type = Heavy

Engine No = 20395420

---

Vehicle Type = Four Wheeler

Vehicle Name = Nano

Vehicle Price = 100000.00

Mileage = 40.00

Transmission Type = Manual

Type = Light

Engine No = 23947832

---

Vehicle Type = Four Wheeler

Vehicle Name = Fortuner

Vehicle Price = 1200000.00

Mileage = 28.00

Transmission Type = Auto

Type = Heavy

Engine No = 239847328

---

Vehicle Type = Four Wheeler

Vehicle Name = i10

Vehicle Price = 460000.00

Mileage = 30.00

Transmission Type = Manual

Type = Light

Engine No = 23986432

---

Vehicle Type = Four Wheeler

Vehicle Name = Rapid

Vehicle Price = 2100000.00

Mileage = 25.00

Transmission Type = Auto

Type = Light

Engine No = 298374833

---

Press 1 To Display All Vehicles.

Press 2 To Display Two Wheelers

Press 3 To Display Four Wheelers

Press 4 For Exit

## Assignment 2

```
*****
*****
*****//*****
*****
*****
```

NAME : Pradip S Karmakar

ROLL NO : 10

CLASS : MCA (SEM-2)

SUBJECT : Advance Programing (AP)

```
*****
*****
*****/
```

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

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

```
int getData(int[], int);
```

```
int reverseData(int[], int);
```

```
void main()
```

```
{
```

```
    int numbers[20], n, i;
```

```
    int *p = numbers;
```

```
    printf("Enter Total Number of data You Want : ");
```

```
    scanf("%d",&n);
```

```
    getData(numbers, n);
```

```
    reverseData(numbers, n);
```

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



```
}
```

```
int getData(int arr[], int n)
```

```
{
```

```
    int i;
```

```
    int *ptr = arr;
```

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

```
        printf("Enter Number %d : ", i + 1);
```

```
        scanf("%d", ptr);
```

```
        ptr++;
```

```
    }
```

```
}
```

```
int reverseData(int arr[], int n)
```

```
{
```

```
    int i, *ptr = arr + n - 1;
```

```
    printf("\nNumbers in reverse Order \n");
```

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

```
        printf("%d ", *ptr);
```

```
        ptr--;
```

```
    }
```

```
}
```

Output:

Enter Total Number of data You Want : 5

Enter Number 1 : 45

Enter Number 2 : 34

Enter Number 3 : 67

Enter Number 4 : 12

Enter Number 5 : 98

Numbers in reverse Order

98 12 67 34 45

```
*****  
*****  
*****/
```

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

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

```
void getnumbers(int *,int);
```

```
void findmaxmin(int *, int, int *, int *);
```

```
int getlimit();
```

```
void main()
```

```
{  
    int numbers[20],limit,*ptr,*max,*min;  
    ptr = numbers;  
    limit = getlimit();  
    getnumbers(ptr,limit);  
    findmaxmin(ptr,limit,max,min);  
}
```

```
void getnumbers(int *p,int limit)
```

```
{  
    int i = 0;  
    while( i < limit )  
    {  
        printf("Fill Number %d : ",i+1);  
        scanf("%d",p);
```

```
        p++;  
        ++i;  
    }  
}
```

```
int getlimit()  
{  
    int l;  
    printf("Enter Total Numbers You Want : ");  
    scanf("%d",&l);  
    return l;  
}
```

```
void findmaxmin(int *p, int limit, int *max, int *min)  
{  
    int i;  
    max = p;  
    min = p;  
    for( i = 0; i < limit; i++ )  
    {  
        if( *p > *max )  
        {  
            max = p;  
        }  
        else if( *p < *min )  
        {  
            min = p;  
        }  
        p++;  
    }  
    printf("\nMaximum Number : %d at Address %p\n",*max,max);
```

```
printf("\nMinimum Number : %d at Address %p\n",*min,min);  
}
```

Output :

Enter Total Numbers You Want : 5

Fill Number 1 : 23

Fill Number 2 : 56

Fill Number 3 : 1

Fill Number 4 : 78

Fill Number 5 : 65

max56min1max78

Maximum Number : 78 at Address 000000000061FDBC

Minimum Number : 1 at Address 000000000061FDB8

```
*****  
*****  
*****/
```

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

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

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

```
void filterString(char[]);
```

```
void main()
```

```
{
```

```
    char string1[50];
```

```
    printf("Enter String : ");
```

```

scanf("%[^\\n]",string1);
filterString(string1);
printf("\\n");
}

```

```

void filterString(char string1[])
{
    char *cptr;

    int Vowels = 0, Consonants = 0, Numbers = 0, Whitespaces = 0, Special_Charaacters = 0;

    cptr = string1;
    while(*cptr != '\\0') {

        if(*cptr == 65 || *cptr == 69 || *cptr == 73 || *cptr == 79 || *cptr == 85 || *cptr == 97 || *cptr
== 101 || *cptr == 105 || *cptr == 111 || *cptr == 117) {

            Vowels++;

        }

        else if (*cptr > 47 && *cptr < 58) {

            Numbers++;

        }

        else if(*cptr == 9 || *cptr == 32) {

            Whitespaces++;

        }

        else if((*cptr > 64 && *cptr < 91) || (*cptr > 96 && *cptr < 123)){

            Consonants++;

        }

        else {

            Special_Charaacters++;

        }

        cptr++;

    }

    printf("\\n Vowels: %d\\n", Vowels);
}

```

```

printf("\n Numbers: %d\n", Numbers);
printf("\n Whitespaces: %d\n", Whitespaces);
printf("\n Consonants: %d\n", Consonants);
printf("\n Special Characters: %d\n", Special_Characters);
}

```

Output:

Enter String : Pradip @s Karmakar 23

Vowels: 5

Numbers: 2

Whitespaces: 3

Consonants: 10

Special Characters: 1

```

*****
*****
***** /

```

```

#include<stdio.h>

```

```

#include<conio.h>

```

```

int getrow();

```

```

int getcol();

```

```

void getdata(int[][50],int,int);

```

```

void transpose(int[][50],int[][50],int,int);

```

```

void printdata(int[][50],int,int);

```

```
void main()
{
    int matrix[50][50],temp[50][50],i,j;
    i = getrow();
    j = getcol();
    getdata(matrix,i,j);
    transpose(matrix,temp,i,j);
    printdata(temp,i,j);
    getch();
}
```

```
int getrow()
{
    int n;
    printf("Enter The Number Rows : ");
    scanf("%d",&n);
    return n;
}
```

```
int getcol()
{
    int n;
    printf("Enter The Number Columns : ");
    scanf("%d",&n);
    return n;
}
```

```
void getdata(int matrix[][50],int r, int c)
{
    int i,j,(*ptr)[50];
```

```

ptr = matrix;
for( i = 0; i < r; i++ )
{
    for( j = 0; j < c; j++ )
    {
        printf("Enter The Value for Matrix[%d][%d] : ",i,j);
        scanf("%d",&*(ptr + i + j));
    }
}

```

```

void printdata(int temp[][50],int r, int c)
{
    int i,j,(*ptr)[50];
    ptr = temp;
    for( i = 0; i < c; i++ )
    {
        for( j = 0; j < r; j++ )
        {
            printf("Value At Matrix[%d][%d] : %d\n",*(ptr + i + j));
        }
    }
}

```

```

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

```



```

        for( j = 0; j < c; j++ )
        {
            *(*(ptr2 + j) + i) = *(*(ptr + i) + j);
        }
    }
}

```

Output :

Enter The Number Rows : 3

Enter The Number Columns : 2

Enter The Value for Matrix[0][0] : 4

Enter The Value for Matrix[0][1] : 76

Enter The Value for Matrix[1][0] : 2

Enter The Value for Matrix[1][1] : 5

Enter The Value for Matrix[2][0] : 7

Enter The Value for Matrix[2][1] : 2

4 2 7

76 5 2

```

*****
*****
*****/

```

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

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

```
int getcolrow(int);
```

```
void getdata(int[][50],int,int);
```

```
void multiply(int[][50],int[][50],int[][50],int,int);
```

```
void printdata(int[][50],int,int);
```

```

void main()
{
    int i, j, mat1[50][50], mat2[50][50], rest[50][50];
    i = getcolrow(1);
    j = getcolrow(2);
    getdata(mat1,i,j);
    getdata(mat2,i,j);
    multiply(mat1,mat2,rest,i,j);
    printdata(rest,i,j);
    getch();
}

```

```

int getcolrow(int x)
{
    int *p;
    p = &x;
    if( *p == 1 )
    {
        printf("Enter The Rows : ");
    }
    else
    {
        printf("Enter The Columns : ");
    }
    scanf("%d",p);
    return *p;
}

```

```

void getdata(int matrix[][50],int r,int c)
{

```

```

int i,j,(*ptr)[50];

ptr = matrix;

for( i = 0; i < r; i++ )
{
    for( j = 0; j < c; j++ )
    {
        printf("\nEnter The (%d,%d) of matrix : ",i,j);
        scanf("%d",&(*ptr + i) + j));
    }
}
}

```

```

void multiply(int mat1[][50], int mat2[][50], int rest[][50], int r, int c)
{
    int i , j, k, temp=0,(*ptr)[50],(*ptr2)[50],(*ptr3)[50];
    ptr = mat1;
    ptr2 = mat2;
    ptr3 = rest;
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            for(k=0;k<c;k++)
            {
                temp += (*(ptr + i) + k) * (*(ptr2 + k) + j));
            }
            *(*(ptr3 + i) + j) = temp;
            temp=0;
        }
        printf("\n");
    }
}

```

```
}
```

```
void printdata(int rest[][50],int r, int c)
```

```
{
```

```
    int i,j,(*ptr3)[50];
```

```
    ptr3 = rest;
```

```
    for( i = 0; i < r; i++ )
```

```
    {
```

```
        for( j = 0; j < c; j++ )
```

```
        {
```

```
            printf("%3d",*(ptr3 + i) + j));
```

```
        }
```

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

```
    }
```

```
}
```

Output:

Enter The Columns : 3

Enter The (0,0) of matrix : 1

Enter The (0,1) of matrix : 2

Enter The (0,2) of matrix : 3

Enter The (1,0) of matrix : 1

Enter The (1,1) of matrix : 2

Enter The (1,2) of matrix : 3

Enter The (2,0) of matrix : 1

Enter The (2,1) of matrix : 2

Enter The (2,2) of matrix : 3

Enter The (0,0) of matrix : 1

Enter The (0,1) of matrix : 2

Enter The (0,2) of matrix : 3

Enter The (1,0) of matrix : 1

Enter The (1,1) of matrix : 2

Enter The (1,2) of matrix : 3

Enter The (2,0) of matrix : 1

Enter The (2,1) of matrix : 2

Enter The (2,2) of matrix : 3

6 12 18

6 12 18

6 12 18

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*/

```

#include<stdio.h>

#include<conio.h>


int getrowcol(int);

void getdata(int[][50],int,int);

void add(int[][50],int[][50],int[][50],int,int);

void printdata(int[][50],int,int);

void main()
{
    int mat1[50][50],mat2[50][50],rest[50][50],row,col;
    row = getrowcol(1);
    col = getrowcol(2);
    printf("\nMatrix 1\n_____ \n");
    getdata(mat1,row,col);
    printf("\nMatrix 2\n_____ \n");
    getdata(mat2,row,col);
    add(mat1,mat2,rest,row,col);
    printf("\nMatrix 1\n_____ \n");
    printdata(mat1,row,col);
    printf("\nMatrix 2\n_____ \n");
    printdata(mat2,row,col);
    printf("\nResult\n_____ \n");
    printdata(rest,row,col);
    getch();
}


int getrowcol(int x)
{
    int *p;

```

```

p = &x;
if(*p == 1)
{
    printf("\nEnter The Rows For Matrix : ");
}
else
{
    printf("\nEnter The Columns For Matrix : ");
}
scanf("%d",p);
return *p;
}

```

```

void getdata(int matrix[][50],int r,int c)
{
    int i, j, (*ptr)[50];
    ptr = matrix;

    for( i = 0; i < r; i++ )
    {
        for( j = 0; j < c; j++ )
        {
            printf("\nEnter The Value For Matrix[%d][%d] : ",i,j);
            scanf("%d",&(*(ptr + i) + j));
        }
    }
    printf("_____ \n");
}

```

```

void add(int matrix1[][50],int matrix2[][50],int resultmatrix[][50],int r, int c)
{

```

```

int i, j, (*ptr)[50],(*ptr2)[50],(*ptr3)[50];

ptr = matrix1;

ptr2 = matrix2;

ptr3 = resultmatrix;

for( i = 0; i < r; i++ )
{
    for( j = 0; j < c; j++ )
    {
        *(*ptr3 + i) + j) = (*(*ptr + i) + j)) + (*(*ptr2 + i) + j));
    }
}

```

```

void printdata(int matrix[][50],int r, int c)
{
    int i, j, (*ptr)[50];

    ptr = matrix;

    for( i = 0; i < r; i++ )
    {
        for( j = 0; j < c; j++ )
        {
            printf("%3d", *(*ptr + i) + j));

        }
        printf("\n");
    }
}

```

Output :

Enter The Rows For Matrix : 3



Enter The Columns For Matrix : 3

Matrix 1

---

Enter The Value For Matrix[0][0] : 2

Enter The Value For Matrix[0][1] : 5

Enter The Value For Matrix[0][2] : 1

Enter The Value For Matrix[1][0] : 7

Enter The Value For Matrix[1][1] : 9

Enter The Value For Matrix[1][2] : 3

Enter The Value For Matrix[2][0] : 5

Enter The Value For Matrix[2][1] : 7

Enter The Value For Matrix[2][2] : 8

---

Matrix 2

---

Enter The Value For Matrix[0][0] : 2

Enter The Value For Matrix[0][1] : 4

Enter The Value For Matrix[0][2] : 7

Enter The Value For Matrix[1][0] : 5

Enter The Value For Matrix[1][1] : 8

Enter The Value For Matrix[1][2] : 0

Enter The Value For Matrix[2][0] : 3

Enter The Value For Matrix[2][1] : 5

Enter The Value For Matrix[2][2] : 2

---

Matrix 1

---

2 5 1

7 9 3

5 7 8

Matrix 2

---

2 4 7

5 8 0

3 5 2

Result

---

4 9 8

12 17 3

8 12 10

```
*****  
*****  
*****/
```

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

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

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

```
int getlist();
```

```
void getdata(char[][15],int);
```

```
void sort(char[][15],int);
```

```
void printdata(char[][15],int);
```

```
void main()
```

```
{
```

```
    char string[10][15],list;
```

```
    list = getlist();
```

```
    getdata(string,list);
```

```
    sort(string,list);
```

```
    printdata(string,list);
```

```
    getch();
```

```
}
```

```
int getlist()
```

```
{
```

```
    int x;
```

```
    printf("Enter The List Strings Count : ");
```

```
    scanf("%d",&x);
```

```

        return x;
    }

void getdata(char s[][15],int list)
{
    int i, j;
    char (*ptr)[15];
    ptr = s;
    for( i = 0; i < list; i++)
    {
        scanf("%s",*ptr);
        ptr++;
    }
}

```

```

void sort(char s[][15],int list)
{
    int i, j,check = 0;
    char (*ptr)[15],temp[15];
    ptr = s;
    for( i = 0; i < list - 1; i++ )
    {
        for( j = 0; j < list - i - 1; j++ )
        {
            check = strcmp(*(ptr + j), *(ptr + j + 1));
            if(check > 0)
            {
                strcpy(temp,*(ptr + j));
                strcpy(*(ptr + j),*(ptr + j + 1));
                strcpy(*(ptr + j + 1),temp);
            }
        }
    }
}

```

```

        }
    }
}

```

```
void printdata(char s[][15],int list)
```

```

{
    int i, j;
    char (*ptr)[15];
    ptr = s;
    printf("\n\n_____After Sorting_____\\n");
    for(i = 0; i < list; i++)
    {
        printf("%s\\n",*ptr);
        ptr++;
    }
}

```

Output :

Enter The List Strings Count : 5

pradip

lakshya

ajinkya

nirav

milind

\_\_\_\_\_After Sorting\_\_\_\_\_

ajinkya

lakshya

milind

nirav

pradip

```
*****  
*****  
*****/
```

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

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

```
int getlimit(int);
```

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

```
int setdata(int [],int,int);
```

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

```
void main()
```

```
{
```

```
    int arr[50],limit,setme;
```

```
    limit = getlimit(1);
```

```
    getdata(arr,limit);
```

```
    setme = getlimit(2);
```

```
    limit = setdata(arr,limit,setme);
```

```
    printdata(arr,limit);
```

```
    getch();
```

```
}
```

```
int getlimit(int i)
```

```
{
```

```
    int n;
```

```
    if( i == 1 )
```

```
{
```

```
    printf("Enter The Total Number You Want in Array : ");
```

```
}  
  
else  
  
{  
    printf("Enter The Number You Want to set in Array : ");  
}  
  
scanf("%d",&n);  
return n;  
}
```

```
void getdata(int arr[],int l)  
{  
    int i,*ptr;  
    ptr = arr;  
    for( i = 0; i < l; i++,ptr++ )  
    {  
        scanf("%d",ptr);  
    }  
}
```

```
int setdata(int arr[],int l,int set)  
{  
    int i,*ptr,temp = 0;  
    ptr = arr;  
  
    for(i=0;i<l;i++,ptr++)  
    {  
        if(set <= *ptr)  
        {  
            temp = *ptr;  
            *ptr = set;
```

```

        ++ptr;
        while( *ptr != '\0')
        {
            set = temp;
            temp = *ptr;
            *ptr = set;
            ptr++;
        }
        *ptr = temp;
        l = l+1;
        i = l;
    }
}
return l;
}

```

```

void printdata(int arr[],int l)
{
    int i, *ptr;
    ptr = arr;
    for ( i = 0; i < l; i++,ptr++)
    {
        printf("%d\n",*ptr);
    }
}

```

Output :

Enter The Total Number You Want in Array : 5

12

15



18

21

25

Enter The Number You Want to set in Array : 19

12

15

18

19

21

25

```
*****  
*****  
*****/
```

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

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

```
void getinput(float *);
```

```
int getdecimal();
```

```
void print(float *,int);
```

```
void main()
```

```
{
```

```
    int decimal = 0;
```

```
    float num,*ptr;
```

```
    ptr = &num;
```

```
    getinput(ptr);
```

```
    decimal = getdecimal();
```

```
    print(ptr,decimal);
```

```

    getch();
}

void getinput(float *ptr)
{
    printf("Enter The Decimal Number : ");
    scanf("%f",ptr);
}

int getdecimal()
{
    int i = 0;
    printf("Enter How Many Decimal Point You Want : ");
    scanf("%d",&i);
    return i;
}

void print(float *ptr,int i)
{
    printf("%.*f",i,*ptr);
}

```

Output :

Enter The Decimal Number : 15.7896

Enter How Many Decimal Point You Want : 2

15.79

Enter The Decimal Number : 27.4567

Enter How Many Decimal Point You Want : 3

27.457

```
*****
*****
*****/
```

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

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

```
void getnumber(int *);
```

```
void exchange(int *, int *);
```

```
void print(int *);
```

```
void main()
```

```
{
```

```
    int a,*ptr,a,b,*ptrb;
```

```
    ptr = &a;
```

```
    ptrb = &b;
```

```
    getnumber(ptr);
```

```
    getnumber(ptrb);
```

```
    printf("\nValue : %d At Address : %p",*ptr,ptr);
```

```
    printf("\nValue : %d At Address : %p\n",*ptrb,ptrb);
```

```
    exchange(ptr,ptrb);
```

```
    printf("\nAfter Exchange Values \n");
```

```
    printf("\nValue : %d At Address : %p",*ptr,ptr);
```

```
    printf("\nValue : %d At Address : %p\n",*ptrb,ptrb);
```

```
}
```

```
void getnumber(int *ptr)
```

```
{
```

```
    printf("\nEnter Number : ");
```

```
    scanf("%d",ptr);
```

```
}
```

```
void exchange(int *ptra, int *ptrb)
{
    *ptra = *ptra + *ptrb;
    *ptrb = *ptra - *ptrb;
    *ptra = *ptra - *ptrb;
}
```

```
void print(int *ptr)
{
    printf("%d\n", *ptr);
}
```

Output :

Enter Number : 45

Enter Number : 78

Value : 45 At Address : 000000000061FE0C

Value : 78 At Address : 000000000061FE08

After Exchange Values

Value : 78 At Address : 000000000061FE0C

Value : 45 At Address : 000000000061FE08

```
*****
*****
*****/
```

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

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

```
void getstring(char *);
```

```
char finding();
```

```
int occurance(char *,char);
```

```
void print(int);
```

```
void main()
```

```
{
```

```
    char str[15],*ptr,findme;
```

```
    int index = 0;
```

```
    ptr = str;
```

```
    getstring(ptr);
```

```
    findme = finding();
```

```
    index = occurance(ptr,findme);
```

```
    print(index);
```

```
}
```

```
void getstring(char *ptr)
```

```
{
```

```
    printf("Enter the String : ");
```

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

```
}
```

```
char finding()
```

```
{
```

```
    char s;
```

```
    printf("Enter The Character You Want Find : ");
```

```
    scanf(" %c",&s);
```

```
    return s;
```

```
}
```

```

int occurance(char *ptr,char s)
{
    int index = 0,flag = 0;
    while(*ptr != '\0' && flag != 1)
    {
        if( *ptr == s )
        {
            flag = 1;
        }
        else
        {
            index++;
            ptr++;
        }
    }
    if( flag == 1 )
    {
        return index;
    }
    else
    {
        return index = 101;
    }
}

```

```

void print(int index)
{
    if (index == 101)
    {

```

```

        printf("No Match Found");
    }
    else
    {
        printf("Found At Index : %d",index);
    }
}

```

Output :

Enter the String : pradip

Enter The Character You Want Find : p

Found At Index : 0

Enter the String : pradip

Enter The Character You Want Find : a

Found At Index : 2

```

*****
*****
*****/

```

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

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

```
void getstring(char *);
```

```
int checksubset(char *,char *);
```

```
void print(int);
```

```
void main()
```

```
{
```

```
    int index;
```

```

char mainstr[15], str[10], *mainptr, *ptr;

mainptr = mainstr;

ptr = str;

printf( "Enter The Main String : " );

getstring(mainstr);

printf( "Enter The Sub String : " );

getstring(ptr);

index = checksubset(mainptr,ptr);

print(index);
}

```

```

void getstring(char *ptr)
{
    scanf("%s",ptr);
}

```

```

int checksubset(char *main,char *sub)
{
    int flag = 0, index = 0, subindex = 0, returnvalue = 404;
    char *temp;
    temp = sub;
    while( *main != '\0' && flag != 1 )
    {
        if( *main == *sub )
        {
            index++;
            sub++;
            main++;
            subindex++;
            if(*sub == '\0')
            {

```



```

        flag = 1;

        returnvalue = index - subindex;
    }
}
else if( *main != *sub )
{
    index++;
    main++;
    subindex = 0;
    sub = temp;
}
}
return returnvalue;
}

```

```

void print(int n)
{
    if(n != 404)
    {
        printf( "\nString Found At Index : %d" ,n);
    }
    else
    {
        printf( "\nSubstring Not Found\n" );
    }
}

```

Output:

Enter The Main String : Pradip

Enter The Sub String : Karmakar

Substring Not Found

PS E:\MCA\MCA SEM 2\AP\Assignment 2\XT11\XT11\_b> ./XT11\_b.exe

Enter The Main String : Karmakar

Enter The Sub String : mak

String Found At Index : 3

PS E:\MCA\MCA SEM 2\AP\Assignment 2\XT11\XT11\_b> ./XT11\_b.exe

Enter The Main String : Karmakar

Enter The Sub String : rma

String Found At Index : 2

```
*****  
*****  
*****/
```

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

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

```
void getstring(char *);
```

```
char getchart();
```

```
void deleting(char *,char);
```

```
void main()
```

```
{
```

```
    char str[10], delete, *strptr;
```

```
    strptr = str;
```

```
    getstring(strptr);
```

```
    delete = getchart();
```

```
    deleting(strptr,delete);
```

```
}
```

```
void getstring(char *s)
```

```
{  
    printf( "Enter The String : " );  
    scanf("%s",s);  
}
```

```
char getchart()
```

```
{  
    char d;  
    printf( "Enter The Deletion Character : " );  
    scanf(" %c",&d);  
    return d;  
}
```

```
void deleting(char *s,char d)
```

```
{  
    printf("\nYour String : %s",s);  
    printf("\nCharacter to be Deleted : %c",d);  
    char *trap,*store;  
    store = s;  
    while( *s != '\0' )  
    {  
        if( *s == d )  
        {  
            trap = s;  
            while( *trap != '\0' )  
            {  
                *trap = *(trap + 1);  
                trap++;  
            }  
        }  
    }
```

```

    }
    s++;
}
s = store;
printf("\nResult : %s",s);
}

```

Output:

Enter The String : Pradip

Enter The Deletion Character : a

Your String : Pradip

Character to be Deleted : a

Result : Prdip

```

*****
*****
*****/

```

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

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

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

```
void getstring(char *);
```

```
int getlenstr2(char *);
```

```
void stringremover(char *, char *,int);
```

```
void main()
```

```
{
```

```
    int length = 0;
```

```
    char string1[20],string2[10],*strptr1,*strptr2;
```

```
    strptr1 = string1;
    strptr2 = string2;
    printf("Enter The First String : ");
    getstring(strptr1);
    printf("Enter The Second String : ");
    getstring(strptr2);
    length = getlenstr2(strptr2);
    stringremover(strptr1,strptr2,length);

}
```

```
void getstring(char *string)
{
    scanf("%s",string);
}
```

```
int getlenstr2(char *s2)
{
    int count = 0;
    while( *s2 != '\0' )
    {
        ++count;
        s2++;
    }
    return count;
}
```

```
void stringremover(char *s1, char *s2, int length_s2)
{
    char *string1,*string2,*temp;
    temp = s1;
```

```

string2 = s2;
while( *s1 != '\0' )
{
    if( *s1 == *s2 )
    {
        string1 = s1;
        while( *string2 != '\0' && *string1 == *string2 )
        {
            string2++;
            string1++;
        }
        if( *string2 == '\0' )
        {
            string2 = s2;
            while( *s1 != '\0' )
            {
                *s1 = *(s1 + length_s2);
                s1++;
            }
            s1 = temp;
        }
        else
        {
            string2 = s2;
        }
    }
    s1++;
}
printf("%s",temp);
}

```

Output:

Enter The First String : karmakar

Enter The Second String : kar

ma

```
*****  
*****  
*****/
```

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

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

```
void getstring(char *);
```

```
char getchart();
```

```
void deleting(char *,char);
```

```
void main()
```

```
{
```

```
    char str[10], delete, *strptr;
```

```
    strptr = str;
```

```
    getstring(strptr);
```

```
    delete = getchart();
```

```
    deleting(strptr,delete);
```

```
}
```

```
void getstring(char *s)
```

```
{
```

```
    printf( "Enter The String : " );
```

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

```
}
```

```

char getchart()
{
    char d;

    printf( "Enter The Deletion Character : " );

    scanf(" %c",&d);

    return d;
}

```

```

void deleting(char *s,char d)
{
    printf("\n%s",s);
    printf("\n%c",d);
    char *trap,*store;
    store = s;
    while( *s != '\0' )
    {
        if( *s == d || *s == d - 32 || *s == d + 32 )
        {
            trap = s;
            while( *trap != '\0' )
            {
                *trap = *(trap + 1);
                trap++;
            }
        }
        s++;
    }
    s = store;
    printf("\n%s",s);
}

```



Output:

Enter The String : Pradip

Enter The Deletion Character : p

Pradip

p

radi

```
*****
*****
*****/
```

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

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

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

```
void getstring(char *);
```

```
int getlenstr2(char *);
```

```
void stringremover(char *, char *,int);
```

```
char upper(char);
```

```
char lower(char);
```

```
void main()
```

```
{
```

```
    int length = 0;
```

```
    char string1[20],string2[10],*strptr1,*strptr2;
```

```
    strptr1 = string1;
```

```
    strptr2 = string2;
```

```
    printf("Enter The First String : ");
```

```
    getstring(strptr1);
```

```
    printf("Enter The Second String : ");  
    getstring(strptr2);  
    length = getlenstr2(strptr2);  
    stringremover(strptr1,strptr2,length);  
  
}
```

```
void getstring(char *string)  
{  
    scanf("%s",string);  
}
```

```
int getlenstr2(char *s2)  
{  
    int count = 0;  
    while( *s2 != '\0' )  
    {  
        ++count;  
        s2++;  
    }  
    return count;  
}
```

```
void stringremover(char *s1, char *s2, int length_s2)  
{  
    int flag = 0;  
    char *string1,*string2,*temp;  
    temp = s1;  
    string2 = s2;  
    while( *s1 != '\0' && flag != 1 )  
    {
```

```

if( *s1 == *s2 || *s1 == (*s2 + 32) || *s1 == (*s2 - 32) )
{
    string1 = s1;
    while( *string2 != '\0' && (*string1 == *string2 || *string1 == (*string2 + 32) || *string1 ==
(*string2 - 32) ) )
    {
        string2++;
        string1++;
    }
    if( *string2 == '\0' )
    {
        string2 = s2;
        while( *s1 != '\0')
        {
            *s1 = *(s1 + length_s2);
            s1++;
        }
        s1 = temp;
    }
    else
    {
        string2 = s2;
    }
}
s1++;
}
printf("%s",temp);
}

```

Output:

Enter The First String : rollwalacomputerrollwala

Enter The Second String : wala

rollcomputerroll

Enter The First String : rollwalacomputerrollWALA

Enter The Second String : wAla

rollcomputerroll

```
*****  
*****  
*****/
```

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

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

```
void getstr(char *);
```

```
void copy(char *,char *);
```

```
void main()
```

```
{
```

```
    char string1[20], string2[10],*ptrstr1, *ptrstr2;
```

```
    ptrstr1 = string1;
```

```
    ptrstr2 = string2;
```

```
    printf("\nEnter The Main String : ");
```

```
    getstr(ptrstr1);
```

```
    printf("\nEnter The String to be Copy : ");
```

```
    getstr(ptrstr2);
```

```
    copy(ptrstr1,ptrstr2);
```

```
}
```

```

void getstr(char *s)
{
    scanf("%s",s);
}

void copy(char *s1,char *s2)
{
    char *temp;
    temp = s1;
    while( *s2 != '\0' )
    {
        *s1 = *s2;
        s2++;
        s1++;
    }
    *s1 = '\0';
    printf("\nAfter Copy To The First String : %s",temp);
}

```

Output:

Enter The Main String : Pradip

Enter The String to be Copy : karmakar

After Copy To The First String : karmakar

```

*****
*****
*****/

```

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

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

```
void getstr(char *);
```

```
int getnchar();
```

```
void copy(char *,char *,int);
```

```
void main()
```

```
{
```

```
    char string1[20], string2[10],*ptrstr1, *ptrstr2;
```

```
    int limit = 0;
```

```
    ptrstr1 = string1;
```

```
    ptrstr2 = string2;
```

```
    printf("\nEnter The Main String : ");
```

```
    getstr(ptrstr1);
```

```
    printf("\nEnter The String to be Append : ");
```

```
    getstr(ptrstr2);
```

```
    limit = getnchar();
```

```
    copy(ptrstr1,ptrstr2,limit);
```

```
}
```

```
void getstr(char *s)
```

```
{
```

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

```
}
```

```
int getnchar()
```

```
{
```

```
    int d;
```

```
    printf("How Many Char You Want to Append from String 2 : ");  
    scanf("%d",&d);  
    return d;  
}
```

```
void copy(char *s1, char *s2, int i)  
{  
    char *temp;  
    temp = s1;  
  
    while( i > 0 && *s2 != '\0')  
    {  
        *s1 = *s2;  
        i--;  
        s1++;  
        s2++;  
    }  
  
    *s1 = '\0';  
    printf("\n%s",temp);  
}
```

Output:

Enter The Main String : Karmakar

Enter The String to be Append : Pradip

How Many Char You Want to Append from String 2 : 4

Prad

```
*****
*****
*****/
```

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

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

```
void getstring(char *);
```

```
int getlength(char *);
```

```
void toggle(char *);
```

```
void main()
```

```
{
```

```
    char string[20],*ptrstr;
```

```
    int length = 0;
```

```
    ptrstr = string;
```

```
    getstring(ptrstr);
```

```
    length = getlength(ptrstr);
```

```
    printf("Length of The String Is : %d\n",length);
```

```
    toggle(ptrstr);
```

```
}
```

```
void getstring(char *s)
```

```
{
```

```
    printf("\nEnter The String : ");
```

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

```
}
```

```
int getlength(char *s)
```

```
{
```

```
    int i = 0;
```

```
    while( *s != '\0' )
```



```

{
    s++;

    i++;
}
return i;
}

void toggle(char *s)
{
    char *temp;
    temp = s;
    while( *s != '\0' )
    {
        if( *s > 64 && *s < 91 )
        {
            *s += 32;
        }
        else if( *s > 96 && *s < 123 )
        {
            *s -= 32;
        }
        s++;
    }
    printf("\nAfter Toggle : %s",temp);
}

```

Output:

Enter The String : Pradip

Length of The String Is : 6

After Toggle : pRADIP

```
*****  
*****  
*****/
```

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

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

```
void getdata(char *);
```

```
void upper(char *);
```

```
void main()
```

```
{
```

```
    char string[10],*ptrstr;
```

```
    ptrstr = string;
```

```
    getdata(ptrstr);
```

```
    upper(ptrstr);
```

```
}
```

```
void getdata(char *s)
```

```
{
```

```
    printf("\nEnter the string : ");
```

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

```
}
```

```
void upper(char *s)
```

```
{
```

```
    char *temp;
```

```
    temp = s;
```

```
    while( *s != '\0' )
```

```
    {
```

```

        if( *s > 96 && *s < 123 )
        {
            *s -= 32;
        }

        s++;
    }

    printf("\n%s",temp);
}

```

Output:

Enter the string : Pradip

PRADIP

```

*****
*****
*****/

```

```

#include<stdio.h>

```

```

#include<conio.h>

```

```

void getdata(char *);

```

```

void upper(char *);

```

```

void main()

```

```

{

```

```

    char string[10],*ptrstr;

```

```

    ptrstr = string;

```

```

    getdata(ptrstr);

```

```

    upper(ptrstr);

```

```

}

```

```

void getdata(char *s)
{
    printf("\nEnter the string : ");
    scanf("%s",s);
}

```

```

void upper(char *s)
{
    char *temp;
    temp = s;
    while( *s != '\0' )
    {
        if( *s > 64 && *s < 91 )
        {
            *s += 32;
        }
        s++;
    }
    printf("\n%s",temp);
}

```

Output:

Enter the string : PRAdip

pradip

```

*****
*****
*****/

```

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

```
int getlist();
void getdata(char[][15],int);
void sort(char[][15],int);
void printdata(char[][15],int);
```

```
void main()
{
    char string[10][15],list;

    list = getlist();
    getdata(string,list);
    sort(string,list);
    printdata(string,list);
    getch();
}
```

```
int getlist()
{
    int x;

    printf("Enter The List Strings Count : ");
    scanf("%d",&x);

    return x;
}
```

```
void getdata(char s[][15],int list)
{
```

```

int i, j;

char (*ptr)[15];

ptr = s;

for( i = 0; i < list; i++)
{
    scanf("%s",*ptr);
    ptr++;
}
}

```

```

void sort(char s[][15],int list)
{
    int i, j,check = 0;
    char (*ptr)[15],temp[15];
    ptr = s;
    for( i = 0; i < list - 1; i++ )
    {
        for( j = 0; j < list - i - 1; j++ )
        {
            check = strcmp(*(ptr + j), *(ptr + j + 1));
            if(check > 0)
            {
                strcpy(temp,*(ptr + j));
                strcpy(*(ptr + j),*(ptr + j + 1));
                strcpy(*(ptr + j + 1),temp);
            }
        }
    }
}
}

```

```

void printdata(char s[][15],int list)

```

```

{
    int i, j;
    char (*ptr)[15];
    ptr = s;
    printf("\n\n_____After Sorting_____\\n");
    for(i = 0; i < list; i++)
    {
        printf("%s\\n",*ptr);
        ptr++;
    }
}

```

Output:

Enter The List Strings Count : 4

pradip

ajinkya

nirav

lakshya

\_\_\_\_\_After Sorting\_\_\_\_\_

ajinkya

lakshya

nirav

pradip

```

*****
*****
*****/

```

#include<stdio.h>

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

```
void getstr(char *);
```

```
void append(char *,char *);
```

```
void main()
```

```
{
```

```
    char string1[20], string2[10],*ptrstr1, *ptrstr2;
```

```
    ptrstr1 = string1;
```

```
    ptrstr2 = string2;
```

```
    printf("\nEnter The Main String : ");
```

```
    getstr(ptrstr1);
```

```
    printf("\nEnter The String to be Append : ");
```

```
    getstr(ptrstr2);
```

```
    append(ptrstr1,ptrstr2);
```

```
}
```

```
void getstr(char *s)
```

```
{
```

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

```
}
```

```
void append(char *s1,char *s2)
```

```
{
```

```
    char *temp;
```

```
    temp = s1;
```

```
    while( *s1 != '\0' )
```

```
    {
```

```
        s1++;
```

```
    }
```



```

while( *s2 != '\0' )
{
    *s1 = *s2;

    s2++;

    s1++;
}

printf("\n %s",temp);
}

```

Output:

Enter The Main String : Pradip

Enter The String to be Append : Karmakar

PradipKarmakar

```

*****
*****
*****/

```

```

#include<stdio.h>

```

```

#include<conio.h>

```

```

void getstr(char *);

```

```

int getnchar();

```

```

void append(char *,char *,int);

```

```

void main()

```

```

{

```

```

char string1[20], string2[10], *ptrstr1, *ptrstr2;

int limit = 0;

ptrstr1 = string1;

ptrstr2 = string2;

printf("\nEnter The Main String : ");

getstr(ptrstr1);

printf("\nEnter The String to be Append : ");

getstr(ptrstr2);

limit = getnchar();

append(ptrstr1, ptrstr2, limit);

printf("\n%s", string1);
}

```

```

void getstr(char *s)
{
    scanf("%s", s);
}

```

```

int getnchar()
{
    int d;

    printf("How Many Char You Want to Append from String 2 : ");

    scanf("%d", &d);

    return d;
}

```

```

void append(char *s1, char *s2, int i)
{
    char *temp;

    temp = s1;

    while( *s1 != '\0' )

```

```

{
    s1++;
}

while( i > 0 && *s2 != '\0')
{
    *s1 = *s2;
    i--;
    s1++;
    s2++;
}
*s1 = '\0';
}

```

Output:

Enter The Main String : Pradip

Enter The String to be Append : Karmakar

How Many Char You Want to Append from String 2 : 5

PradipKarma

```

*****
*****
*****/

```

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

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

```
void getstr( char * );
```

```
void reverse( char * );
```

```
int length( char * );
```

```
void main()
```

```
{
```

```
    char string[20],*ptrstr;
```

```
    ptrstr = string;
```

```
    getstr(ptrstr);
```

```
    reverse(ptrstr);
```

```
}
```

```
void getstr( char *s )
```

```
{
```

```
    printf("Enter The String : ");
```

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

```
}
```

```
void reverse( char *s )
```

```
{
```

```
    int leng = length(s),endlength = leng/2;
```

```
    char *tempstr, tempchar, *s1;
```

```
    tempstr = s;
```

```
    s1 = s+(leng-1);
```

```
    while( endlength > 0 )
```

```
    {
```

```
        tempchar = *s;
```

```
        *s = *s1;
```

```
        *s1 = tempchar;
```

```
        endlength--;
```

```
        s++;
```

```
        s1--;
```

```

    }
    printf("%s",tempstr);
}

```

```

int length( char *s )
{
    int len = 0;
    while( *s != '\0' )
    {
        len++;
        s++;
    }
    return len;
}

```

Output:

Enter The String : Pradip

pidarP

```

*****
*****
*****/

```

```

#include<stdio.h>

```

```

#include<conio.h>

```

```

void getstring(char *);

```

```

int comparestring(char *,char *);

```

```

void main()

```

```

{
    char string1[20],string2[20],*ptrstr1,*ptrstr2;
    int rtnvalue = 0;
    ptrstr1 = string1;
    ptrstr2 = string2;
    printf("Enter The First String : ");
    getstring(ptrstr1);
    printf("Enter The Second String : ");
    getstring(ptrstr2);
    rtnvalue = comparestring(ptrstr1,ptrstr2);
    printf("\n%d",rtnvalue);
}

```

```

void getstring(char *s)

```

```

{
    scanf("%s",s);
}

```

```

int comparestring(char *s1,char *s2)

```

```

{
    int value = 0;
    while( value == 0 && (*s1 != '\0' || *s2 != '\0') )
    {
        if( *s1 > *s2 )
        {
            value = 1;
        }
        else if( *s1 < *s2 )
        {
            value = -1;
        }
    }
}

```

```

        s1++;

        s2++;
    }
    return value;
}

```

Output:

Enter The First String : Pradip

Enter The Second String : Karmakar

1

Enter The First String : karmakar

Enter The Second String : pradip

-1

Enter The First String : pradip

Enter The Second String : pradip

0

```

*****
*****
*****/

```

```

#include<stdio.h>

```

```

#include<conio.h>

```

```

void getstring(char *);

```

```

int comparestring(char *,char *);

```

```

void main()
{
    char string1[20],string2[20],*ptrstr1,*ptrstr2;
    int rtnvalue = 0;
    ptrstr1 = string1;
    ptrstr2 = string2;
    printf("Enter The First String : ");
    getstring(ptrstr1);
    printf("Enter The Second String : ");
    getstring(ptrstr2);
    rtnvalue = comparestring(ptrstr1,ptrstr2);
    printf("\n%d",rtnvalue);
}

```

```

void getstring(char *s)
{
    scanf("%s",s);
}

```

```

int comparestring(char *s1,char *s2)
{
    int value = 0;
    while( value == 0 && (*s1 != '\0' || *s2 != '\0') )
    {
        if( *s1 > 64 && *s1 < 91 )
        {
            *s1 += 32;
        }
        if( *s2 > 64 && *s2 < 91 )
        {
            *s2 += 32;
        }
    }
}

```



```
}  
if( *s1 > * s2 )  
{  
    value = 1;  
}  
else if( *s1 < *s2 )  
{  
    value = -1;  
}  
s1++;  
s2++;  
}  
return value;  
}
```

Output:

Enter The First String : pradip

Enter The Second String : Karmakar

1

Enter The First String : PrADIP

Enter The Second String : pradip

0

Enter The First String : Karmakar

Enter The Second String : Pradip

-1

```
*****
*****
*****/
```

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

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

```
void getstring(char *);
```

```
int getnvalue();
```

```
int comparestring(char *,char *,int);
```

```
void main()
```

```
{
```

```
    char string1[20],string2[20],*ptrstr1,*ptrstr2;
```

```
    int rtnvalue = 0, n = 0;
```

```
    ptrstr1 = string1;
```

```
    ptrstr2 = string2;
```

```
    printf("Enter The First String : ");
```

```
    getstring(ptrstr1);
```

```
    printf("Enter The Second String : ");
```

```
    getstring(ptrstr2);
```

```
    n = getnvalue();
```

```
    rtnvalue = comparestring(ptrstr1,ptrstr2,n);
```

```
    printf("\n%d",rtnvalue);
```

```
}
```

```
void getstring(char *s)
```

```
{
```

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

```
}
```

```

int getnvalue()
{
    int i;

    printf("Enter the Limit for Comparing : ");

    scanf("%d",&i);

    return i;
}

int comparestring(char *s1,char *s2, int i)
{
    int value = 0;

    while( value == 0 && i > 0 && (*s1 != '\0' || *s2 != '\0'))
    {
        if( *s1 > *s2 )
        {
            value = 1;
        }

        else if( *s1 < *s2 )
        {
            value = -1;
        }

        s1++;
        s2++;
        i--;
    }

    return value;
}

```

Output:

Enter The First String : Pradip

Enter The Second String : Pradip

Enter the Limit for Comparing : 4

0

Enter The First String : Karmakar

Enter The Second String : Karnakar

Enter the Limit for Comparing : 4

-1

Enter The First String : Karnakar

Enter The Second String : Karmakar

Enter the Limit for Comparing : 4

1

```
*****
*****
*****/
```

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

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

```
void getstring(char *);
```

```
int getnvalue();
```

```
int comparestring(char *,char *,int);
```

```
void main()
```

```
{
```

```
    char string1[20],string2[20],*ptrstr1,*ptrstr2;
```

```

int rtnvalue = 0, n = 0;

ptrstr1 = string1;
ptrstr2 = string2;

printf("Enter The First String : ");
getstring(ptrstr1);
printf("Enter The Second String : ");
getstring(ptrstr2);
n = getnvalue();
rtnvalue = comparestring(ptrstr1,ptrstr2,n);
printf("\n%d",rtnvalue);
}

```

```

void getstring(char *s)
{
    scanf("%s",s);
}

```

```

int getnvalue()
{
    int i;
    printf("Enter the Limit for Comparing : ");
    scanf("%d",&i);
    return i;
}

```

```

int comparestring(char *s1,char *s2, int i)
{
    int value = 0;
    while( value == 0 && (*s1 != '\0' || *s2 != '\0') && i > 0 )
    {
        if( *s1 > 64 && *s1 < 91 )

```

```

{
    *s1 += 32;
}
if( *s2 > 64 && *s2 < 91 )
{
    *s2 += 32;
}
if( *s1 > *s2 )
{
    value = 1;
}
else if( *s1 < *s2 )
{
    value = -1;
}
s1++;
s2++;
i--;
}
return value;
}

```

Output:

Enter The First String : PrADip

Enter The Second String : pradip

Enter the Limit for Comparing : 5

0

Enter The First String : KarNaKar

Enter The Second String : KarMakar

Enter the Limit for Comparing : 5

1

Enter The First String : KArmAkar

Enter The Second String : KarNAkar

Enter the Limit for Comparing : 5

-1

# Assignment 3

```
*****
*****
*****//*****
*****
```

Name : Pradip . S . Karmakar

Roll-No : 10

Class : MCA-2

Subject : Advanced Programming

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

Questions : Write a modular C program to create a singly linked list & Display In  
FIFO Pattern.

```
*****
*****/
```

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

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

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

```
// Structure declaration
```

```
struct node{
```

```
    int data;
```

```
    struct node *next;
```

```
};
```

```
// function declarations
```



```
int ask_selection();

void menu( struct node *, struct node * );

struct node * get_link_list( struct node *, struct node * );

int get_input();

void display(struct node *);
```

```
// main function
```

```
void main()
{
    struct node *new,*head = NULL;
    menu(new,head);
}
```

```
// menu function gives user to choose option from menu
```

```
void menu(struct node *new,struct node *head)
{
    int selection = ask_selection();
    switch (selection)
    {
        case(1):
            head = get_link_list(new,head);
            menu(new,head);
        case(2):
            display(head);
            menu(new,head);
        default:
            exit(0);
    }
}
```

// ask\_selection function will get the user option selected

```
int ask_selection()
{
    int n;
    printf("\n 1 . Input Data To Link-List. \n 2 . Display The Link List in FIFO. \n 3 . Exit. \n");
    scanf(" %d",&n);
    if( n > 0 && n < 4 )
    {
        return n;
    }
    else{
        printf("\n Wrong Selection Please Choose Correct Options. \n");
        ask_selection();
    }
}
```

// get\_link\_list funtion use for inserting elements in linklist from user

```
struct node * get_link_list( struct node *new, struct node *head )
{
    int input = get_input();
    struct node *temp;
    temp = head;
    new = (struct node *)malloc(sizeof(struct node));
    if( head == NULL )
    {
        head = new;
        temp = head;
    }
    else{
```

```

        while( temp->next != NULL )
        {
            temp = temp->next;
        }
    }
    temp->next = new;
    new->data = input;
    new->next = NULL;
    return head;
}

```

// get\_input will take input from user which is insert to link list.

```

int get_input()
{
    int in;
    printf("\n Enter The Number : ");
    scanf("%d",&in);
    return in;
}

```

// display the linklist till now

```

void display(struct node *head)
{
    int count = 0;
    struct node *temp;
    if(head == NULL)
    {
        printf("\nThere Is Nothing To Display.\n");
    }
}

```

```

else
{
    temp = head;
    printf("\nThe List is : \n");
    while(temp->next != NULL)
    {
        printf(" %d => ",temp->data);
        temp = temp->next;
        count++;
    }
    printf(" %d \n Total Data Found : %d \n",temp->data,count+1);
}
}

```

```

/*****
*****

```

OUTPUT:

1 . Input Data To Link-List.

2 . Display The Link List.

3 . Exit.

1

Enter The Number : 3

1 . Input Data To Link-List.

2 . Display The Link List.

3 . Exit.

1

Enter The Number : 6

1 . Input Data To Link-List.

2 . Display The Link List.

3 . Exit.

1

Enter The Number : 8

1 . Input Data To Link-List.

2 . Display The Link List.

3 . Exit.

1

Enter The Number : 2

1 . Input Data To Link-List.

2 . Display The Link List.

3 . Exit.

2

The List is :

3 => 6 => 8 => 2

Total Data Found : 4

1 . Input Data To Link-List.

2 . Display The Link List.

3 . Exit.

```

/*****
*****/

```

Name : Pradip . S . Karmakar

Roll-No : 10

Class : MCA-2

Subject : Advanced Programming

```

*****/

```

Questions : Write a modular C program to create a singly linked list & Display In  
LIFO Pattern.

```

*****/

```

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

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

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

```
// Structure declaration
```

```
struct node{
    int data;
    struct node *next;
};
```

```
// function declarations
```

```
int ask_selection();
```

```
void menu( struct node *, struct node * );
```

```
struct node * get_link_list( struct node *, struct node * );
```

```
int get_input();
```

```
void display(struct node *);
```

```
// main function
```

```
void main()
```

```
{
```

```
    struct node *new,*head = NULL;
```

```
    menu(new,head);
```

```
}
```

```
// menu function gives user to choose option from menu
```

```
void menu(struct node *new,struct node *head)
```

```
{
```

```
    int selection = ask_selection();
```

```
    switch (selection)
```

```
    {
```

```
    case(1):
```

```
        head = get_link_list(new,head);
```

```
        menu(new,head);
```

```
    case(2):
```

```
        display(head);
```

```
        menu(new,head);
```

```
    default:
```

```
        exit(0);
```

```
    }
```

```
}
```

```
// ask_selection function will get the user option selected
```

```

int ask_selection()
{
    int n;

    printf("\n 1 . Input Data To Link-List. \n 2 . Display The Link List in LIFO. \n 3 . Exit. \n");

    scanf(" %d",&n);

    if( n > 0 && n < 4 )
    {
        return n;
    }
    else{
        printf("\n Wrong Selection Please Choose Correct Options. \n");
        ask_selection();
    }
}

```

// get\_link\_list funtion use for inserting elements in linklist from user

```

struct node * get_link_list( struct node *new, struct node *head )
{
    int input = get_input();

    new = (struct node *)malloc(sizeof(struct node));

    new->data = input;

    new->next = head;

    head = new;

    return head;
}

```

// get\_input will take input from user which is insert to link list.

```

int get_input()
{

```



```
int in;

printf("\n Enter The Number : ");

scanf("%d",&in);

return in;

}
```

// display the linklist till now

```
void display(struct node *head)

{
    int count = 0;
    struct node *temp;
    if(head == NULL)
    {
        printf("\nThere Is Nothing To Display.\n");
    }
    else
    {
        temp = head;
        printf("\nThe List is : \n");
        while(temp->next != NULL)
        {
            printf(" %d => ",temp->data);
            temp = temp->next;
            count++;
        }
        printf(" %d \n Total Data Found : %d \n",temp->data, count+1);
    }
}
```

```
/*****  
*****/
```

OUTPUT:

- 1 . Input Data To Link-List.
- 2 . Display The Link List in LIFO.
- 3 . Exit.

1

Enter The Number : 4

- 1 . Input Data To Link-List.
- 2 . Display The Link List in LIFO.
- 3 . Exit.

1

Enter The Number : 7

- 1 . Input Data To Link-List.
- 2 . Display The Link List in LIFO.
- 3 . Exit.

1

Enter The Number : 8

- 1 . Input Data To Link-List.
- 2 . Display The Link List in LIFO.
- 3 . Exit.

1

Enter The Number : 5

- 1 . Input Data To Link-List.
  - 2 . Display The Link List in LIFO.
  - 3 . Exit.
- 1

Enter The Number : 9

- 1 . Input Data To Link-List.
  - 2 . Display The Link List in LIFO.
  - 3 . Exit.
- 2

The List is :

9 => 5 => 8 => 7 => 4

Total Data Found : 5

- 1 . Input Data To Link-List.
- 2 . Display The Link List in LIFO.
- 3 . Exit.

/\*\*\*\*\*  
\*\*\*\*\*

Name : Pradip . S . Karmakar

Roll-No : 10

Class : MCA-2

Subject : Advanced Programming

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

Questions : Write a menu driven program for insert delete and display link list.

```
*****
*****/
```

```
#include<stdio.h> //c standard library
```

```
#include<conio.h> //c input ouput library
```

```
#include<stdlib.h> //c stdlib Library file
```

```
// Structure Declaration
```

```
struct node{
    int data;
    struct node *next;
};
```

```
// Functions Declaration
```

```
void menu(struct node *,struct node *);
int get_n(char);
struct node * insert_beg(struct node *,struct node *,int);
struct node * insert_end(struct node *,struct node *, int);
struct node * insert_atany(struct node *,struct node *, int);
struct node * delete_data(struct node *,struct node *, int);
void display_link(struct node *);
```

```
// Void Main
```

```
void main()
{
    struct node *new;
    struct node *head = NULL;
```

```

    menu(new,head); // Calling menu funtion
}

// menu function
void menu( struct node *new, struct node *head )
{
    int n,getnum;

    printf("\n 1 . Add New Data To Linklist From Begining. \n 2 . Add New Data To Linklist From
Ending.\n 3 . Add New Data To Linklist At Any Place. \n 4 . Delete a Number From The Link-List. \n 5 .
Display LinkList Till Now. \n 6 . Exit. \n");

    scanf("%d",&n);

    // Switch case which check the user input and run specified function
    switch(n)
    {
        case(1):
            getnum = get_n('i');
            head = insert_beg(new,head,getnum); //insertion from begining linklist function call
            menu(new,head); //void menu function call
        case(2):
            getnum = get_n('i');
            head = insert_end(new,head,getnum); //insertion from ending linklist function call
            menu(new,head); //void menu function call
        case(3):
            getnum = get_n('i');
            head = insert_atany(new,head,getnum); //insertion from any point linklist function call
            menu(new,head); //void menu function call
        case(4):
            getnum = get_n('d');
            head = delete_data(new,head,getnum);
            menu(new,head);
        case(5):
            display_link(head); //display linklist function call
    }
}

```

```

        menu(new,head); //void menu function call
case(6):
    exit(0); //exit function call which terminated the program
default:
    printf("\n Please Enter Valid Number.");
    menu(new,head); //void menu function call
    }
}

// function for taking input from user

int get_n(char a)
{
    int n;
    if( a == 'i' )
    {
        printf(" Enter The Number : ");
    }
    else{
        printf(" Enter The Number to Delete : ");
    }
    scanf("%d",&n);
    return n;
}

// function insert_beg, use for linklist begining insertion
struct node * insert_beg( struct node *new, struct node *head,int n )
{
    new = (struct node *)malloc(sizeof(struct node));
    new->data = n;
    new->next = head;

```

```
    head = new;
    return head;
}
```

// function insert\_end, use for linklist ending insertion

```
struct node * insert_end( struct node *new, struct node *head, int n )
```

```
{
    struct node *temp;
    new = (struct node *)malloc(sizeof(struct node));
    if( head == NULL )
    {
        head = new;
        temp = head;
    }
    else{
        temp = head;
        while( temp->next != NULL ) // loop until next has NULL
        {
            temp = temp->next;
        }
    }
    temp->next = new;
    new->data = n;
    new->next = NULL;
    return head;
}
```

// function insert\_atany, use for linklist any-point insertion

```
struct node * insert_atany( struct node *new, struct node *head, int n )
```

```
{
    struct node *first;
```

```

struct node *last;

first = head;

new = (struct node *)malloc(sizeof(struct node));

if( head == NULL || head->data >= n ) // check if head already NULL or input value of user need to
insert at beginning
{
    new->data = n;
    new->next = head;
    head = new;
}
else{
    while( first != NULL && first->data < n ) // loop until user input in greater
    {
        last = first; // store last linklist address
        first = first->next; // store next linklist address
    }
    new->data = n;
    new->next = first;
    last->next = new;
}
return head;
}

```

```

struct node * delete_data( struct node *new, struct node *head,int n )
{
    struct node *temp,*tempstore;
    temp = head;
    if( head == NULL )
    {
        printf("\n There is Nothing To Delete. \n");
    }
}

```



```

else if( temp->data == n )
{
    head = temp->next;
    free(temp);
}
else{
    if( temp->data != n && temp->next == NULL )
    {
        printf("\n No Such Data To Delete. \n");
    }
    else if( temp->data == n && temp->next == NULL )
    {
        free(temp);
        head = NULL;
    }
    else{
        while( temp->next->data != n )
        {
            if( temp->next->next != NULL)
            {
                temp = temp->next;
            }
            else{
                printf("\n No Such Data To Delete. \n");
                menu(new,head);
            }
        }
        tempstore = temp->next;
        temp->next = temp->next->next;
        free(tempstore);
    }
}

```

```

    }

    return head;
}

// function display_link will display the linklist elements
void display_link(struct node *head)
{
    struct node *temp;
    if(head == NULL) // check wheater the head is null
    {
        printf("\nThere Is Nothing To Display.\n");
    }
    else
    {
        temp = head;
        printf("\nThe List is : \n");
        while(temp->next != NULL) // print all the elements from the link-list
        {
            printf(" %d => ",temp->data);
            temp = temp->next;
        }
        printf(" %d \n",temp->data);
    }
}

```

```

/*****
*****

```

OUTPUT:

- 1 . Add New Data To Linklist From Begining.
- 2 . Add New Data To Linklist From Ending.
- 3 . Add New Data To Linklist At Any Place.
- 4 . Delete a Number From The Link-List.
- 5 . Display LinkList Till Now.
- 6 . Exit.

1

Enter The Number : 5

- 1 . Add New Data To Linklist From Begining.
- 2 . Add New Data To Linklist From Ending.
- 3 . Add New Data To Linklist At Any Place.
- 4 . Delete a Number From The Link-List.
- 5 . Display LinkList Till Now.
- 6 . Exit.

1

Enter The Number : 4

- 1 . Add New Data To Linklist From Begining.
- 2 . Add New Data To Linklist From Ending.
- 3 . Add New Data To Linklist At Any Place.
- 4 . Delete a Number From The Link-List.
- 5 . Display LinkList Till Now.
- 6 . Exit.

1

Enter The Number : 2

- 1 . Add New Data To Linklist From Begining.
- 2 . Add New Data To Linklist From Ending.
- 3 . Add New Data To Linklist At Any Place.
- 4 . Delete a Number From The Link-List.

5 . Display LinkList Till Now.

6 . Exit.

2

Enter The Number : 7

1 . Add New Data To Linklist From Begining.

2 . Add New Data To Linklist From Ending.

3 . Add New Data To Linklist At Any Place.

4 . Delete a Number From The Link-List.

5 . Display LinkList Till Now.

6 . Exit.

5

The List is :

2 => 4 => 5 => 7

1 . Add New Data To Linklist From Begining.

2 . Add New Data To Linklist From Ending.

3 . Add New Data To Linklist At Any Place.

4 . Delete a Number From The Link-List.

5 . Display LinkList Till Now.

6 . Exit.

3

Enter The Number : 8

1 . Add New Data To Linklist From Begining.

2 . Add New Data To Linklist From Ending.

3 . Add New Data To Linklist At Any Place.

4 . Delete a Number From The Link-List.

5 . Display LinkList Till Now.

6 . Exit.

5

The List is :

2 => 4 => 5 => 7 => 8

- 1 . Add New Data To Linklist From Begining.
- 2 . Add New Data To Linklist From Ending.
- 3 . Add New Data To Linklist At Any Place.
- 4 . Delete a Number From The Link-List.
- 5 . Display LinkList Till Now.
- 6 . Exit.

3

Enter The Number : 3

- 1 . Add New Data To Linklist From Begining.
- 2 . Add New Data To Linklist From Ending.
- 3 . Add New Data To Linklist At Any Place.
- 4 . Delete a Number From The Link-List.
- 5 . Display LinkList Till Now.
- 6 . Exit.

5

The List is :

2 => 3 => 4 => 5 => 7 => 8

- 1 . Add New Data To Linklist From Begining.
- 2 . Add New Data To Linklist From Ending.
- 3 . Add New Data To Linklist At Any Place.
- 4 . Delete a Number From The Link-List.
- 5 . Display LinkList Till Now.
- 6 . Exit.

4

Enter The Number to Delete : 7

- 1 . Add New Data To Linklist From Begining.
- 2 . Add New Data To Linklist From Ending.
- 3 . Add New Data To Linklist At Any Place.
- 4 . Delete a Number From The Link-List.
- 5 . Display LinkList Till Now.
- 6 . Exit.

5

The List is :

2 => 3 => 4 => 5 => 8

- 1 . Add New Data To Linklist From Begining.
- 2 . Add New Data To Linklist From Ending.
- 3 . Add New Data To Linklist At Any Place.
- 4 . Delete a Number From The Link-List.
- 5 . Display LinkList Till Now.
- 6 . Exit.

4

Enter The Number to Delete : 9

No Such Data To Delete.

- 1 . Add New Data To Linklist From Begining.
- 2 . Add New Data To Linklist From Ending.
- 3 . Add New Data To Linklist At Any Place.
- 4 . Delete a Number From The Link-List.
- 5 . Display LinkList Till Now.
- 6 . Exit.

```

/*****
*****/

```

Name : Pradip . S . Karmakar

Roll-No : 10

Class : MCA-2

Subject : Advanced Programming

```

*****/

```

Questions : Write a C program to create a ordered singly linked list & Display.

```

*****/

```

```

#include<stdio.h>

```

```

#include<conio.h>

```

```

#include<stdlib.h>

```

```

// structure declaration

```

```

struct node{

```

```

    int data;

```

```

    struct node *next;

```

```

};

```

```

// ask_selection function will get the user option selected

```

```

int ask_selection();

```

```

void menu( struct node *, struct node * );

```

```
struct node * get_link_list( struct node *, struct node * );  
int getinput();  
void display(struct node *);
```

```
// main function
```

```
void main()  
{  
    struct node *new,*head = NULL;  
    menu(new,head);  
}
```

```
// menu function give user menu driven outlet
```

```
void menu(struct node *new,struct node *head)  
{  
    int selection = ask_selection();  
    switch (selection)  
    {  
        case(1):  
            head = get_link_list(new,head);  
            menu(new,head);  
        case(2):  
            display(head);  
            menu(new,head);  
        default:  
            exit(0);  
    }  
}
```



```

// ask_selection function will get the user option selected
int ask_selection()
{
    int n;
    printf("\n 1 . Input Data To Link-List. \n 2 . Display The Link List. \n 3 . Exit. \n");
    scanf(" %d",&n);
    if( n > 0 && n < 4 )
    {
        return n;
    }
    else{
        printf("\n Wrong Selection Please Choose Correct Options. \n");
        ask_selection();
    }
}

```

```

// take user input for inserting data
int getinput()
{
    int in;
    printf("\n Enter The Number : ");
    scanf("%d",&in);
    return in;
}

```

```

// insert user data into link list
struct node * get_link_list( struct node *new, struct node *head )
{
    int i,input = getinput();

```

```

struct node *first,*last;

new = (struct node *)malloc(sizeof(struct node));

if(head == NULL || head->data >= input)
{
    new->data = input;
    new->next = head;
    head = new;
}
else{
    first = head;
    while( first != NULL && first->data < input )
    {
        last = first;
        first = first->next;
    }
    new->data = input;
    new->next = first;
    last->next = new;
}
return head;
}

```

```

// display all data available in linklist
void display(struct node *head)
{
    struct node *temp;
    if(head == NULL)
    {
        printf("\nThere Is Nothing To Display.\n");
    }
}

```

```

else
{
    temp = head;
    printf("\nThe List is : \n");
    while(temp->next != NULL)
    {
        printf(" %d => ",temp->data);
        temp = temp->next;
    }
    printf(" %d \n",temp->data);
}
}

```

```

/*****
*****

```

OUTPUT:

- 1 . Input Data To Link-List.
  - 2 . Display The Link List.
  - 3 . Exit.
- 1

Enter The Number : 4

- 1 . Input Data To Link-List.
  - 2 . Display The Link List.
  - 3 . Exit.
- 1

Enter The Number : 3

1 . Input Data To Link-List.

2 . Display The Link List.

3 . Exit.

2

The List is :

3 => 4

1 . Input Data To Link-List.

2 . Display The Link List.

3 . Exit.

1

Enter The Number : 6

1 . Input Data To Link-List.

2 . Display The Link List.

3 . Exit.

2

The List is :

3 => 4 => 6

1 . Input Data To Link-List.

2 . Display The Link List.

3 . Exit.

1

Enter The Number : 5

1 . Input Data To Link-List.

2 . Display The Link List.

3 . Exit.

2

The List is :

3 => 4 => 5 => 6

1 . Input Data To Link-List.

2 . Display The Link List.

3 . Exit.

1

Enter The Number : 2

1 . Input Data To Link-List.

2 . Display The Link List.

3 . Exit.

2

The List is :

2 => 3 => 4 => 5 => 6

1 . Input Data To Link-List.

2 . Display The Link List.

3 . Exit.

1

Enter The Number : 8

- 1 . Input Data To Link-List.
- 2 . Display The Link List.
- 3 . Exit.

2

The List is :

2 => 3 => 4 => 5 => 6 => 8

- 1 . Input Data To Link-List.
- 2 . Display The Link List.
- 3 . Exit.

1

Enter The Number : 7

- 1 . Input Data To Link-List.
- 2 . Display The Link List.
- 3 . Exit.

2

The List is :

2 => 3 => 4 => 5 => 6 => 7 => 8

- 1 . Input Data To Link-List.
- 2 . Display The Link List.
- 3 . Exit.

/\*\*\*\*\*  
\*\*\*\*\*

Name : Pradip . S . Karmakar

Roll-No : 10

Class : MCA-2

Subject : Advanced Programming

\*\*\*\*\*  
\*\*\*\*\*

Questions : Write a modular C program to create a singly linked list in Reversed Order.

& Display.

\*\*\*\*\*  
\*\*\*\*\*/

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

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

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

```
// Structure Declaration
```

```
struct node{
```

```
    int data;
```

```
    struct node *next;
```

```
};
```

```
// Functions Declaration
```

```
void menu( struct node *, struct node * );
```

```
int ask_selection();
```

```
struct node * get_linklist_with_reverse( struct node *, struct node * );
```

```
int get_input();
```

```
int count_linklist( struct node * );
```

```
struct node * reverse_linklist( struct node * );
```

```
void display_linklist(struct node *);
```

```
// Main Function
```

```
void main()
```

```
{
```

```
    struct node *new,*head = NULL;
```

```
    menu(new,head);
```

```
}
```

```
// Menu Function
```

```
void menu(struct node *new,struct node *head)
```

```
{
```

```
    int selection = ask_selection(),linklist_count = 0;
```

```
    switch (selection)
```

```
    {
```

```
    case(1):
```

```
        head = get_linklist_with_reverse(new,head);
```

```
        menu(new,head);
```

```
    case(2):
```

```
        linklist_count = count_linklist(head);
```

```
        if( linklist_count > 1 )
```

```
        {
```

```
            head = reverse_linklist(head);
```

```
            printf("\n Link List Reversed. \n");
```

```
        }
```

```
    else{
```

```
        printf(" \n No Need To Reverse LinkList. \n");
```

```
    }
```

```
    menu(new,head);
```

```
    case(3):
```



```

        display_linklist(head);

        menu(new,head);
default:
        exit(0);
    }
}

```

// ask\_selection Function

```

int ask_selection()
{
    int n;

    printf("\n 1 . Input Data To Link-List. \n 2 . Reverse Link-List. \n 3 . Display The Link List. \n 4 . Exit. \n");

    scanf(" %d",&n);

    if( n > 0 && n < 5 )
    {
        return n;
    }
    else{
        printf("\n Wrong Selection Please Choose Correct Options. \n");
        ask_selection();
    }
}

```

// getting elements to linklist with reverse functionality Function

```

struct node * get_linklist_with_reverse( struct node *new, struct node *head )
{
    int i,input = get_input();

    new = (struct node *)malloc(sizeof(struct node));

```

```
new->data = input;
new->next = head;
head = new;
return head;
}
```

// Function for getting input from user

```
int get_input()
{
    int in;
    printf("\n Enter The Number : ");
    scanf("%d",&in);
    return in;
}
```

// Counting The Total Elements Available in Linklist

```
int count_linklist( struct node *head)
{
    int count = 0;
    struct node *temp;
    temp = head;
    while( temp != NULL )
    {
        count++;
        temp = temp->next;
    }
    return count;
}
```

```
// Function to reverse the linklist
struct node * reverse_linklist( struct node *head )
{
    struct node *recent,*last = NULL,*first = NULL;
    recent = head;
    while( recent != NULL )
    {
        first = recent->next;
        recent->next = last;
        last = recent;
        recent = first;
    }
    head = last;
    return head;
}
```

```
// Funtion Display will show all elements in linklist
void display_linklist(struct node *head)
{
    struct node *temp;
    if(head == NULL)
    {
        printf("\nThere Is Nothing To Display.\n");
    }
    else
    {
        temp = head;
        printf("\nThe List is : \n");
        while(temp->next != NULL)
```

```

    {
        printf(" %d => ",temp->data);
        temp = temp->next;
    }
    printf(" %d \n",temp->data);
}
}

```

```

/*****
*****

```

OUTPUT:

- 1 . Input Data To Link-List.
- 2 . Reverse Link-List.
- 3 . Display The Link List.
- 4 . Exit.

1

Enter The Number : 5

- 1 . Input Data To Link-List.
- 2 . Reverse Link-List.
- 3 . Display The Link List.
- 4 . Exit.

1

Enter The Number : 6

- 1 . Input Data To Link-List.

- 2 . Reverse Link-List.
- 3 . Display The Link List.
- 4 . Exit.

1

Enter The Number : 3

- 1 . Input Data To Link-List.
- 2 . Reverse Link-List.
- 3 . Display The Link List.
- 4 . Exit.

1

Enter The Number : 5

- 1 . Input Data To Link-List.
- 2 . Reverse Link-List.
- 3 . Display The Link List.
- 4 . Exit.

3

The List is :

5 => 3 => 6 => 5

- 1 . Input Data To Link-List.
- 2 . Reverse Link-List.
- 3 . Display The Link List.
- 4 . Exit.

1

Enter The Number : 9

- 1 . Input Data To Link-List.
- 2 . Reverse Link-List.
- 3 . Display The Link List.
- 4 . Exit.

1

Enter The Number : 2

- 1 . Input Data To Link-List.
- 2 . Reverse Link-List.
- 3 . Display The Link List.
- 4 . Exit.

3

The List is :

2 => 9 => 5 => 3 => 6 => 5

- 1 . Input Data To Link-List.
- 2 . Reverse Link-List.
- 3 . Display The Link List.
- 4 . Exit.

2

Link List Reversed.

- 1 . Input Data To Link-List.
- 2 . Reverse Link-List.
- 3 . Display The Link List.
- 4 . Exit.

3

The List is :

5 => 6 => 3 => 5 => 9 => 2

- 1 . Input Data To Link-List.
- 2 . Reverse Link-List.
- 3 . Display The Link List.
- 4 . Exit.

```
/*****  
*****
```

Name : Pradip . S . Karmakar

Roll-No : 10

Class : MCA-2

Subject : Advanced Programming

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

Questions : Write a modular C program to create a singly linked list Add All the Elements  
& Display.

```
*****  
*****/
```

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

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

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

```
// Structure Declaration
```

```
struct node{  
    int data;  
    struct node *next;  
};
```

```
// Functions Declaration
```

```
void menu( struct node *, struct node * );  
int ask_selection();  
struct node * get_linklist( struct node *, struct node * );  
int get_input();  
void Addition_Element_linklist( struct node * );  
void display_linklist(struct node *);
```

```
// Main Function
```

```
void main()  
{  
    struct node *new,*head = NULL;  
    menu(new,head);  
}
```

```
// Menu Function
```

```
void menu(struct node *new,struct node *head)  
{  
    int selection = ask_selection();  
    switch (selection)  
    {  
        case(1):  
            head = get_linklist(new,head);
```



```

        menu(new,head);
case(2):
    Addition_Element_linklist(head);
    menu(new,head);
case(3):
    display_linklist(head);
    menu(new,head);
default:
    exit(0);
}
}

```

// ask\_selection Function

```

int ask_selection()
{
    int n;

    printf("\n 1 . Input Data To Link-List. \n 2 . Display The Addition of Element from Link List. \n 3 .
Display The Link List. \n 4 . Exit. \n");

    scanf("%d",&n);

    if( n > 0 && n < 5 )
    {
        return n;
    }

    else{
        printf("\n Wrong Selection Please Choose Correct Options. \n");
        ask_selection();
    }
}
}

```

// getting elements to linklist with reverse functionality Function

struct node \* get\_linklist( struct node \*new, struct node \*head )

```
{
    int i,input = get_input();
    new = (struct node *)malloc(sizeof(struct node));
    new->data = input;
    new->next = head;
    head = new;
    return head;
}
```

// Function for getting input from user

int get\_input()

```
{
    int in;
    printf("\n Enter The Number : ");
    scanf("%d",&in);
    return in;
}
```

// Function to add element of the linklist

void Addition\_Element\_linklist( struct node \*head )

```
{
    int sum = 0;
    struct node *temp;
    temp = head;
    while( temp != NULL )
    {
        sum += temp->data;
```

```

        temp = temp->next;
    }
    printf(" \n The Total of the All Elements : %d \n",sum);
}

```

// Funtion Display will show all elements in linklist

```
void display_linklist(struct node *head)
```

```

{
    struct node *temp;
    if(head == NULL)
    {
        printf("\nThere Is Nothing To Display.\n");
    }
    else
    {
        temp = head;
        printf("\nThe List is : \n");
        while(temp->next != NULL)
        {
            printf(" %d => ",temp->data);
            temp = temp->next;
        }
        printf(" %d \n",temp->data);
    }
}

```

```

/*****
*****

```

OUTPUT:

- 1 . Input Data To Link-List.
- 2 . Display The Addition of Element from Link List.
- 3 . Display The Link List.
- 4 . Exit.

1

Enter The Number : 3

- 1 . Input Data To Link-List.
- 2 . Display The Addition of Element from Link List.
- 3 . Display The Link List.
- 4 . Exit.

1

Enter The Number : 5

- 1 . Input Data To Link-List.
- 2 . Display The Addition of Element from Link List.
- 3 . Display The Link List.
- 4 . Exit.

1

Enter The Number : 7

- 1 . Input Data To Link-List.
- 2 . Display The Addition of Element from Link List.
- 3 . Display The Link List.
- 4 . Exit.

1

Enter The Number : 8

- 1 . Input Data To Link-List.
  - 2 . Display The Addition of Element from Link List.
  - 3 . Display The Link List.
  - 4 . Exit.
- 3

The List is :

8 => 7 => 5 => 3

- 1 . Input Data To Link-List.
  - 2 . Display The Addition of Element from Link List.
  - 3 . Display The Link List.
  - 4 . Exit.
- 2

The Total of the All Elements : 23

- 1 . Input Data To Link-List.
- 2 . Display The Addition of Element from Link List.
- 3 . Display The Link List.
- 4 . Exit.

/\*\*\*\*\*  
\*\*\*\*\*

Name : Pradip . S . Karmakar

Roll-No : 10

Class : MCA-2

Subject : Advanced Programming

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

Questions : Write a modular C program to create two singly linked list & Append Into

First Linklist & Display

```
*****  
*****/
```

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

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

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

```
// Structure declaration
```

```
struct node{
```

```
    int data;
```

```
    struct node *next;
```

```
};
```

```
// function declarations
```

```
void menu( struct node *, struct node *, struct node *, struct node * );
```

```
int ask_selection();
```

```
struct node * get_linklist( struct node *new, struct node *head );
```

```
int get_input();
```

```
void display_linklist( struct node *, struct node * );
```

```
void append_link_list( struct node *, struct node * );
```

```
// main function
```

```
void main()
```

```
{
```

```
    struct node *list1,*list2,*head1 = NULL,*head2 = NULL;
```

```
    menu(list1,list2,head1,head2);
```

```
}
```

```
// menu function gives user to choose option from menu
```

```
void menu( struct node *list1, struct node *list2, struct node *head1, struct node *head2 )
```

```
{
```

```
    int selection = ask_selection();
```

```
    switch (selection)
```

```
    {
```

```
    case(1):
```

```
        head1 = get_linklist(list1,head1);
```

```
        menu(list1,list2,head1,head2);
```

```
    case(2):
```

```
        head2 = get_linklist(list2,head2);
```

```
        menu(list1,list2,head1,head2);
```

```
    case(3):
```

```
        display_linklist(head1,head2);
```

```
        menu(list1,list2,head1,head2);
```

```
    case(4):
```

```
        append_link_list(head1,head2);
```

```
        menu(list1,list2,head1,head2);
```

```
    default:
```

```
        exit(0);
```

```
    }
```

```
}
```

```
// ask_selection function will get the user option selected
```

```
int ask_selection()
```

```
{
```

```
    int n;
```

```
    printf("\n 1 . Input Data To First Link-List. \n 2 . Input Data To Second Link-List. \n 3 . Display The  
Elements from Both Link List. \n 4 . Append The Second Link List into First Link List. \n 5 . Exit. \n");
```

```
    scanf("%d",&n);
```

```
    if( n > 0 && n < 6 )
```

```
    {
```

```
        return n;
```

```
    }
```

```
    else{
```

```
        printf("\n Wrong Selection Please Choose Correct Options. \n");
```

```
        ask_selection();
```

```
    }
```

```
}
```

```
// get_link_list funtion use for inserting elements in linklist from user
```

```
struct node * get_linklist( struct node *new, struct node *head )
```

```
{
```

```
    int i,input = get_input();
```

```
    struct node *temp;
```

```
    new = (struct node *)malloc(sizeof(struct node));
```

```
    if( head == NULL )
```

```
    {
```

```
        head = new;
```

```
        new->data = input;
```

```
        new->next = NULL;
```



```

    }
    else{
        temp = head;
        while( temp->next != NULL )
        {
            temp = temp->next;
        }
        temp->next = new;
        new->data = input;
        new->next = NULL;
    }
    return head;
}

```

// get\_input will take input from user which is insert to link list.

```

int get_input()
{
    int in;
    printf("\n Enter The Number : ");
    scanf("%d",&in);
    return in;
}

```

// display both link list

```

void display_linklist(struct node *head1, struct node *head2 )
{
    struct node *temp;
    if(head1 == NULL)

```

```

{
    printf("\nFirst Link List : NULL\n");
}
else
{
    temp = head1;
    printf("\nFirst Link List : ");
    while(temp->next != NULL)
    {
        printf(" %d => ",temp->data);
        temp = temp->next;
    }
    printf(" %d \n",temp->data);
}

if(head2 == NULL)
{
    printf("\nSecond Link List : NULL\n");
}
else
{
    temp = head2;
    printf("\nSecond Link List : ");
    while(temp->next != NULL)
    {
        printf(" %d => ",temp->data);
        temp = temp->next;
    }
    printf(" %d \n",temp->data);
}
}

```

```
// append second linklist into first linklist
void append_link_list( struct node *head1, struct node *head2 )
{
    struct node *temp_head1,*temp_head2,*prev = NULL;
    if( head2 == NULL )
    {
        printf("\n Nothing To Append. \n");
    }
    else{
        temp_head1 = head1;
        temp_head2 = head2;

        while( temp_head1->next != NULL)
        {
            temp_head1 = temp_head1->next;
        }
        temp_head1->next = temp_head2;
    }
}

/*****
*****
```

OUTPUT:

- 1 . Input Data To First Link-List.
- 2 . Input Data To Second Link-List.
- 3 . Display The Elements from Both Link List.

4 . Append The Second Link List into First Link List.

5 . Exit.

1

Enter The Number : 5

1 . Input Data To First Link-List.

2 . Input Data To Second Link-List.

3 . Display The Elements from Both Link List.

4 . Append The Second Link List into First Link List.

5 . Exit.

1

Enter The Number : 6

1 . Input Data To First Link-List.

2 . Input Data To Second Link-List.

3 . Display The Elements from Both Link List.

4 . Append The Second Link List into First Link List.

5 . Exit.

3

First Link List : 5 => 6

Second Link List : NULL

1 . Input Data To First Link-List.

2 . Input Data To Second Link-List.

3 . Display The Elements from Both Link List.

4 . Append The Second Link List into First Link List.

5 . Exit.

4

Nothing To Append.

- 1 . Input Data To First Link-List.
- 2 . Input Data To Second Link-List.
- 3 . Display The Elements from Both Link List.
- 4 . Append The Second Link List into First Link List.
- 5 . Exit.

2

Enter The Number : 4

- 1 . Input Data To First Link-List.
- 2 . Input Data To Second Link-List.
- 3 . Display The Elements from Both Link List.
- 4 . Append The Second Link List into First Link List.
- 5 . Exit.

2

Enter The Number : 7

- 1 . Input Data To First Link-List.
- 2 . Input Data To Second Link-List.
- 3 . Display The Elements from Both Link List.
- 4 . Append The Second Link List into First Link List.
- 5 . Exit.

2

Enter The Number : 8

- 1 . Input Data To First Link-List.
  - 2 . Input Data To Second Link-List.
  - 3 . Display The Elements from Both Link List.
  - 4 . Append The Second Link List into First Link List.
  - 5 . Exit.
- 3

First Link List : 5 => 6

Second Link List : 4 => 7 => 8

- 1 . Input Data To First Link-List.
  - 2 . Input Data To Second Link-List.
  - 3 . Display The Elements from Both Link List.
  - 4 . Append The Second Link List into First Link List.
  - 5 . Exit.
- 4

- 1 . Input Data To First Link-List.
  - 2 . Input Data To Second Link-List.
  - 3 . Display The Elements from Both Link List.
  - 4 . Append The Second Link List into First Link List.
  - 5 . Exit.
- 3

First Link List : 5 => 6 => 4 => 7 => 8

Second Link List : 4 => 7 => 8

- 1 . Input Data To First Link-List.
- 2 . Input Data To Second Link-List.

- 3 . Display The Elements from Both Link List.
- 4 . Append The Second Link List into First Link List.
- 5 . Exit.

```
/*  
*****
```

Name : Pradip . S . Karmakar

Roll-No : 10

Class : MCA-2

Subject : Advanced Programming

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

Questions : Write a modular C program to swap two consecutive value from the  
linklist & display.( only value swap )

```
*****  
***** /
```

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

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

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

```
// Structure Declaration
```

```
struct node{
```

```
    int data;
```

```
    struct node *next;
```

```
};
```

```
// Functions Declaration
```

```
void menu( struct node *, struct node * );
```

```
int ask_selection();
```

```
struct node * get_linklist( struct node *, struct node * );
```

```
int get_input(int);
```

```
void Swap_Element_linklist( struct node * );
```

```
void display_linklist(struct node *);
```

```
// Main Function
```

```
void main()
```

```
{
```

```
    struct node *new,*head = NULL;
```

```
    menu(new,head);
```

```
}
```

```
// Menu Function
```

```
void menu(struct node *new,struct node *head)
```

```
{
```

```
    int selection = ask_selection();
```

```
    switch (selection)
```

```
    {
```

```
    case(1):
```

```
        head = get_linklist(new,head);
```

```
        menu(new,head);
```

```
    case(2):
```

```
        Swap_Element_linklist(head);
```

```
        menu(new,head);
```

```
    case(3):
```



```

        display_linklist(head);

        menu(new,head);
default:
        exit(0);
    }
}

```

// ask\_selection Function

```

int ask_selection()
{
    int n;

    printf("\n 1 . Input Data To Link-List. \n 2 . Swap The Consecutive Values. \n 3 . Display The Link
List. \n 4 . Exit. \n");

    scanf(" %d",&n);

    if( n > 0 && n < 5 )
    {
        return n;
    }
    else{
        printf("\n Wrong Selection Please Choose Correct Options. \n");
        ask_selection();
    }
}

```

// getting elements to linklist Function

```

struct node * get_linklist( struct node *new, struct node *head )
{
    int i,input = get_input(0);

    new = (struct node *)malloc(sizeof(struct node));

```

```
new->data = input;
new->next = head;
head = new;
return head;
}
```

// Function for getting input from user

```
int get_input(int a)
{
    int in;
    if( a == 0 )
    {
        printf("\n Enter The Number : ");
    }
    else{
        printf("\n Enter The Swapping Number : ");
    }
    scanf("%d",&in);
    return in;
}
```

// Function to swap element of the linklist

```
void Swap_Element_linklist( struct node *head )
{
    int swap_me = get_input(1),temp_hold = 0;
    struct node *temp;
    temp = head;
    while( temp != NULL && temp->data != swap_me )
    {
```

```

        temp = temp->next;
    }
    if( temp == NULL || temp->next == NULL )
    {
        printf("\n Swapping Not Possible. \n");
    }
    else{
        temp_hold = temp->data;
        temp->data = temp->next->data;
        temp->next->data = temp_hold;
        printf(" \n Swapping Done. \n");
    }
}

```

// Funtion Display will show all elements in linklist

```

void display_linklist(struct node *head)
{
    struct node *temp;
    if(head == NULL)
    {
        printf("\nThere Is Nothing To Display.\n");
    }
    else
    {
        temp = head;
        printf("\nThe List is : \n");
        while(temp->next != NULL)
        {
            printf(" %d => ",temp->data);
            temp = temp->next;
        }
    }
}

```

```

    }
    printf(" %d \n",temp->data);
}
}

```

```

/*****
*****

```

OUTPUT:

- 1 . Input Data To Link-List.
- 2 . Swap The Consecutive Values.
- 3 . Display The Link List.
- 4 . Exit.

1

Enter The Number : 5

- 1 . Input Data To Link-List.
- 2 . Swap The Consecutive Values.
- 3 . Display The Link List.
- 4 . Exit.

1

Enter The Number : 7

- 1 . Input Data To Link-List.
- 2 . Swap The Consecutive Values.
- 3 . Display The Link List.
- 4 . Exit.

1

Enter The Number : 4

- 1 . Input Data To Link-List.
- 2 . Swap The Consecutive Values.
- 3 . Display The Link List.
- 4 . Exit.

1

Enter The Number : 9

- 1 . Input Data To Link-List.
- 2 . Swap The Consecutive Values.
- 3 . Display The Link List.
- 4 . Exit.

3

The List is :

9 => 4 => 7 => 5

- 1 . Input Data To Link-List.
- 2 . Swap The Consecutive Values.
- 3 . Display The Link List.
- 4 . Exit.

2

Enter The Swapping Number : 7

Swapping Done.

- 1 . Input Data To Link-List.
- 2 . Swap The Consecutive Values.
- 3 . Display The Link List.
- 4 . Exit.

3

The List is :

9 => 4 => 5 => 7

- 1 . Input Data To Link-List.
- 2 . Swap The Consecutive Values.
- 3 . Display The Link List.
- 4 . Exit.

2

Enter The Swapping Number : 4

Swapping Done.

- 1 . Input Data To Link-List.
- 2 . Swap The Consecutive Values.
- 3 . Display The Link List.
- 4 . Exit.

3

The List is :

9 => 5 => 4 => 7

- 1 . Input Data To Link-List.
- 2 . Swap The Consecutive Values.
- 3 . Display The Link List.

4 . Exit.

```
/*****  
*****/
```

Name : Pradip . S . Karmakar

Roll-No : 10

Class : MCA-2

Subject : Advanced Programming

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

Questions : Write a modular C program to swap two consecutive value from the  
linklist & display. ( Only Address Swap )

```
*****  
*****/
```

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

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

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

```
// Structure Declaration
```

```
struct node{  
    int data;  
    struct node *next;  
};
```

```
// Functions Declaration
```

```
void menu( struct node *, struct node * );  
  
int ask_selection();  
  
struct node * get_linklist( struct node *, struct node * );  
  
int get_input(int);  
  
struct node * Swap_Element_linklist( struct node * );  
  
void display_linklist(struct node *);
```

```
// Main Function
```

```
void main()  
{  
    struct node *new,*head = NULL;  
    menu(new,head);  
}
```

```
// Menu Function
```

```
void menu(struct node *new,struct node *head)  
{  
    int selection = ask_selection();  
    switch (selection)  
    {  
        case(1):  
            head = get_linklist(new,head);  
            menu(new,head);  
        case(2):  
            head = Swap_Element_linklist(head);  
            menu(new,head);  
        case(3):  
            display_linklist(head);  
            menu(new,head);
```



```

    default:
        exit(0);
    }
}

// ask_selection Function
int ask_selection()
{
    int n;

    printf("\n 1 . Input Data To Link-List. \n 2 . Swap Address of The Consecutive Values. \n 3 . Display
The Link List. \n 4 . Exit. \n");

    scanf("%d",&n);
    if( n > 0 && n < 5 )
    {
        return n;
    }
    else{
        printf("\n Wrong Selection Please Choose Correct Options. \n");
        ask_selection();
    }
}

```

```

// getting elements to linklist Function
struct node * get_linklist( struct node *new, struct node *head )
{
    int i,input = get_input(0);
    new = (struct node *)malloc(sizeof(struct node));
    new->data = input;
    new->next = head;
}

```

```
    head = new;
    return head;
}
```

```
// Function for getting input from user
```

```
int get_input(int a)
{
    int in;
    if( a == 0 )
    {
        printf("\n Enter The Number : ");
    }
    else{
        printf("\n Enter The Swapping Number : ");
    }
    scanf("%d",&in);
    return in;
}
```

```
// Function to swap address of the linklist
```

```
struct node * Swap_Element_linklist( struct node *head )
{
    int swap_me = get_input(1);
    struct node *temp,*first = NULL,*prev = NULL;
    temp = head;
    while( temp != NULL && temp->data != swap_me )
    {
        prev = temp;
        temp = temp->next;
    }
}
```

```

    }

    if( temp == NULL || temp->next == NULL )
    {
        printf("\n Swapping Not Possible. \n");
    }
    else{
        if( prev == NULL )
        {
            head = temp->next;
            first = temp->next->next;
            temp->next->next = temp;
            temp->next = first;
        }
        else{
            first = temp->next->next;
            prev->next = temp->next;
            temp->next->next = temp;
            temp->next = first;
        }

        printf(" \n Swapping Done. \n");
    }

    return head;
}

```

// Funtion Display will show all elements in linklist

```

void display_linklist(struct node *head)
{
    struct node *temp;
    if(head == NULL)

```

```

{
    printf("\nThere Is Nothing To Display.\n");
}
else
{
    temp = head;
    printf("\nThe List is : \n");
    while(temp->next != NULL)
    {
        printf(" %d => ",temp->data);
        temp = temp->next;
    }
    printf(" %d \n",temp->data);
}
}

```

```

/*****
*****

```

OUTPUT:

- 1 . Input Data To Link-List.
- 2 . Swap Address of The Consecutive Values.
- 3 . Display The Link List.
- 4 . Exit.

1

Enter The Number : 2

- 1 . Input Data To Link-List.

2 . Swap Address of The Consecutive Values.

3 . Display The Link List.

4 . Exit.

1

Enter The Number : 4

1 . Input Data To Link-List.

2 . Swap Address of The Consecutive Values.

3 . Display The Link List.

4 . Exit.

1

Enter The Number : 6

1 . Input Data To Link-List.

2 . Swap Address of The Consecutive Values.

3 . Display The Link List.

4 . Exit.

1

Enter The Number : 7

1 . Input Data To Link-List.

2 . Swap Address of The Consecutive Values.

3 . Display The Link List.

4 . Exit.

1

Enter The Number : 9

- 1 . Input Data To Link-List.
- 2 . Swap Address of The Consecutive Values.
- 3 . Display The Link List.
- 4 . Exit.

1

Enter The Number : 16

- 1 . Input Data To Link-List.
- 2 . Swap Address of The Consecutive Values.
- 3 . Display The Link List.
- 4 . Exit.

3

The List is :

16 => 9 => 7 => 6 => 4 => 2

- 1 . Input Data To Link-List.
- 2 . Swap Address of The Consecutive Values.
- 3 . Display The Link List.
- 4 . Exit.

2

Enter The Swapping Number : 9

Swapping Done.

- 1 . Input Data To Link-List.
- 2 . Swap Address of The Consecutive Values.
- 3 . Display The Link List.
- 4 . Exit.

3

The List is :

16 => 7 => 9 => 6 => 4 => 2

- 1 . Input Data To Link-List.
- 2 . Swap Address of The Consecutive Values.
- 3 . Display The Link List.
- 4 . Exit.

2

Enter The Swapping Number : 16

Swapping Done.

- 1 . Input Data To Link-List.
- 2 . Swap Address of The Consecutive Values.
- 3 . Display The Link List.
- 4 . Exit.

3

The List is :

7 => 16 => 9 => 6 => 4 => 2

- 1 . Input Data To Link-List.
- 2 . Swap Address of The Consecutive Values.
- 3 . Display The Link List.
- 4 . Exit.

2

Enter The Swapping Number : 6

Swapping Done.

- 1 . Input Data To Link-List.
  - 2 . Swap Address of The Consecutive Values.
  - 3 . Display The Link List.
  - 4 . Exit.
- 3

The List is :

7 => 16 => 9 => 4 => 6 => 2

- 1 . Input Data To Link-List.
- 2 . Swap Address of The Consecutive Values.
- 3 . Display The Link List.
- 4 . Exit./



# Assignment 4

\*\*\*\*\*  
\*\*\*\*\*

Name : Pradip . S . Karmakar

Roll-No : 10

Class : MCA-2

Subject : Advanced Programming

\*\*\*\*\*  
\*\*\*\*\*

-----

Q1

-----

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

```
void main()
```

```
{
```

```
    int a = 1;
```

```
    char mystring[50];
```

```
    char *cptr;
```

```
    FILE *f1 = fopen("alternate.txt", "w");
```

```
    if(f1) {
```

```
        printf ("Enter a string : ");
```

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

```

cptr = mystring;
while(*cptr != '\0') {
    if(a) {
        fprintf(f1, "%c", *cptr);
        printf("%c", *cptr);
        a = 0;
    } else {
        a = 1;
    }
    cptr++;
}
fclose (f1);
}

else {
    printf("Unable to open file");
}

printf("\n");
}

```

Output :

Enter a string : Pradip

Pai

```

*****
*****

```

-----

Q2

-----

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

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

```
void main()
```

```
{
```

```
    int no_of_lines = 0;
```

```
    char line[1000];
```

```
    FILE *f1 = fopen("read.txt", "r");
```

```
    FILE *f2 = fopen("newread.txt", "w");
```

```
    if(f1 && f2) {
```

```
        if (fgetc(f1) == EOF) {
```

```
            printf ("No data found\n");
```

```
            exit(0);
```

```
        }
```

```
        while(fgets(line, sizeof line, f1)) {
```

```
            no_of_lines++;
```

```
            // fputs (line, stdout);
```

```
            fputs (line, f2);
```

```
        }
```

```
        printf("\n%d lines yanked and pasted", no_of_lines);
```

```
        fclose(f1);
```

```
        fclose(f2);
```

```
}
```

```

else {
    ferror(f1);
    ferror(f2);
}
puts ("\n");
}

```

Output :

5 lines yanked and pasted

```

*****
*****

```

-----

Q3

-----

```

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

```

```

void main()

```

```

{
    int i = 0, length = 0, count = 0, finding = 0, first_occurance = 0;
    char search_this_word[30], search;
    char c;
    FILE *fp = fopen ("read.txt", "r");

```

```
printf("Enter the word you want to search: ");  
scanf("%s", search_this_word);
```

```
length = strlen (search_this_word);
```

```
if ( fp ) {  
    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) {
        printf ("\\"%s\" found 1st time at %d position \n", search_this_word, first_occurance - length +
1);
        printf ("\\"%s\" found %d times\n", search_this_word, count);
    }
    else {
        printf ("\\"%s\" not found\n", search_this_word);
    }
}
else {
    puts ("Cannot open file to read");
}
printf ("\n");
}

```

\* Data in file :

Pradip karmakar

\* Output :

Enter the word you want to search: kar

"kar" found 1st time at 8 position

"kar" found 2 times

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

```
-----
```

Q4

```
-----
```

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

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

```
#define DIGIT 3
```

```
int * readandsort()
```

```
{
```

```
    int i = 0, j, index;
```

```
    static int myarray[10];
```

```
    int swap;
```

```
    char line[1000];
```

```
    FILE *f1 = fopen ("file1.txt", "r");
```

```
    FILE *f2 = fopen ("file2.txt", "r");
```

```
    if (f1 && f2) {
```

```
        while (fgets(line, sizeof line, f1)) {
```

```
            myarray[i++] = atoi(line);
```

```
        }
```

```
        while (fgets(line, sizeof line, f2)) {
```

```
            myarray[i++] = atoi(line);
```

```
        }
```

```

for (i = 0; i < DIGIT * 2 - 1; i++) {

    index = i;

    for (j = i + 1; j < DIGIT * 2; j++) {

        if (myarray[j] < myarray[index]) {
            index = j;
        }

    }

    swap = myarray[i];
    myarray[i] = myarray[index];
    myarray[index] = swap;
}

}

else {
    printf("Unable to open files");
    exit (1);
}

return myarray;
}

void writeSortedData(int *p)
{

```



```

int i;

FILE *f = fopen ("sorted-data.txt", "w");

for (i = 0; i < DIGIT * 2; i++) {
    fprintf(f, "%d ", *(p + i));
}

printf ("Write success to \"sorted-data.txt\"");
}

int main()
{
    int num = DIGIT, data, *p;
    FILE *f1 = fopen ("file1.txt", "w");
    FILE *f2 = fopen ("file2.txt", "w");

    if(f1 && f2) {
        printf("Enter %d numbers in FILE1\n", DIGIT);
        while(num--) {
            scanf ("%d",&data);
            fprintf (f1, "%d\n", data);
        }

        num = DIGIT;

        printf("Enter %d numbers in FILE2\n", DIGIT);
        while(num--) {
            scanf ("%d",&data);
            fprintf (f2, "%d\n", data);
        }
    }
}

```

```

    }

    fclose (f1);
    fclose (f2);

    p = readandsort();
    writeSortedData(p);
}

else {
    puts ("Unable to open files");
    exit(1);
}

printf("\n");

return 0;
}

```

Output :

Enter 3 numbers in FILE1

12 22 11

Enter 3 numbers in FILE2

1 212 121

Write success to "sorted-data.txt"

```

*****
*****

```

-----

Q5

-----

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

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

```
void main()
```

```
{
```

```
    FILE *fp1;
```

```
    int data,asscis[100]={0};
```

```
    fp1=fopen("filetxt.txt","r");
```

```
    if(fp1!=NULL)
```

```
    {
```

```
        while((data=getc(fp1))!=EOF)
```

```
        {
```

```
            asscis[data-97]=asscis[data-97]+1;
```

```
        }
```

```
        printf("\n Occarancies of Each character");
```

```
        for(int i=0;i<=65;i++)
```

```
        {
```

```
            if(asscis[i]>0)
```

```
            {
```

```
                printf(" %c- %d | ",(i+97),asscis[i]);
```

```
            }
```

```

        }

    }
else
{
    printf("\n Can't Open File for Reading.");
}

getch();
}

```

Data In File :

pradip karmakar

Output :

Occarancies of Each character : a- 4 | d- 1 | i- 1 | k- 2 | m- 1 | p- 2 | r- 3

```

*****
*****

```

-----

Q6

-----

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

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

```

void getdetail(char* file)
{
    FILE *input;
    char data;
    int ch=0,space=0,word=0,tab=0,newline=0;

    input=fopen(file,"r");

    if(input!=NULL)
    {
        printf("\n File Content:--");
        printf("\n-----\n");
        while((data=getc(input))!=EOF)
        {

            printf("%c",data);
            if(data==' ')
                space++;
            else if(data=='\n')
                newline++;
            else if(data=='\t')
                tab++;
            else
                ch++;
        }

        word=space + newline + tab + 1;
        printf("\n-----");
        printf("\n Total space is:%d",space);
    }
}

```

```

printf("\n Total newline is:%d",newline);

printf("\n Total tab is:%d",tab);

printf("\n Total character is:%d",ch);

printf("\n Total Words is:%d",word);


    }

    else

    {

        printf("\n Can't Open File for Reading.");

    }


    fclose(input);

}

void main()

{

    char file[20];

    printf("\n Enter File name:");

    scanf("%s",&file);

    getdetail(file);

    getch();

}

```

Output :

Enter File name:input.txt

File Content:--

-----

pradip Karm akar

roll no : 10

-----

Total space is:4

Total newline is:1

Total tab is:1

Total character is:23

Total Words is:7

\*\*\*\*\*  
\*\*\*\*\*

-----

Q7

-----

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

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

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

```
void copyFile()
```

```
{
```

```
    int delFile;
```

```
    char line[1000];
```

```
    FILE *f1 = fopen("tempFile.txt", "r");
```

```
    FILE *f2 = fopen("read.txt", "w");
```

```

if(f1 && f2) {

    while(fgets(line, sizeof line, f1)) {
        fputs (line, f2);
    }

    fclose(f1);
    fclose(f2);

    delFile = remove("tempFile.txt");

    if(delFile) {
        printf ("File not deleted");
    }
}

else {
    printf ("Error: ");
    ferror(f1);
    ferror(f2);
}
}

void main()

{
    int i = 0, j, length = 0, count = 0, finding = 0;
    char search_this_word[50], search;
    char c;
    FILE *f = fopen ("read.txt", "r+");
    FILE *f1 = fopen ("tempFile.txt", "w");

```



```
printf("Enter the word you want to search: ");
```

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

```
length = strlen (search_this_word);
```

```
if (f) {
```

```
    if(!length) {
```

```
        exit(0);
```

```
    }
```

```
    search = search_this_word[0];
```

```
    while((c = fgetc(f)) != EOF){
```

```
        fprintf (f1, "%c", c);
```

```
        if (search == c) {
```

```
            finding = 1;
```

```
            if(length == i + 1) {
```

```
                i = finding = 0;
```

```
                count++;
```

```
                fseek (f1, -length, SEEK_CUR);
```

```
                for (j = 0; j < length; j++) {
```

```
                    //fprintf (f1, "*");
```

```
                }
```

```
                search = search_this_word[i];
```

```
            }
```

```
        else {
```

```
            search = search_this_word[++i];
```

```
        }
```

```

    }

    else {

        finding = 0;

    }

}

printf ("\\"%s\" found %d times", search_this_word, count);

if (count) {

    printf (" and deleted every time from the file");

}

fclose(f);

fclose(f1);

copyFile();

}

else {

    puts ("Cannot open file to read");

}

printf ("\n");

}

```

Output :

Enter the word you want to search: kar

"kar" found 3 times and deleted every time from the file

```

*****
*****

```

-----  
Q8  
-----

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

```
void copyFile()
{
    int delFile;
    char line[1000];
    FILE *f1 = fopen("tempFile.txt", "r");
    FILE *f2 = fopen("read.txt", "w");
    if(f1 && f2) {

        while(fgets(line, sizeof line, f1)) {
            fputs (line, f2);
        }

        fclose(f1);
        fclose(f2);

        delFile = remove("tempFile.txt");

        if(delFile) {
            printf ("File not deleted");
        }
    }

    else {
```

```

        ferror(f1);

        ferror(f2);
    }
}

void main()
{
    char prev = '\n', c;
    int count_lines = 0;
    FILE *f = fopen ("read.txt", "r+");
    FILE *f1 = fopen ("tempFile.txt", "w");

    if (f) {
        while((c = fgetc(f)) != EOF){
            if(c == '\n' && prev == '\n') {
                count_lines = 1;
            }
            else {
                fprintf (f1, "%c", c);
            }
            prev = c;
        }
        fclose(f);
        fclose(f1);
        copyFile();

        printf ("Unwanted lines deleted");

    }
    else {
        puts ("Cannot open file to read");
    }
}

```

```

    }

    printf ("\n");
}

```

Output :

Unwanted lines deleted

```

*****
*****

```

-----

Q9

-----

```

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

```

```

void copyFile()
{
    int delFile;
    char line[1000];
    FILE *f1 = fopen("tempFile.txt", "r");
    FILE *f2 = fopen("readlinebyline.txt", "w");
    if(f1 && f2) {

        while(fgets(line, sizeof line, f1)) {
            fputs (line, f2);
        }
    }
}

```

```

fclose(f1);

fclose(f2);


delFile = remove("tempFile.txt");


if(delFile) {
    printf ("File not deleted");
}
}

else {
    ferror(f1);
    ferror(f2);
}
}

void main()
{
    char prev = 'A', c;
    int stop = 0, first_occurrence = 0, second_occurrence = 0, long_comments = 0;
    FILE *f = fopen ("readlinebyline.txt", "r+");
    FILE *f1 = fopen ("tempFile.txt", "w");

    if (f) {
        while((c = fgetc(f)) != EOF){
            if(long_comments == 1) {
                if (c == '/' && prev == '*') {
                    long_comments = 0;
                }
            }
        }
    }
}

```

```

if (first_occurrence == 1) {
    if (c == '*') {
        long_comments = 1;
    }
    else if(c != '/') {
        stop = 0;
    }else {
        second_occurrence = 1;
    }
    first_occurrence = 0;
}

if(stop == 1) {
    if (c == '\n') {
        stop = 0;
        second_occurrence = 0;
    }
}

else if(c == '/') {
    first_occurrence = 1;
    stop = 1;
}

if(stop == 0 && second_occurrence == 0 && long_comments == 0) {
    if (prev == '/' && c != '\n'){
        fprintf (f1, "/");
    }
    fprintf (f1, "%c", c);
}
prev = c;
}

```

```

fclose(f);

fclose(f1);

copyFile();


printf("Comments Removed");


}

else {

    puts ("Cannot open file to read");

}

printf ("\n");
}

```

Data in file

```

/*hello this is comment*/
hello

```

Output :

Comments Removed

```

*****
*****

```

-----

Q10

-----

```

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

```



```
struct customers
```

```
{  
    char name[60],telephone[10];  
};
```

```
void getdata(char *fname, struct customers cu[],int n)
```

```
{  
    int i;  
    FILE *fp;  
    fp=fopen(fname,"w");  
    if(fp==NULL)  
    {  
        printf("\n Error in opening an file...");  
        exit(0);  
    }  
    for(i=0;i<n;i++)  
    {  
        fflush(stdin);  
        printf("\n Enter Name of Customer:");  
        gets(cu[i].name);  
        fflush(stdin);  
        printf(" Enter Telephone No:");  
        scanf("%[^\\n]",&cu[i].telephone);  
  
        fprintf(fp,"%s \\t %s \\n",cu[i].name,cu[i].telephone);  
    }  
    fclose(fp);  
}
```

```
void displayR(char *fname,struct customers cu[],int n)
```

```

{
    int i;
    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++)
    {
        fscanf(fp,"%s",&cu[i].name);
        fscanf(fp,"%s",&cu[i].telephone);

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

    fclose(fp);
}

```

```

void main()
{
    int n;
    char file_name[80];

    struct customers cu[10];

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

```

```
printf("\n Enter number of records: ");  
scanf("%d",&n);  
  
getdata(file_name,cu,n);  
  
printf("\n ===== Display Records =====");  
  
displayR(file_name,cu,n);  
  
}
```

Output:

Enter File name: records.txt

Enter number of records: 2

Enter Name of Customer:Pradip

Enter Telephone No:1234567890

Enter Name of Customer:Sudip

Enter Telephone No:0987654321

===== Display Records =====

Pradip      1234567890

Sudip      0987654321

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

-----

Q11

-----

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

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

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

```
struct Employee
```

```
{
```

```
    int empno;
```

```
    char name[20],address[50],phone[20];
```

```
    double salary;
```

```
};
```

```
void getdata(char *fname,struct Employee e1[],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++)
```

```
    {
```

```
        fflush(stdin);
```

```

        printf("\n Enter Employee ID: ");
        scanf("%d", &e1[i].empno);
        printf(" Enter Employee name: ");
        scanf("%s",&e1[i].name);
        fflush(stdin);
        printf(" Enter Employee address: ");
        scanf("%s",&e1[i].address);
        printf(" Enter phone number:");
        fflush(stdin);
        scanf("%s",&e1[i].phone);
        printf(" Enter Salary:");
        fflush(stdin);
        scanf("%lf",&e1[i].salary);

        fwrite(&e1[i], sizeof(e1[i]), 1, fp);

//      fprintf(fp,"\n %d \t %s \t %f",e1[i].empno,e1[i].name,e1[i].salary);
    }

    fclose(fp);
}

```

```

void display_namewise(char *fname,struct Employee e1[],int n)

```

```

{
    int i,cnt=0;
    char name[20];

    FILE *fp;

    fp=fopen(fname,"r");
    fseek(fp,0L,0);
    if(fp==NULL)

```

```

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

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

for(int i=0;i<n;i++)
{

    fread(&e1[i],sizeof(e1[i]),1,fp);
    if(strcmp(name,e1[i].name)==0)
    {
        cnt++;
        printf("\n %d \t %s \t %s \t %s \t
%lf",e1[i].empno,e1[i].name,e1[i].address,e1[i].phone,e1[i].salary);
    }

}

if(cnt==0)
    printf("\n Recored Doesent exist");

fclose(fp);
}

void display(char *fname,struct Employee e1[],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(&e1[i],sizeof(e1[i]),1,fp);

    printf("\n %d \t %s \t %s \t %s \t
%lf",e1[i].empno,e1[i].name,e1[i].address,e1[i].phone,e1[i].salary);

}

fclose(fp);
}

```

```

void modify(char *fname,struct Employee 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 Employee Id:");
scanf("%d",&empid);

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

    if(e1[i].empno==empid)
    {
        cnt++;
        fflush(stdin);
        printf("\nEnter Employee ID: ");
        scanf("%d", &e1[i].empno);
        printf(" Enter Employee name: ");
        scanf("%s",&e1[i].name);
        fflush(stdin);
        printf(" Enter Employee address: ");
        scanf("%s",&e1[i].address);
        printf(" Enter phone number:");
        fflush(stdin);
        scanf("%s",&e1[i].phone);
        printf(" Enter Salary:");
        fflush(stdin);
        scanf("%lf",&e1[i].salary);
    }
}

```



```

        fwrite(&e1[i], sizeof(e1[i]), 1, fptr);
    }
    else
    {
        fwrite(&e1[i], sizeof(e1[i]), 1, fptr);
    }
}

if(cnt>0)
    printf("\n Employee Edited Successfully.");
else
    printf("\n Employee 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);

}

int removeR(char *fname,struct Employee 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 Employee Id:");
scanf("%d",&empid);

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

    if(e1[i].empno==empid)
    {
        cnt++;
    }
    else
    {
        fwrite(&e1[i], sizeof(e1[i]), 1, fp);
    }
}

if(cnt>0)
    printf("\n Employee Deleted Successfully.");
else
    printf("\n Employee Not Exist...!!");

fclose(fp);
fclose(fp);

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];
    int cnt=0;
    struct Employee e1[10];

```

```

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

do
{
    printf("\n\n 1.Add a new record.");
    printf("\n 2.Delete a record. ");
    printf("\n 3.Modify an existing record.");
    printf("\n 4.Retrieve and display an entire record for a given name.");
    printf("\n 5.Generate a complete list of all names, addresses and telephone
numbers.\n");

    printf("\n Enter Your choice:");
    scanf("%d",&ch);

    switch(ch)
    {
        case 1:printf("\n Enter number of records to Add:");
                scanf("%d",&n);
                cnt=cnt+n;
                getdata(file_name,e1,n);
                break;

        case 2:
                res=removeR(file_name,e1,cnt);
                // int remove(char *fname,struct Employee e1[],int n)

                if(res)
                    cnt=cnt-1;
                break;

        case 3:
                modify(file_name,e1,cnt);

```

```

                                break;

                        case 4:

                                display_namewise(file_name,e1,cnt);

                                break;

                        case 5:

                                printf("\n Display Records");

                                display(file_name,e1,cnt);

                                break;

                }

                printf("\n Do You want to continue:");

                scanf("%s",&choice);


        }while(choice=='y');

}

```

```

*****
*****

```

-----

Q12

-----

```

#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,struct 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(" Enter Country Name:");
        scanf("%s",&c1[i].name);
    }
}
```

```

        fflush(stdin);

        printf(" 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,struct 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 -----");

    printf("\n 1.Add a new Record.");

    printf("\n 2.Display a Record. ");

    printf("\n -----");

    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:country.txt

-----

1.Add a new Record.

2.Display a Record.

-----

Enter Your choice:1

Enter number of records for Add:2

Enter Country id:1

Enter Country Name:India

Enter Capital Name:Delhi

Enter Country id:2

Enter Country Name:America

Enter Capital Name:WasingtonDC

Do You want to continue:y

Enter Your choice:2

-----

Display Records

-----

- 1 India Delhi
- 2 America WasingtonDC

Do You want to continue:n

\*\*\*\*\*  
\*\*\*\*\*

-----

Q13

-----

```
#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:country.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

-----

111    India    Delhi

222    America WasingtonDC

Do You want to continue:y

Enter Your choice:2

Enter Country:America

222    America    WasingtonDC



Do You want to continue:n

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

-----

Q14

-----

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

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

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

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

```
void Insert(char *fname,int n)
```

```
{
```

```
    char str[100];
```

```
    int i;
```

```
    FILE *fp;
```

```
    fp=fopen(fname,"w");
```

```
    if(fp==NULL)
```

```
    {
```

```
        printf("\n Error in opening an file...");
```

```
        exit(0);
```

```
    }
```

```
    for(i=0;i<=n;i++)
```

```
    {
```

```

        fgets(str,sizeof(str),stdin);
        fputs(str,fp);
    }
    fclose(fp);
}

void get_line(char *fname,int n,int lno)
{
    char str[100];
    int i,cnt=0;
    FILE *fp;
    fp=fopen(fname,"r");
    if(fp==NULL)
    {
        printf("\n Error in opening an file...");
        exit(0);
    }

    for(i=0;i<=n;i++)
    {
        fgets(str,sizeof(str),fp);
        if(i==lno)
        {
            fputs(str,stdout);
            cnt++;
        }
    }
    fclose(fp);
    //printf("\n cnt is:%d",cnt);
}

```

```

void Insert_line_at_k(char *fname,int n,int k)
{
    char str[100],str2[100],ch;
    int i,cnt=0,n2;
    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);
    }

    for(i=0;i<=n;i++)
    {
        fgets(str,sizeof(str),fp);
        fputs(str,fptr);
        if(i==k)
        {
            printf("\n Enter Number of line:");
            scanf("%d",&n2);

            for(int j=0;j<=n2;j++)
            {

```

```

                fgets(str2,sizeof(str2),stdin);
                if(j!=0)
                {
                        fputs(str2,fptr);
                }
        }

        cnt++;
}

```

```

}
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);
}

```

```

while((ch=getc(fptr))!=EOF)
{

```

```
        fprintf(fp,"%c",ch);
    }
    fclose(fp);
    fclose(fptr);
}
```

```
void Delete_line_at_k(char *fname,int n,int k)
```

```
{
    char str[100],str2[100],ch;
    int i,cnt=0,n2;
    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(fptr==NULL)
    {
        printf("\n Error in opening an file...");
        exit(0);
    }

    for(i=0;i<=n;i++)
    {
        fgets(str,sizeof(str),fp);
        fputs(str,stdout);
    }
}
```

```
fputs(str,fptr);  
if(i==k)  
{  
    printf("\n Enter Number of line:");  
    scanf("%d",&n2);  
  
    for(int j=0;j<n2;j++)  
    {  
        fgets(str,sizeof(str),fp);  
        i++;  
    }  
}  
  
}  
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);  
}
```

```

    }

    while((ch=getc(fptr))!=EOF)
    {
        fprintf(fp,"%c",ch);
    }
    fclose(fp);
    fclose(fptr);
}

void display(char *fname,int n)
{
    int i;
    FILE *fp;
    char ch;

    fp=fopen(fname,"r");
    fseek(fp,0L,0);
    if(fp==NULL)
    {
        printf  ("\n Error in opening an file...");
        exit(0);
    }
    ch = fgetc(fp);
    while(ch != EOF)
    {
        printf("%c", ch);
        ch = fgetc(fp);
    }
}

```

```

        fclose(fp);
    }

void main()
{
    int n,res,lno;
    char file_name[80],choice='n',name[20],tele[10],ch[10],ch2;
    static int cnt=0;

    printf("\n Enter File name:");
    scanf("%s",&file_name);
    printf("\n -----");
    printf("\n 1.$E-Enter new text ");
    printf("\n 2.$L-list the entire block of text");
    printf("\n 3.$Fk-find(retrieve)line number k ");
    printf("\n 4.$In-insert n lines after line number k ");
    printf("\n 5.$Dn-delete n lines after line number k ");
    printf("\n 6.$S-save the edited block of text and end the computation");
    printf("\n -----");
    do
    {
        printf("\n Enter Your choice:");
        scanf("%s",&ch);
        switch(ch[0])
        {
            case 69:printf("\n Input the number of lines to be written : ");
                    scanf("%d", &n);
                    cnt=cnt+n;
                    Insert(file_name,cnt);
                    break;

            case 76:

```



```
printf("\n-----");  
printf("\n Display Files");  
printf("\n-----");  
display(file_name,cnt);  
break;
```

case 70:

```
ch2=ch[1]-'0';  
lno=ch2;  
if(lno>cnt)  
    printf("\n Invalid line number");  
else  
    get_line(file_name,cnt,lno);  
break;
```

case 73:

```
ch2=ch[1]-'0';  
lno=ch2;  
if(lno>cnt)  
    printf("\n Invalid line number");  
else  
    Insert_line_at_k(file_name,cnt,lno);  
break;
```

case 68:

```
ch2=ch[1]-'0';  
lno=ch2;  
if(lno>cnt)  
    printf("\n Invalid line number");  
else  
    Delete_line_at_k(file_name,cnt,lno);  
  
break;
```

```

        case 83:

            exit(0);

            break;

        default :

            printf("\n Invalid Choice");

            break;

    }

    printf("\n Do You want to continue:");

    scanf("%s",&choice);

}while(choice=='y' || choice=='Y');

    getch();
}

```

```

*****
*****

```

-----

Q15

-----

```

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<string.h>

```

```

void Insert(char *fname,int n)
{
    char str[100];
    int i;
    FILE *fp;
    fp=fopen(fname,"w");
    if(fp==NULL)
    {
        printf("\n Error in opening an file...");
        exit(0);
    }

    for(i=0;i<=n;i++)
    {
        fgets(str,sizeof(str),stdin);
        fputs(str,fp);
    }
    fclose(fp);
}

```

```

void get_line(char *fname,int n,int lno)
{
    char str[100];
    int i,cnt=0;
    FILE *fp;
    fp=fopen(fname,"r");
    if(fp==NULL)
    {
        printf("\n Error in opening an file...");
        exit(0);
    }
}

```

```

    }

    for(i=0;i<=n;i++)
    {
        fgets(str,sizeof(str),fp);
        if(i==lno)
        {
            fputs(str,stdout);
            cnt++;
        }
    }
    fclose(fp);
    //printf("\n cnt is:%d",cnt);
}

```

```

void Insert_line_at_k(char *fname,int n,int k)
{
    char str[100],str2[100],ch;
    int i,cnt=0,n2;
    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);
    }

    for(i=0;i<=n;i++)
    {
        fgets(str,sizeof(str),fp);
        fputs(str,fptr);
        if(i==k)
        {
            printf("\n Enter Number of line:");
            scanf("%d",&n2);

            for(int j=0;j<=n2;j++)
            {
                fgets(str2,sizeof(str2),stdin);
                if(j!=0)
                {
                    fputs(str2,fptr);
                }
            }

            cnt++;
        }
    }

    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);
}

while((ch=getc(fptr))!=EOF)
{
    fprintf(fp,"%c",ch);
}

fclose(fp);
fclose(fptr);
}

```

```

void Delete_line_at_k(char *fname,int n,int k)
{
    char str[100],str2[100],ch;
    int i,cnt=0,n2;
    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);
    }

    for(i=0;i<=n;i++)
    {
        fgets(str,sizeof(str),fp);
        fputs(str,stdout);
        fputs(str,fptr);
        if(i==k)
        {
            printf("\n Enter Number of line:");
            scanf("%d",&n2);

            for(int j=0;j<n2;j++)
            {
                fgets(str,sizeof(str),fp);
                i++;
            }

        }

    }

    }

    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);
```

```
}
```

```
while((ch=getc(fptr))!=EOF)
```

```
{
```

```
    fprintf(fp,"%c",ch);
```

```
}
```

```
fclose(fp);
```

```
fclose(fptr);
```

```
}
```

```
void display(char *fname,int n)
```

```
{
```

```
    int i;
```

```
    FILE *fp;
```



```

char ch;

fp=fopen(fname,"r");
fseek(fp,0L,0);
if(fp==NULL)
{
    printf ("\n Error in opening an file...");
    exit(0);
}
ch = fgetc(fp);
while(ch != EOF)
{
    printf("%c", ch);
    ch = fgetc(fp);
}
fclose(fp);
}

void main()
{
    int n,res,lno;
    char file_name[80],choice='n',name[20],tele[10],ch[10],ch2;
    static int cnt=0;

    printf("\n Enter File name:");
    scanf("%s",&file_name);
    printf("\n -----");
    printf("\n 1.$E-Enter new text ");
    printf("\n 2.$L-list the entire block of text");
    printf("\n 3.$Fk-find(retrieve)line number k ");
    printf("\n 4.$In-insert n lines after line number k ");

```

```

printf("\n 5.$Dn-delete n lines after line number k ");
printf("\n 6.$S-save the edited block of text and end the computation");
printf("\n -----");
do
{
    printf("\n Enter Your choice:");
    scanf("%s",&ch);
    switch(ch[0])
    {
        case 69:printf("\n Input the number of lines to be written : ");
                scanf("%d", &n);
                cnt=cnt+n;
                Insert(file_name,cnt);
                break;

        case 76:
                printf("\n-----");
                printf("\n Display Files");
                printf("\n-----");
                display(file_name,cnt);
                break;

        case 70:
                ch2=ch[1]-'0';
                lno=ch2;
                if(lno>cnt)
                    printf("\n Invalid line number");
                else
                    get_line(file_name,cnt,lno);
                break;

        case 73:
                ch2=ch[1]-'0';
                lno=ch2;

```

```

        if(lno>cnt)
            printf("\n Invalid line number");
        else
            Insert_line_at_k(file_name,cnt,lno);
        break;

    case 68:

        ch2=ch[1]-'0';
        lno=ch2;
        if(lno>cnt)
            printf("\n Invalid line number");
        else
            Delete_line_at_k(file_name,cnt,lno);

        break;

    case 83:

        exit(0);
        break;

    default :

        printf("\n Invalid Choice");
        break;

    }

    printf("\n Do You want to continue:");
    scanf("%s",&choice);

}while(choice=='y' || choice=='Y');

getch();
}

```

OUTPUT :

Enter File name:file.txt

-----

- 1.\$E-Enter new text
- 2.\$L-list the entire block of text
- 3.\$Fk-find(retrieve)line number k
- 4.\$In-insert n lines after line number k
- 5.\$Dn-delete n lines after line number k
- 6.\$S-save the edited block of text and end the computation

-----

Enter Your choice:E

Input the number of lines to be written : 3

Pradip karmakar

hello world

practise wise

effieciint usage

Do You want to continue:y

Enter Your choice:L

-----

Display Files

-----

Pradip karmakar

hello world

practise wise

effieciint usage

Do You want to continue:Y

Enter Your choice:F1

Pradip karmakar

Do You want to continue:Y

Enter Your choice:I2

Enter Number of line:1

C++ is Object oriented

Do You want to continue:y

Enter Your choice:L

-----

Display Files

-----

Pradip karmakar

hello world

C++ is Object oriented

practise wise

effieciint usage

Do You want to continue:Y

Enter Your choice:D4

Enter Number of line:1

Do You want to continue:Y

Enter Your choice:L

-----

Display Files

-----

Pradip karmakar

hello world

C++ is Object oriented

effiecint usage

Do You want to continue:N

\*\*\*\*\*  
\*\*\*\*\*

-----

Q16

-----

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 :

PS E:\MCA\MCA SEM 2\AP\Assignment 4>XT16 78 96 84 45

Average: 75.75

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

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 :

PS E:\MCA\MCA SEM 2\AP\Assignment 4>XT16 3 5 8 2 4

Factorials are as follows:

03: 6

05: 120

08: 40320

02: 2

04: 24

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

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");  
  
    // printf("\nWorking on File: %s \n", d_name);  
  
    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) {

        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) {
    // printf ("\"%s\" found 1st time at %d position \n", search_this_word, first_occurance -
length + 1);

    // printf ("\"%s\" found %d times\n", search_this_word, count);

    return first_occurance - length + 1;
}

```

```

    else {
        // printf ("\\"%s\\" not found\\n", search_this_word);
        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)
{
    // Assuming max 50 files in directory
    int arr[50], count = 0;

    int *ptr = findWordInDirectory(argv, arr);
    // Return Position where Word was found, else returns -1

    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 :

PS E:\MCA\MCA SEM 2\AP\Assignment 4>XT16 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

PS E:\MCA\MCA SEM 2\AP\Assignment 4>XT16 Temp.txt Real.txt

File renamed successfully.

PS E:\MCA\MCA SEM 2\AP\Assignment 4>XT16

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:

PS E:\MCA\MCA SEM 2\AP\Assignment 4>XT16

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:

PS E:\MCA\MCA SEM 2\AP\Assignment 4>XT16 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

Exception Handeling

PS E:\MCA\MCA SEM 2\AP\Assignment 4>XT16 b.txt e2.txt c.txt d.txt

Working on "b.txt" file:

5 lines yanked and pasted

Working on "e2.txt" file:

No such File exists

Working on "c.txt" file:

3 lines yanked and pasted

All files are NOT copied

\*\*\*\*\*  
\*\*\*\*\*