ID: 20DCE019

Name: Yatharth Chauhan

Class: DCE-1

Subject: CCP

SET-1 (PRACTICAL-1)

Aim: Write C program that will output this passage by Michael Singer.

Make sure your output looks exactly as shown here. Use Required Escape Sequence and ASCII Value. There are three shapes in the output: Smiling Face, Diamond & Heart.

The ASCII Value for Smiling face is 1.

The ASCII Value for Diamond is 4.

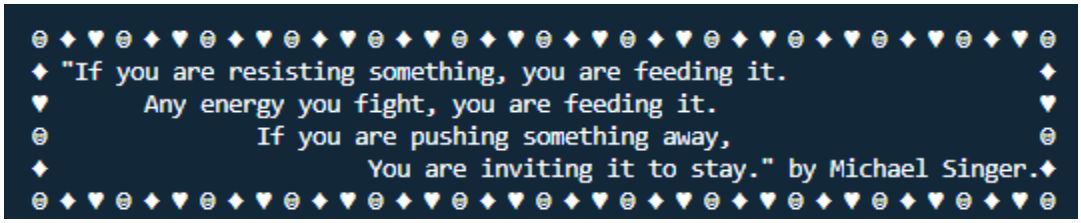
The ASCII Value for Heart is 3.

Source Code:

```
#include <stdio.h>


void main()
{
    char a = 1, b = 4, c = 3;


    printf("\n %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c",
        a, b, c, a, b, c, a, b, c, a, b, c, a, b, c, a, b, c, a, b, c, a, b, c, a, b, c, a, b, c, a, b, c, a, b, c, a);
    printf("\n %c \"If you are resisting something, you are feeding it.          %c", b, b);
    printf("\n %c     Any energy you fight, you are feeding it.            %c", c, c);
    printf("\n %c          If you are pushing something away,           %c", a, a);
    printf("\n %c               You are inviting it to stay.\" by Michael Singer.%c", b, b);
    printf("\n %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c %c",
```

```
    a, b, c, a, b, c, a, b, c, a, b, c, a, b, c, a, b, c, a, b, c, a, b, c, a, b, c, a, b, c, a, a, a);
}
```

Output:



SET-1 (PRACTICAL-2)

Aim: In a town, the percentage of men is 52. The percentage of total literacy is 48. If total percentage of literate men is 35 of the total population, write a program to find the total number of illiterate men and women if the population of the town is 80,000. Write Algorithms and Flowchart of this program.

Source Code:

```
#include <stdio.h>

int main()

{

  int p = 80000, m, w, tl, lm, lw;


  printf("\n Total Population is = %d", p);


  m = (p * 52) / 100;

  w = p - m;


  printf("\n Total Men in Population are = %d", m);

  printf("\n Total Women in Population are = %d", w);
```

```
 tl = (p * 48) / 100;
 lm = (p * 35) / 100;
 lw = tl - lm;

 printf("\n Total Literate Men = %d", lm);
 printf("\n Total Literate Women = %d", lw);

 printf("\n Total Iliterate Men = %d", m - lm);
 printf("\n Total Iliterate Women = %d", w - lw);

 return 0;
}
```

Output:

```
 Total Population is = 80000
 Total Men in Population are = 41600
 Total Women in Population are = 38400
 Total Literate Men = 28000
 Total Literate Women = 10400
 Total Iliterate Men = 13600
 Total Iliterate Women = 28000
PS F:\C Language Program\Program 2020-2021>
```

SET-1 (PRACTICAL-3)

Aim: A cashier has currency notes of denominations 10,50 and 100. If the amount to be withdrawn is input through the keyboard in hundreds, find the total number of currency notes of each denomination the cashier will have to give to the withdrawer.

Source Code:

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

```
{
    int N_10, N_50, N_100;
    int amount;

    printf("\nEnter the amount to withdraw: ");
    scanf("%d", &amount);

    N_100 = amount / 100;
    amount = amount % 100;

    N_50 = amount / 50;
    amount = amount % 50;

    N_10 = amount / 10;
    amount = amount % 10;

    printf("\n100 * % d = % d", N_100, 100 * N_100);
    printf("\n50 * % d = % d", N_50, 50 * N_50);
    printf("\n10 * % d = % d", N_10, 10 * N_10);
    printf("\nreaming amount: % d", amount);

    return 0;
}
```

Output:

```
Enter the amount to withdraw: 2362

100 *  23 =  2300
50 *  1 =  50
10 *  1 =  10
reaming amount:  2
```

SET-2 (PRACTICAL-1)

Aim: Write a program to calculate Net Salary. User has to input Basic Salary and Output should be: Enter Basic Salary: 5000 (e.g. 5000) Allowances:

DA = 70% of Basic Salary HRA = 7% of Basic Salary

MA = 2% of Basic Salary

TA = 4% of Basic Salary Deduction:

PF = 12% of Basic Salary

IT = any value (e.g. 500)

Net Salary = Basic Salary + Allowances – Deduction

Source Code:

```
#include <stdio.h>

int main()

{

 int bs, da, hra, ma, ta, pf, it = 500, net_salary;


 printf("\n Enter The Basic Salary: ");

 scanf("%d", &bs);


 printf("\n Allowances: \n");


 da = (bs * 70) / 100;

 printf("\n DA = %d", da);


 hra = (bs * 7) / 100;

 printf("\n HRA = %d", hra);


 ma = (bs * 2) / 100;
```

```
  printf("\n MA = %d", ma);

  ta = (bs * 4) / 100;
  printf("\n TA = %d", ta);

  printf("\n");
  printf("\n Deducation: \n");

  pf = (bs * 12) / 100;
  printf("\n PF = %d", pf);
  printf("\n IT = %d", it);

  net_salary = bs + (da + hra + ma + ta) - (pf + it);
  printf("\n Net Salary = %d", net_salary);

  return 0;
}
```

Output:

```
Enter The Basic Salary: 90000

Allowances:

DA = 63000
HRA = 6300
MA = 1800
TA = 3600

Deducation:

PF = 10800
IT = 500
Net Salary = 153400
```

SET-2 (PRACTICAL-2)

Aim: The distance between two cities (in km) is input through the keyboard. Write a program to convert and print its distance in meters, feet, inches and centimeters.

Source Code:

```
#include <stdio.h>
int main()
{
   float km, m, cm, ft, in;

   printf("Enter distance in kilometers: ");
   scanf("%f", &km);

   m = km * 1000;
   cm = km * 1000 * 100;
   ft = km * 3280.84;
   in = km * 39370.08;

   printf("\nThe distance in Feet: %f\n", ft);
   printf("The distance in Inches: %f\n", in);
   printf("The distance in Meters: %f\n", m);
   printf("The distance in Centimeters: %f\n", cm);

   return (0);
}
```

Output:

```
Enter distance in kilometers: 10

The distance in Feet: 32808.398438
The distance in Inches: 393700.812500
The distance in Meters: 10000.000000
The distance in Centimeters: 1000000.000000
```

SET-3 (PRACTICAL-1)

Aim: Write a program to find the greatest of the three numbers entered through the keyboard using conditional operators.

Source Code:

```
#include <stdio.h>

int main()

{

   int a, b, c, max;


   printf("Enter 1st Number: ");

   scanf("%d", &a);

   printf("Enter 2nd Number: ");

   scanf("%d", &b);

   printf("Enter 3rd Number: ");

   scanf("%d", &c);


   max = a;


   max = (max < b ? b : max);

   max = (max < c ? c : max);


   printf("\nGreatest Number is: %d", max);

}
```

Output:

```
Enter 1st Number: 1
Enter 2nd Number: 2
Enter 3rd Number: 3

Greatest Number is: 3
```

SET-3 (PRACTICAL-2)

Aim: Any year is input through the keyboard. Write a program to determine whether the year is a leap year or not. Use the logical operators && and ||.

Source Code:

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

   printf("Enter the year: ");
   scanf("%d", &year);

   if ((year % 4 == 0 || year % 400 == 0) && (year % 100 != 0))
   {
      printf("%d is leap year", year);
   }
   else
      printf("%d is not a leap year", year);
   return 0;
}
```

Output:

```
Enter the year: 2024
2024 is leap year
```

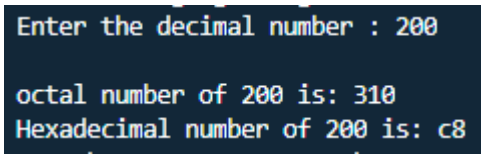
SET-4 (PRACTICAL-1)

Aim: Write a program to convert the decimal number into octal and hexadecimal format Hint: %o and %x

Source Code:

```
#include <stdio.h>

int main()

{

   int n, temp;


   printf("Enter the decimal number : ");

   scanf("%d", &n);


   temp = n;


   printf("\noctal number of %d is: %o", n, n);

   printf("\nHexadecimal number of %d is: %x", temp, temp);


   return 0;

}
```

Output:



SET-4 (PRACTICAL-2)

Aim: Write a C Program to Print multiplication table of number entered by user.

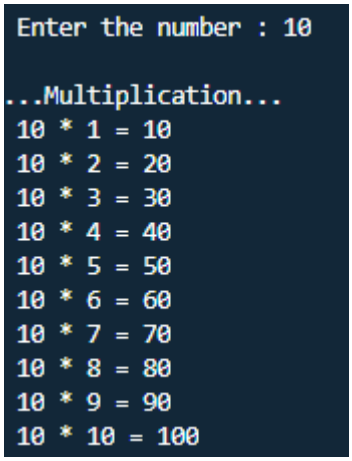
Source Code:

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

```
int main()
{
   int a, i = 1;

   printf("\n Enter the number : ");
   scanf("%d", &a);

   printf("\n...Multiplication...");

   printf("\n %d * %d = %d", a, i, a * i);
   i++;
   printf("\n %d * %d = %d", a, i, a * i);
   i++;
   printf("\n %d * %d = %d", a, i, a * i);
   i++;
   printf("\n %d * %d = %d", a, i, a * i);
   i++;
   printf("\n %d * %d = %d", a, i, a * i);
   i++;
   printf("\n %d * %d = %d", a, i, a * i);
   i++;
   printf("\n %d * %d = %d", a, i, a * i);
   i++;
   printf("\n %d * %d = %d", a, i, a * i);
   i++;
   printf("\n %d * %d = %d", a, i, a * i);
   i++;
   printf("\n %d * %d = %d", a, i, a * i);
   i++;
   printf("\n %d * %d = %d", a, i, a * i);
   i++;

   return 0;
```

}

Output:



SET-5 (PRACTICAL-1)

Aim: If the cost price and selling price of an item is input through the keyboard, write a program to determine whether the seller has made profit or incurred loss. Also determine how much profit he made or loss he incurred.

Source Code:

```
#include <stdio.h>

int main()

{

   float cp, sp, profit, loss;


   printf("Enter the cost price: ");

   scanf("%f", &cp);


   printf("Enter the selling price: ");

   scanf("%f", &sp);


   if (sp > cp)
```

```
    {
        profit = sp - cp;
        printf("\nWe earn %f profit", profit);
    }
    else if (sp < cp)
    {
        loss = cp - sp;
        printf("\nWe incurred %f loss", loss);
    }
    else
    {
        printf("\nwe don't get any profit and loss");
    }
    return 0;
}
```

Output:

```
Enter the cost price: 2000
Enter the selling price: 3000

We earn 1000.000000 profit
```

SET-5 (PRACTICAL-2)

Aim: If the ages of Ram, Shyam and Ajay are input through the keyboard, write a program to determine the youngest of the three. (Hint: Use Nested Switch Statement).

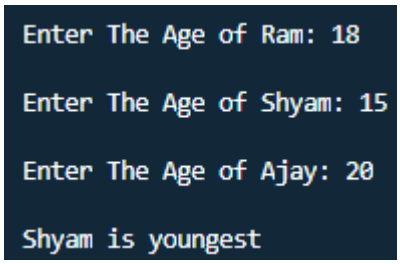
Source Code:

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

```
int ram, shyam, ajay;

printf("\nEnter The Age of Ram: ");
scanf("%d", &ram);

printf("\nEnter The Age of Shyam: ");
scanf("%d", &shyam);

printf("\nEnter The Age of Ajay: ");
scanf("%d", &ajay);

switch (ram < shyam)
{
case 0:
    switch (shyam < ajay)
    {
    case 0:
        printf("\nAjay is youngest than Ram and Shyam");
        break;
    case 1:
        printf("\nShyam is youngest");
        break;
    }
    break;
case 1:
    switch (ram < ajay)
    {
    case 0:
        printf("\nAjay is youngest than Ram and Shyam");
        break;
    case 1:
```

```
            printf("\nRam is youngest");
            break;
        }
        break;
    }
}
```

Output:



SET-5 (PRACTICAL-3)

Aim: The policy followed by a company to process customer orders is given by the following rules:

a) If a customer order is less than or equal to that in stock and 'has credit' is OK, supply 'has requirements'.

b) If 'has credit' is not OK do not supply. Send him intimation.

c) If 'has credit' is OK but the item in stock is less than 'has ordered', supply what is in stock and Intimate him that the balance will be refunded.

Write a C program to implement the company policy.

Source Code:

```
#include <stdio.h>
int main()
{
   int stock = 500, order;
   char credit;
```

```
    printf("\n Enter the credit : ");
    scanf("%c", &credit);

    fflush(stdin);

    printf("\n Enter the Customer Order : ");
    scanf("%d", &order);

    fflush(stdin);

    printf("\n Availablr Stock: %d", stock);

    if (order <= stock && credit == 'y' || credit == 'Y')
        printf("\n Supply has requirement", order);

    else if (credit == 'n' || credit == 'N')
        printf("\n Do not Supply");

    else
        printf("\n Supplied products are %d and balance will be refunded", stock);

    return 0;
}
```

Output:

```
Enter the credit : 1000

Enter the Customer Order : 700

Availablr Stock: 500
Supplied products are 500 and balance will be refunded
```

SET-6 (PRACTICAL-1)

Aim: Two numbers are entered through the keyboard. Write a program to find the value of one number raised to the power of another. (Use While loop)

Source Code:

```
#include <stdio.h>

int main()

{

  int count = 1;

  int n1 = 2;

  int n2 = 10;

  int result = 1;


  while (count <= n2)

  {

    result = n1 * result;

    count++;

  }

  printf("Result: %d", result);


  return 0;

}
```

Output:

```
Result: 1024
```

SET-6 (PRACTICAL-2)

Aim: Write a program to print the multiplication table of the number entered from the keyboard. The table should get displayed in the following form:

12*1=12

12*2=24

….

Source Code:

```
#include <stdio.h>
int main()
{
   int n, i;

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

   for (i = 1; i <= 10; i++)
   {
      printf("%d * %d = %d \n", n, i, n * i);
   }
   return 0;
}
```

Output:

```
12 * 1 = 12
12 * 2 = 24
12 * 3 = 36
12 * 4 = 48
12 * 5 = 60
12 * 6 = 72
12 * 7 = 84
12 * 8 = 96
12 * 9 = 108
12 * 10 = 120
```

SET-6 (PRACTICAL-3)

Aim: Write a menu driven program which has following options: 1. Prime or not 2. Perfect number or not 3. Factorial of a number 4. Exit Use do...while statement so that the menu is displayed at least once. Also use Switch statement.

Source Code:

```
#include <stdio.h>
int main()
{
   int c, i, n1, n2, n3, sum = 0;
   long fact = 1;

   do
   {
      printf("\n 1.Prime or not");
      printf("\n 2.Perfect or not");
      printf("\n 3.Factorial");
      printf("\n 4.Exit");

      printf("\n Enter your choice : ");
      scanf("%d", &c);

      switch (c)
      {
      case 1:
         printf("\n Enter The Number : ");
         scanf("%d", &n1);

         for (i = 2; i <= n1; i++)
         {
            if (n1 % i != 0)
            {
               printf("\n Given number is prime number");
            }
```

```
        else
        {
            printf("\n Given number is not prime number");
        }
        break;
    }

case 2:
    printf("\n Enter The Number : ");
    scanf("%d", &n2);

    for (i = 1; i < n2; i++)
    {
        if (n2 % i == 0)
        {
            sum = sum + i;
        }
    }
    if (sum == n2)
    {
        printf("\n Given number is a Perfect number");
    }
    else
    {
        printf("\n Given number is not a Perfect number");
    }
    break;

case 3:
    printf("\n Enter The Number : ");
    scanf("%d", &n3);
```

```
        for (i = 1; i <= n3; i++)
        {
           fact = fact * i;
        }
        printf("\n Factorial of %d is %ld", n3, fact);
        break;

      case 4:
        exit(0);
        break;
      }
   } while (1);
}
```

Output:

```
1.Prime or not
2.Perfect or not
3.Factorial
4.Exit
Enter your choice : 3

Enter The Number : 5

Factorial of 5 is 120
1.Prime or not
2.Perfect or not
3.Factorial
4.Exit
Enter your choice : 2

Enter The Number : 6

Given number is a Perfect number
1.Prime or not
2.Perfect or not
3.Factorial
4.Exit
Enter your choice : 1

Enter The Number : 29

Given number is prime number
```

SET-6 (PRACTICAL-4)

Aim: Write a program for a match-stick game between the computer and a user. Your Program should ensure that the computer always wins. Rules for the games are as follows:

● There are 21 match-sticks.

● The computer asks the player to pick 1, 2, 3, or 4 match-sticks.

● After the person picks, the computer does its picking.

● Whoever is forced to pick up the last match-stick loses the game. Use while loop, break and Continue Statements.

Source Code:

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

```
int main()
{
   int m = 21, user_choice, computer_choice;

   while (1)
   {
      printf("\nTotal match-sticks = %d\n", m);

      printf("Pick 1 or 2 or 3 or 4 matches\n");
      scanf("%d", &user_choice);

      if (user_choice > 4 || user_choice < 1)
         continue;

      m = m - user_choice;

      printf("Number of matches left = %d\n", m);

      computer_choice = 5 - user_choice;

      printf("out of which computer picked up %d\n", computer_choice);

      m = m - computer_choice;

      if (m == 1)
      {
         printf("\nNumber of matche-sticks left = %d\n", m);
         printf("You lost the Game\n");
         break;
      }
   }
```

```
    return 0;
}
```

Output:

```
Total match-sticks = 21
Pick 1 or 2 or 3 or 4 matches
1
Number of matches left = 20
out of which computer picked up 4

Total match-sticks = 16
Pick 1 or 2 or 3 or 4 matches
2
Number of matches left = 14
out of which computer picked up 3

Total match-sticks = 11
Pick 1 or 2 or 3 or 4 matches
3
Number of matches left = 8
out of which computer picked up 2

Total match-sticks = 6
Pick 1 or 2 or 3 or 4 matches
4
Number of matches left = 2
out of which computer picked up 1

Number of matche-sticks left = 1
You lost the Game
```

SET-7 (PRACTICAL-1)

Aim: Twenty five numbers are entered from the keyboard into an array. Write a program to find out how many of them are positive, negative, how many are even and odd.

Source Code:

```
#include <stdio.h>

int main()
```

```
{
   int arr[25], i, pos = 0, neg = 0, even = 0, odd = 0;

   printf("\nEnter the 25 Elements: ");

   for (i = 0; i < 25; i++)
      scanf("%d", &arr[i]);

   for (i = 0; i < 25; i++)
   {
      if (arr[i] >= 0)
      {
         pos++;
      }
      else
      {
         neg++;
      }

      if (arr[i] % 2 == 0)
      {
         even++;
      }
      else
      {
         odd++;
      }
   }
   printf("\nTotal Postive numbers: %d", pos);
   printf("\nTotal Negative numbers: %d", neg);
   printf("\nTotal even numbers: %d", even);
```

```
    printf("\nTotal odd numbers: %d", odd);


    return 0;
}
```

Output:

```
Enter the 25 Elements: 1 2 3 4 5 -6 -7 -8 -9 -10 -11 -12 13 14 15 16 17 18 19 20 -21 -22 23 24 25

Total Postive numbers: 16
Total Negative numbers: 9
Total even numbers: 12
Total odd numbers: 13
```

SET-7 (PRACTICAL-2)

Aim: Write a program for creating two arrays of different size and merge both arrays into one by sorting those arrays in ascending order. [Merge by sorting]

Source Code:

```
#include <stdio.h>
void main()
{

    int array1[50], array2[50], array3[100], m, n, i, j, k = 0;
    printf("\nEnter size of array Array 1: ");
    scanf("%d", &m);

    printf("\nEnter sorted elements of array 1: ");
    for (i = 0; i < m; i++)
    {
        scanf("%d", &array1[i]);
    }

    printf("\nEnter size of array 2: ");
```

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

printf("\nEnter sorted elements of array 2: ");
for (i = 0; i < n; i++)
{
   scanf("%d", &array2[i]);
}

i = 0;
j = 0;

while (i < m && j < n)
{
   if (array1[i] < array2[j])
   {
      array3[k] = array1[i];
      i++;
   }

   else
   {
      array3[k] = array2[j];
      j++;
   }
   k++;
}

if (i >= m)
{
   while (j < n)
   {
```

```
            array3[k] = array2[j];
            j++;
            k++;
        }
    }

    if (j >= n)
    {
        while (i < m)
        {
            array3[k] = array1[i];
            i++;
            k++;
        }
    }

    printf("\nAfter merging: ");
    for (i = 0; i < m + n; i++)
    {
        printf("\n%d", array3[i]);
    }
}
```

Output:

```
Enter sorted elements of array 1: 1 4

Enter size of array 2: 3

Enter sorted elements of array 2: 3 5 9

After merging:
1
3
4
5
9
```

SET-7 (PRACTICAL-3)

Aim: Write a Program to multiply any two 3*3 Matrices.

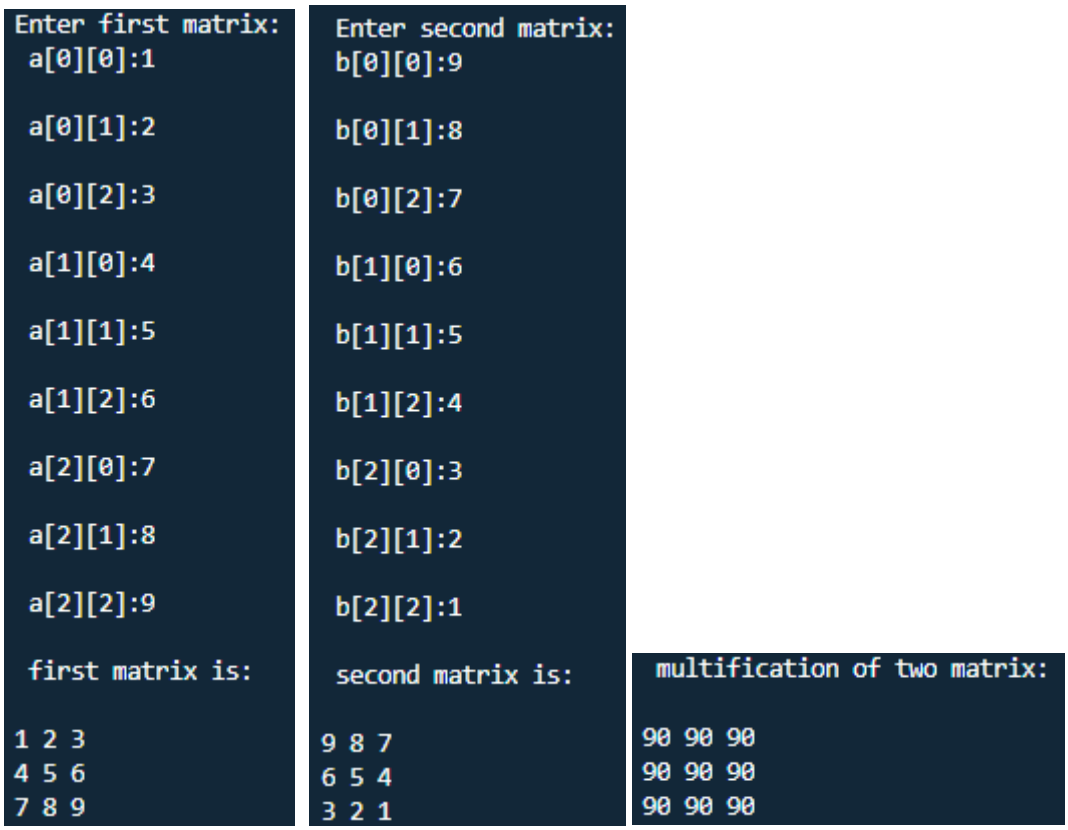
Source Code:

```
#include <stdio.h>
#define r 3
#define c 3

int main()
{
   int a[r][c], b[r][c], sum, i, j, k;

   printf("Enter first matrix:");

   for (i = 0; i < r; i++)
   {
      for (j = 0; j < c; j++)
      {
         printf("\n a[%d][%d]:", i, j);
         scanf("%d", &a[i][j]);
      }
   }

   printf("\n first matrix is: \n");
   for (i = 0; i < r; i++)
   {
      printf("\n");

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

```
    {
        printf("%d ", a[i][j]);
    }
}

printf("\n Enter second matrix:");

for (i = 0; i < r; i++)
{
    for (j = 0; j < c; j++)
    {
        printf("\n b[%d][%d]:", i, j);
        scanf("%d", &b[i][j]);
    }
}

printf("\n second matrix is: \n");
for (i = 0; i < r; i++)
{
    printf("\n");

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

        printf("%d ", b[i][j]);
    }
}

for (i = 0; i < r; i++)
{
    for (j = 0; j < c; j++)
```

```
        {
            sum = 0;
            for (k = 0; k < r; k++)
            {
                sum = sum + a[i][k] * b[k][j];
            }
        }
    }

    printf("\n multification of two matrix: \n");

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

    {
        printf("\n");

        for (j = 0; j < c; j++)
        {
            printf("%d ", sum);
        }
    }

    return 0;
}
```

Output:



SET-8 (PRACTICAL-1)

Aim: Take a user input for a string and calculate the number of alphabets, digits and special characters from the given input.

Source Code:

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

int main()
{
   char name[100];
   int Digits, Alphabet, SpecialChar, Spaces;
   int counter;
```

```
   Digits = Alphabet = SpecialChar = Spaces = 0;

   printf("Enter a string: ");
   gets(name);

   for (counter = 0; name[counter] != '\0'; counter++)
   {

      if (name[counter] >= '0' && name[counter] <= '9')
         Digits++;

      else if ((name[counter] >= 'A' && name[counter] <= 'Z') || (name[counter] >= 'a' &&
name[counter] <= 'z'))
         Alphabet++;

      else if (name[counter] == ' ')
         Spaces++;

      else
         SpecialChar++;
   }

   printf("\nDigits: %d", Digits);
   printf("\nAlphabets: %d", Alphabet);
   printf("\nSpaces: %d", Spaces);
   printf("\nSpecial Characters: %d", SpecialChar);

   return 0;
}
```

Output:

```
Enter a string: Yatharth @ 2362

Digits: 4
Alphabets: 8
Spaces: 2
Special Characters: 1
```

SET-8 (PRACTICAL-2)

Aim: Write a program that takes a set of names of individuals and abbreviates the first, middle and other names except the last name by their first letter.

Source Code:

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

void main()
{
   int i, n;
   char a[20], b[20], c[20];

   printf("Hello World");

   printf("\nEnter your first name:\n");
   scanf("%s", &a[i]);

   printf("enter your middle name:\n");
   scanf("%s", &b[i]);

   printf("Enter your last name:\n");
   scanf("%s", &c[i]);
```

```
    printf("Displaying your name:\n");
    printf("%c", a[0]);
    printf(" ");
    printf("%c", b[0]);
    printf(" ");

    n = strlen(c);

    for (i = 0; i <= n; i++)
    {
        printf("%c", c[i]);
    }
}
```

Output:

```
Enter your first name: Yatharth
enter your middle name: Tejaskumar
Enter your last name: Chauhan
Displaying your name: Y T Chauhan
```

SET-8 (PRACTICAL-3)

Aim: Write a C program to check if the user inputed string is palindrome or not using recursion.

Source Code:

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

void palindrome(char[], int);

int main()
{
```

```
    char word[15];

    printf("\nEnter a word to check if it is a palindrome: ");
    scanf("%s", word);

    palindrome(word, 0);

    return 0;
}

void palindrome(char word[], int i)
{
    int len = strlen(word) - (i + 1);

    if (word[i] == word[len])
    {
        if (i + 1 == len || i == len)
        {
            printf("The entered word is a palindrome\n");
            return;
        }
        palindrome(word, i + 1);
    }
    else
    {
        printf("The entered word is not a palindrome\n");
    }
}
```

Output:

```
Enter a word to check if it is a palindrome: MOM
The entered word is a palindrome
```

SET-9 (PRACTICAL-1)

Aim: Write a C program to check if the entered number is prime or not by using types of user defined functions

(i) No arguments passed and no return value

(ii) No arguments passed but a return value

(iii) Argument passed but no return value

(iv) Argument passed and a return value

Source Code:

```
//(i) No arguments passed and no return value


#include <stdio.h>
void PrimeNumber();
main()
{
   PrimeNumber(); // argument is not passed
   return 0;
}
// return type is void meaning doesn't return any value
void PrimeNumber()
{
   int n, i, flag = 0;
   printf("Enter a positive integer: ");
   scanf("%d", &n);
   for (i = 2; i <= n / 2; ++i)
   {
      if (n % i == 0)
      {
         flag = 1;
```

```
        }
    }
    if (flag == 1)
        printf("%d is not a prime number.", n);
    else
        printf("%d is a prime number.", n);
}
```

Output:

```
Enter a positive integer: 7
7 is a prime number.
```

Source Code:

```
//(ii) No arguments passed but a return value


#include <stdio.h>
int getInteger();
int main()
{
    int n, i, flag = 0;
    // no argument is passed
    n = getInteger();
    for (i = 2; i <= n / 2; ++i)
    {
        if (n % i == 0)
        {
            flag = 1;
            break;
        }
    }
    if (flag == 1)
        printf("%d is not a prime number.", n);
```

```
    else
        printf("%d is a prime number.", n);
    return 0;
}
// returns integer entered by the user
int getInteger()
{
    int n;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    return n;
}
```

Output:

```
Enter a positive integer: 16
16 is not a prime number.
```

Source Code:

```
//(iii) Argument passed but no return value

#include <stdio.h>
void PrimeAndDisplay(int n);
int main()
{
    int n;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    // n is passed to the function
    PrimeAndDisplay(n);
    return 0;
}
// return type is void meaning doesn't return any value
```

```
void PrimeAndDisplay(int n)
{
   int i, flag = 0;
   for (i = 2; i <= n / 2; ++i)
   {
      if (n % i == 0)
      {
         flag = 1;
         break;
      }
   }
   if (flag == 1)
      printf("%d is not a prime number.", n);
   else
      printf("%d is a prime number.", n);
}
```

Output:

```
Enter a positive integer: 100
100 is not a prime number.
```

Source Code:

(iv) Argument passed and a return value

```
#include <stdio.h>
int PrimeNumber(int n);
int main()
{
   int n, flag;
   printf("Enter a positive integer: ");
   scanf("%d", &n);
   // n is passed to the checkPrimeNumber() function
```

```
    // the returned value is assigned to the flag variable
    flag = PrimeNumber(n);

    if (flag == 1)
        printf("%d is not a prime number", n);
    else
        printf("%d is a prime number", n);
    return 0;
}
// int is returned from the function
int PrimeNumber(int n)
{
    int i;
    for (i = 2; i <= n / 2; ++i)
    {
        if (n % i == 0)
            return 1;
    }
    return 0;
}
```

Output:

```
Enter a positive integer: 71
71 is a prime number
```

SET-9 (PRACTICAL-2)

Aim: If the length of the sides of a triangle are denoted by a, b and c, then the area of triangle is given by:

S=a+b+c/2

A=rootof(s(s-a)*(s-b)*(s-c))

Source Code:

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

void area(float a, float b, float c);
int main()
{
   float a, b, c;

   printf("Enter Side of a: ");
   scanf("%f", &a);

   printf("Enter Side of b: ");
   scanf("%f", &b);

   printf("Enter Side of c: ");
   scanf("%f", &c);

   area(a, b, c);
}
void area(float a, float b, float c)
{
   float S, AREA, A;

   S = (a + b + c) / 2;
   AREA = sqrt(S * (S - a) * (S - b) * (S - c));

   printf("Area of Triangle is: %f", AREA);
}
```

Output:

```
Enter Side of a: 4
Enter Side of b: 4
Enter Side of c: 4
Area of Triangle is: 6.928203
```

SET-9 (PRACTICAL-3)

Aim: A positive integer is entered through the keyboard, write a function to find the binary equivalent of this number using recursion.

Source Code:

```
#include <stdio.h>
int rec_bin(int n);
int main()
{
   int n;
   printf("Enter a Number: ");
   scanf("%d", &n);

   printf("Decimal To Binary Using Recursion: %d", rec_bin(n));
}

int rec_bin(int n)
{
   if (n == 0)
   {
      return 0;
   }
   else
```

```
    {
        return ((n % 2) + 10 * rec_bin(n / 2));
    }
}
```

Output:

```
Enter a Number: 23
Decimal To Binary Using Recursion: 10111
```

SET-10 (PRACTICAL-1)

Aim: Write a C program to create a structure of Book Detail and display the details of the book in appropriate format by passing structure as function argument.

Source Code:

```
#include <stdio.h>
struct book_detail
{
    char title[32];
    char author[32];
    char publisher[32];
    int price;
    int pages;
} b1;
int main()
{
    printf("\nEnter book title: ");
    gets(b1.title);
    printf("\nEnter author name: ");
    gets(b1.author);
    printf("\nEnter publisher: ");
```

```
    gets(b1.publisher);
    printf("\nEnter price of book: ");
    scanf("%d", &b1.price);
    printf("\nEnter Pages of Book: ");
    scanf("%d", &b1.pages);

    printf("\nBook details are:");

    printf("\nThe title of book is: %s", b1.title);
    printf("\nThe name of author: %s", b1.author);
    printf("\nName of publisher: %s",b1.publisher);
    printf("\nPrice of book: %d",b1.price);
    printf("\nPages of Book: %d", b1.pages);
}
```

Output:

```
Enter author name: Yatharth Chauhan

Enter publisher: YC

Enter price of book: 1000

Enter Pages of Book: 1500

Book details are:
The title of book is: C Langauge
The name of author: Yatharth Chauhan
Name of publisher: YC
Price of book: 1000
Pages of Book: 1500
```

SET-10 (PRACTICAL-2)

Aim: Create a Union called library to hold accession number, title of the book ,author name, price of the book and flag indicating whether the book is issued or not.(flag = 1 if the book is issued , flag = 0 otherwise). Write a program to enter data of one book and display the data.

Source Code:

```
#include <stdio.h>
union book_detail
{
   char title[40];
   char author[40];
   int price;
   int accession_no;
} b1;
int main()
{
   int flag;

   printf("\nEnter the title of book: ");
   gets(b1.title);

   printf("\nEnter the author of book: ");
   gets(b1.author);

   printf("\nEnter Accession no of book: ");
   scanf("%d", &b1.accession_no);

   printf("\nEnter price of books: ");
   scanf("%d", &b1.price);

   printf("\nEnter your choice");
   printf("\n1. book is issued");

   printf("\n2. book is not issued");
   scanf("%d", &flag);

   printf("\n Book details are :");
```

```
    printf("\nTitle: %s", b1.title);
    printf("\nAuthor %s", b1.author);
    printf("\n Accession_no %d", b1.accession_no);
    printf("\n price %d ", b1.price);
}
```

Output:

```
Enter the title of book: C Language

Enter the author of book: Yatharth Chauhan

Enter Accession no of book: 5

Enter price of books: 700

Enter your choice
1. book is issued
2. book is not issued: 1

 Book details are :
Title: ╝☻
Author ╝☻
Accession_no 700
price 700
```

SET-10 (PRACTICAL-3)

Aim: Write a C program for nested structure to display employee details such as, Age, Name, Address, Salary.

Source Code:

```
#include <stdio.h>
struct employee
{
    char Name[50];
    char ad[50];
    int Age;
```

```
    struct salary
    {
       int sal;
    } s;
} Emp;
void main()
{
    printf("\nEnter Employee Name: ");
    gets(Emp.Name);

    printf("\nEnter Employee address: ");
    gets(Emp.ad);

    printf("\nEnter Employee Age: ");
    scanf("%d", &Emp.Age);

    printf("\nEnter Employee Salary: ");
    scanf("%d", &Emp.s.sal);

    printf("\nThe name of employee is: %s", Emp.Name);
    printf("\nThe address of employee: %s", Emp.ad);
    printf("\nAge of employee: %d", Emp.Age);
    printf("\nSalary of employee: %d", Emp.s.sal);
}
```

Output:

```
Enter Employee Name: Yatharth Chauhan

Enter Employee address: Bharuch

Enter Employee Age: 18

Enter Employee Salary: 90000

The name of employee is: Yatharth Chauhan
The address of employee: Bharuch
Age of employee: 18
Salary of employee: 90000
```