**Submitted by-**

**Abhishek Kumar**

**IT BRANCH**

**CP ASSIGNMENT -1**

**1. WAP(Write A Program) to print ‘See C is Sea’ five times on the console.**

#include <stdio.h>

int main()

{

    for(int i=0;i<5;i++)

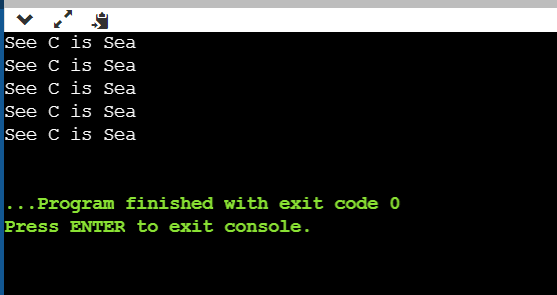
    {

        printf("See C is Sea \n");

    }

    return 0;

}

****

**2. WAP that will print your mailing address in the following format**

**First line: <Your Name>**

**Roll No.: <109389>**

**Branch: <ECE>**

#include <stdio.h>

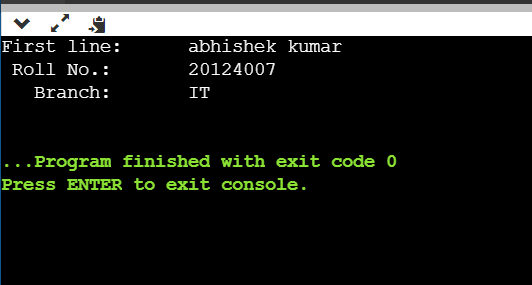
int main()

{

        printf("First line:   \t abhishek kumar   \n Roll No.:   \t 20124007    \n   Branch:   \t IT   \n");

    return 0;

}



**3. WAP which will accept an integer, a decimal number, a character and a string from the keyboard and display them one per line.**

#include <stdio.h>

int main()

{

    printf("enter an integer, a decimal number, a character and a string one by one  \n");

    int a;

    float b;

    char c;

    char d[10];

    scanf("%d %f %c",&a,&b,&c);

    scanf("%s",d);

    printf("the integer you entered is %d \n ",a);

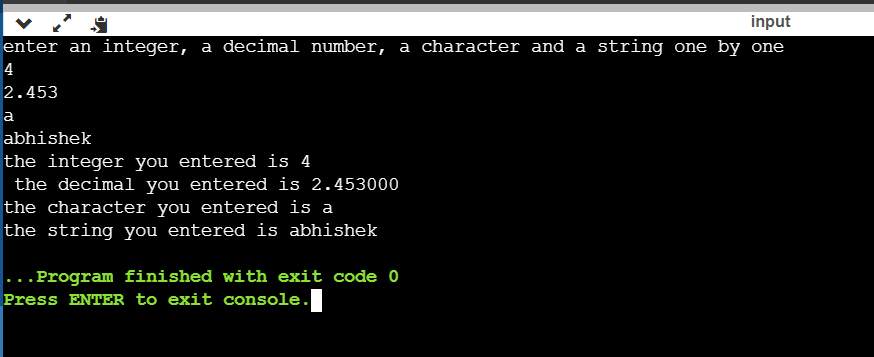
    printf("the decimal you entered is %f \n",b);

    printf("the character you entered is %c \n",c);

    printf("the string you entered is %s",d);

    return 0;

}



**4. WAP which will accept two integers (a and b) from the input device and display the results of their sum, difference, product, division and mod on the console by assuming a simple arithmetic calculator.**

#include <stdio.h>

int main()

{

    printf("enter the two values of a and b \n");

    int a;

    int b;

    scanf("%d %d",&a,&b);

    printf("the sum of numbers you entered is %d \n ",a+b);

    printf("the substraction of numbers you entered is %d \n",a-b);

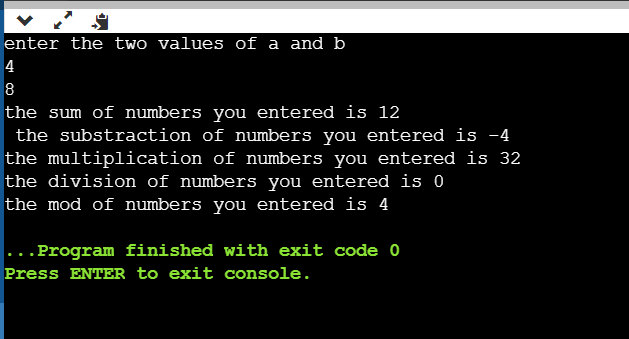
    printf("the multiplication of numbers you entered is %d \n",a\*b);

    printf("the division of numbers you entered is %d \n",a/b);

    printf("the mod of numbers you entered is %d",a%b);

    return 0;

}



**5. Relationship between Celsius (C) and Fahrenheit (F) is governed by the following formula**

**F = 9C/5 + 32 WAP to convert the Celsius to Fahrenheit and vice versa.**

#include <stdio.h>

int main()

{

    char choice;

    float value;

    printf("enter the option \n enter '1' to convert the Celsius to Fahrenheit \n enter '2' to convert the Fahrenheit to Celsius \n");

    scanf("%c",&choice);

    printf("enter the value to convert \n");

    scanf("%f",&value);

    switch(choice)

    {

        case '1' : printf("the convertion of Celsius to Fahrenheit is %f \n ",9\*(value/5) + 32);

        break;

        case '2' :printf("the convertion of convert the Fahrenheit to Celsius is %f \n",5\*((value-32)/9));

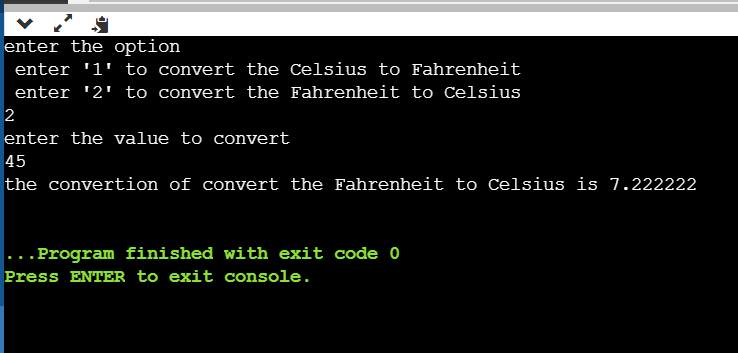
        break;

        default : printf("entered choice is wrong \n");

        break;

    }return 0;

}



**6. WAP to compute the gross salary of an employee whose details are given below:**

**Basic pay: Rs.50, 000, dearness allowance: 50% of Basic pay, house rent allowance: 20% of Basic pay, vehicle allowance: 10% of the Basic pay.**

#include <stdio.h>

int main()

{

    printf("the Basic pay of employee is Rs.50, 000 \n");

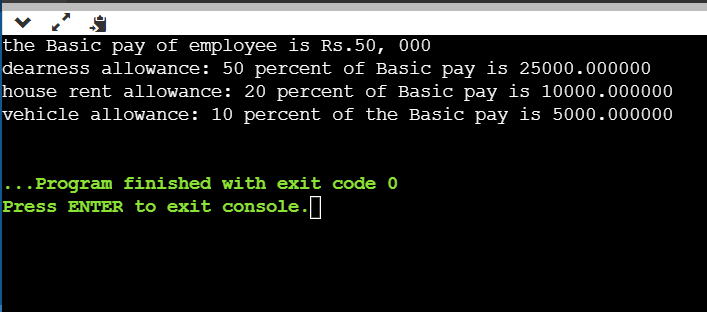
    printf("dearness allowance: 50 percent of Basic pay is %f \n",50000\*0.5);

    printf("house rent allowance: 20 percent of Basic pay is %f \n",50000\*0.2);

     printf("vehicle allowance: 10 percent of the Basic pay is %f \n",50000\*0.1);

    return 0;

}



**7. WAP which will print the ‘$’ symbol in the following format. (Hint: No need to use any loops)**

**$**

**$ $**

**$ $ $**

**$ $ $ $**

**$ $ $ $ $**

#include <stdio.h>

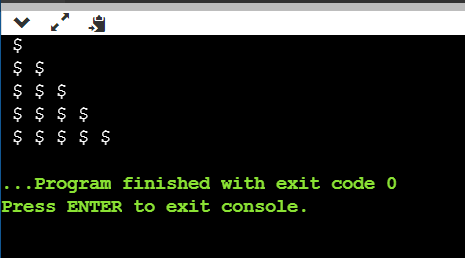
int main()

{

    printf(" $ \n $ $ \n $ $ $ \n $ $ $ $ \n $ $ $ $ $");

    return 0;

}



**8 a. WAP to swap two numbers using and without using a third variable.**

#include <stdio.h>

int main()

{

    int a,b;

    printf("enter the values of a and b to be swaped ");

    scanf("%d%d",&a,&b);

    a=a+b;

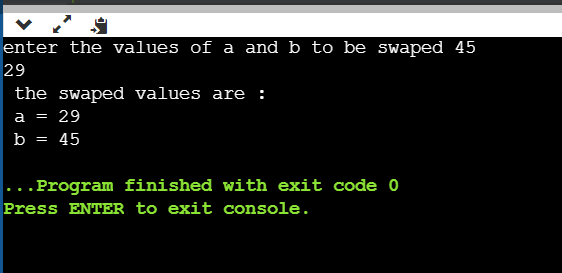
    b=a-b;

    a=a-b;

    printf(" the swaped values are :\n a = %d \n b = %d ",a,b);

    return 0;

}



8 b.

#include <stdio.h>

int main()

{

    int a,b,temp;

    printf("enter the values of a and b to be swaped ");

    scanf("%d%d",&a,&b);

    temp=a;

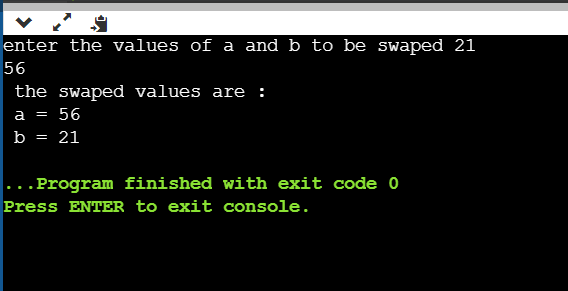
    a=b;

    b=temp;

    printf(" the swaped values are :\n a = %d \n b = %d ",a,b);

    return 0;

}



**9. WAP which will accept a three digits integer number and display the sum and product of all the digits of that number. (Hint: Use / and % operators)**

#include <stdio.h>

int main()

{

    int a;

    printf("enter the three digit number \n ");

    scanf("%d",&a);

    int sum=0,product=1;

    for(int i=0;i<3;i++)

    {

        sum=sum+a%10;

        product=product\*(a- (a/10)\*10);

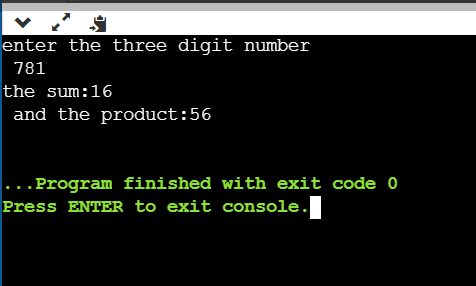
        a=a/10;

    }

    printf("the sum:%d \n and the product:%d \n",sum,product);

    return 0;

}



**10. WAP to find the size of various primitive data types used in C such as *int*, *float, double and char*.**

#include <stdio.h>

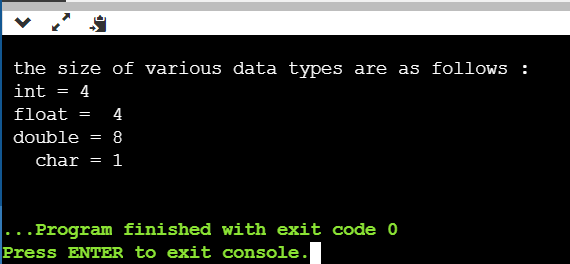
int main()

{

    printf(" the size of various data types are as follows :\n int = %d \n float =  %d  \n double = %d \n   char = %d \n ",sizeof(int),sizeof(float),sizeof(double),sizeof(char));

    return 0;

}



**Homework questions:**

**1. The line joining the points (2, 2) and (5, 6) which lie on the circumference of a circle is the**

**diameter of the circle. WAP to compute the area and perimeter of the circle.**

#include <stdio.h>

#include <math.h>

int main ()

{

  printf("the line joining the points (2,2) & (5,6) is on the diameter of the circle ;\n");

  printf("the coordinates of the center of the circle is given by :  (%f,%f) \n",((2.0+5)/2),((2.0 + 6)/2));

  float dia = sqrt (pow ((5 - 2), 2) + pow ((6 - 2), 2));

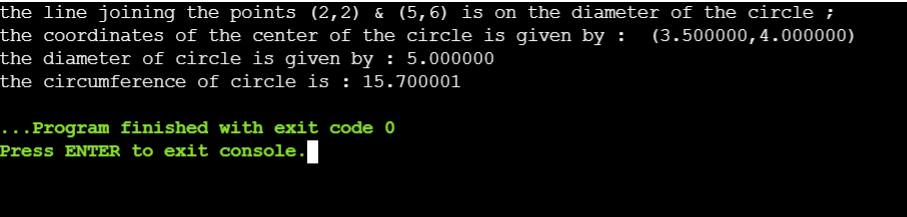
  float pie =3.14;

  printf("the diameter of circle is given by : %f \n",dia);

  printf("the circumference of circle is : %f", pie\*dia);

  return 0;

}



**2. The price of one pen is Rs.10, one pencil is Rs. 5, and one sharpener is Rs. 2. You purchased 2 pens, 3 pencils and 1 sharpener. Compute the total price as per the following format.**

**\*\*\*\*\*\* LIST OF ITEMS \*\*\*\*\*\***

**Item Price (Rs.) Total (Rs.)**

**Pen 10 20**

**Pencil 5 15**

**Sharpener 2 2**

**Grand Total (Rs.) 17 37**

#include <stdio.h>

#include <math.h>

int main ()

{

  printf("the price of a pen is rs 10 , a pencil is rs 5 and a sharpner is rs 2 ;\n");

  printf("if we bought 2 pens , 3 pencils and 1 sharpner ;\n");

  printf("the price of 2 pens is :  %d \n", 2\*10);

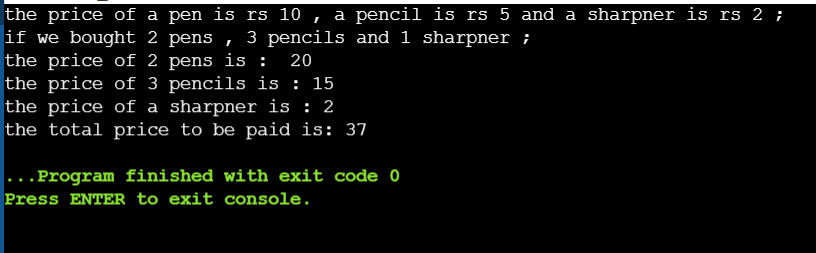
  printf("the price of 3 pencils is : %d\n", 3\*5);

  printf("the price of a sharpner is : %d\n",1\*2);

  printf("the total price to be paid is: %d",2\*10+3\*5+1\*2);

  return 0;

}



**3. WAP which will accept two floating point numbers, assign their sum to an integer variable and then output the values of all three variables.**

#include <stdio.h>

#include <math.h>

int main ()

{

  float v1,v2;

  int sum;

  printf("enter the first floating point no.:\n");

  scanf("%f",&v1);

  printf("enter the second floating point no.:\n");

  scanf("%f",&v2);

  sum=v1+v2;

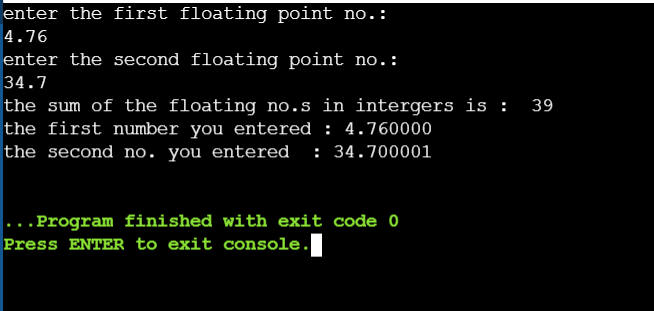
  printf("the sum of the floating no.s in intergers is :  %d \n", sum);

  printf("the first number you entered : %f\n", v1);

  printf("the second no. you entered  : %f\n",v2);

  return 0;

}



**4. Extend the Q8 for three variables a, b, and c such that a holds the value of c, b holds the value of a, and c holds the value of b using and without using a third variable.**

4a

#include <stdio.h>

int main()

{

    int a,b,c;

    printf("/\*we are swaping the three variables a, b, and c\n");

    printf("such that a holds the value of c, b holds the value of a, and c holds the value of b\*/");

    printf("\n enter the values of a ,b and c to be swaped \n");

    scanf("%d%d%d",&a,&b,&c);

    a=a+b+c;

    b=a-(b+c);

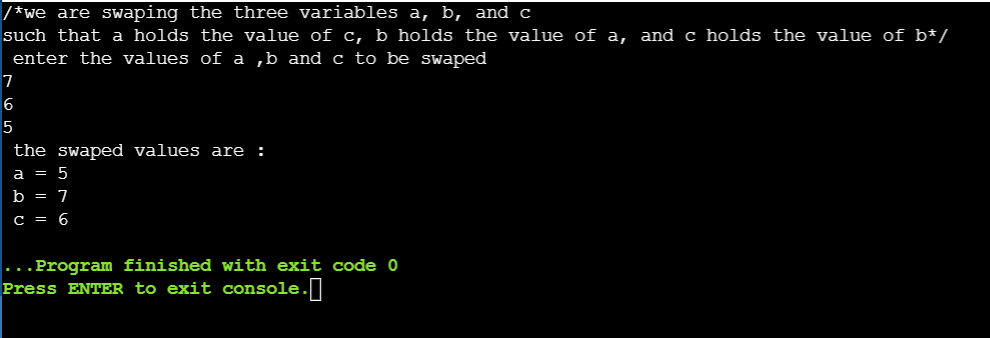
    c=a-(b+c);

    a=a-(b+c);

    printf(" the swaped values are :\n a = %d \n b = %d \n c = %d",a,b,c);

    return 0;

}



4b

#include <stdio.h>

int main()

{

    int a,b,c,temp;

    printf("/\*we are swaping the three variables a, b, and c\n");

    printf("such that a holds the value of c, b holds the value of a, and c holds the value of b\*/");

    printf("\nenter the values of a ,b and c to be swaped :\n");

    scanf("%d%d%d",&a,&b,&c);

    temp=a;

    a=c;

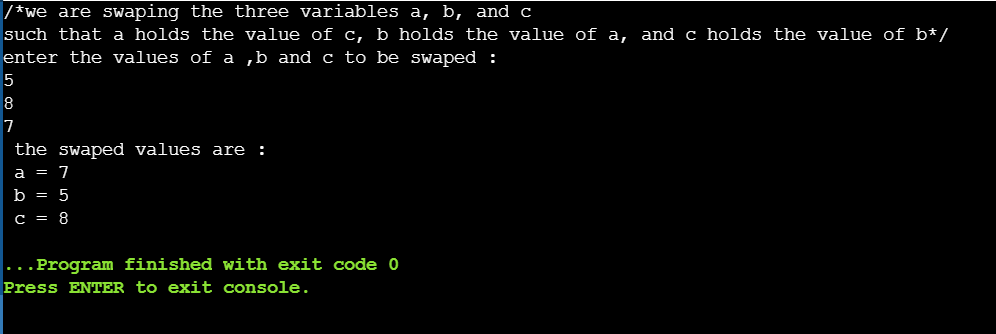
    c=b;

    b=temp;

    printf(" the swaped values are :\n a = %d \n b = %d \n c = %d",a,b,c);

    return 0;

}



**5. Area (A) of a triangle is given by the formula A= √(s(s-a)(s-b)(s-c)) Where a, b and c are sides of the triangle and 2s = a + b + c.**

**WAP to compute the area of the triangle given the values of a, b and c (Take value of a,b,c input from the user).**

#include <stdio.h>

#include <math.h>

int main ()

{

  float a,b,c,s,area;

  printf("enter the first side:\n");

  scanf("%f",&a);

  printf("enter the second side.:\n");

  scanf("%f",&b);

  printf("enter the third side.:\n");

  scanf("%f",&c);

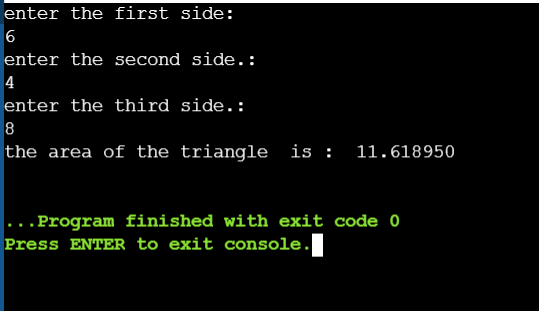
  s=(a+b+c)/2;

  area = sqrt(s\*(s-a)\*(s-b)\*(s-c));

  printf("the area of the triangle  is :  %f \n", area);

  return 0;

}



**CP ASSIGNMENT – 2**

**1. Write a C program to check whether a number is ODD or EVEN, using if else and switch.**

1a

#include<stdio.h>

int main()

{

    int num;

    printf("enter the value to check if it is odd or even :\n");

    scanf("%d",&num);

    if((num%2)==0)

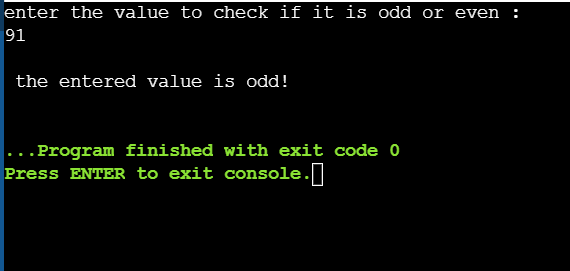
    printf("\n the entered number is even!\n");

    else

    printf("\n the entered value is odd!\n");

    return 0;

}



1b

#include<stdio.h>

int main()

{

    int num,t;

    printf("enter the value to check if it is odd or even :\n");

    scanf("%d",&num);

    (num%2)?(t=1):(t=0);

    switch(t)

    {

        case 1:printf("\n the entered number is odd!");

        break;

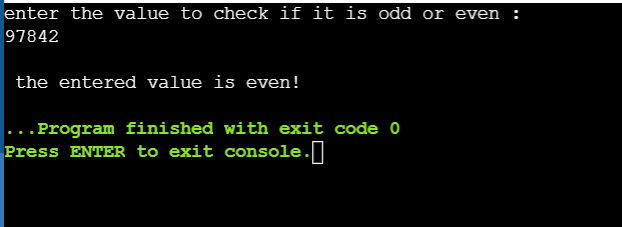
        case 0:printf("\n the entered value is even!");

        break;

    }

    return 0;

}



#include<stdio.h>

#include<ctype.h>

int main()

{

    int t;

    char c;

    printf("enter the character to check if it is vowel or consonant :\n");

    c=getchar();

    c=tolower(c);

    if(c=='a'||c=='e'||c=='i'||c=='o'||c=='u')

    t=1;

    else

    t=0;

    switch(t)

    {

        case 1:printf("\n the entered character is a vowel!");

        break;

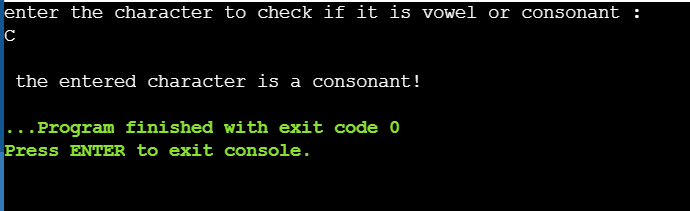
        case 0:printf("\n the entered character is a consonant!");

        break;

    }

    return 0;

}



**2. WAP to check whether a character is VOWEL or CONSONANT using switch statement.**

#include<stdio.h>

#include<ctype.h>

int main()

{

    int t;

    char c;

    printf("enter the character to check if it is vowel or consonant :\n");

    c=getchar();

    c=tolower(c);

    if(c=='a'||c=='e'||c=='i'||c=='o'||c=='u')

    t=1;

    else

    t=0;

    switch(t)

    {

        case 1:printf("\n the entered character is a vowel!");

        break;

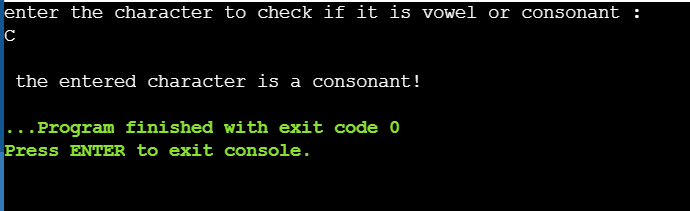
        case 0:printf("\n the entered character is a consonant!");

        break;

    }

    return 0;

}



**3. a) WAP to find the largest among three numbers using (i) nested if statements, (ii) else if statements, (iii) conditional operator?:**

3(i)

#include<stdio.h>

#include<ctype.h>

int main()

{

    int a,b,c;

    printf("enter theree numbers to find the largest of all :\n");

    scanf("%d%d%d",&a,&b,&c);

    if(a==b||b==c||a==c)

    printf("more than one variable have same value!");

    if(a>b)

    {

        if(a>c)

        printf("a is the largest number with value %d ",a);

        if(c>a)

        printf("c is the largest number with value %d ",c);

    }

    if(b>a)

    {

        if(b>c)

        printf("b is the largest number with value %d ",b);

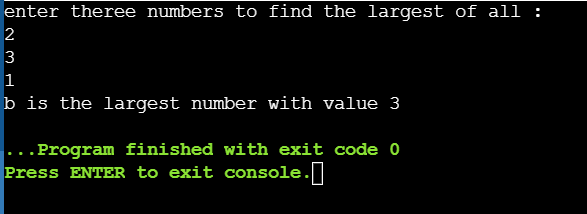
        if(c>b)

        printf("c is the largest number with value %d ",c);

    }

    return 0;

}



3(ii)

#include<stdio.h>

#include<ctype.h>

int main()

{

    int a,b,c;

    printf("enter theree numbers to find the largest of all :\n");

    scanf("%d%d%d",&a,&b,&c);

    if(a==b||b==c||a==c)

    printf("more than one variable have same value!");

    if(a>b)

    {

        if(a>c)

        printf("a is the largest number with value %d ",a);

        if(c>a)

        printf("c is the largest number with value %d ",c);

    }

    else if(b>c)

        printf("b is the largest number with value %d ",b);

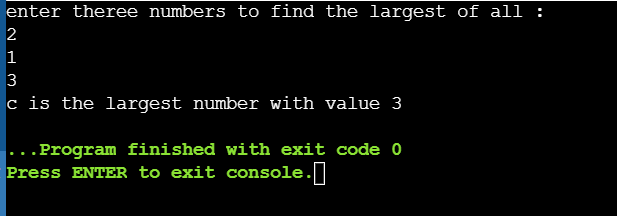
        if(c>b)

        printf("c is the largest number with value %d ",c);

    }

    return 0;

}



3(iii)

#include<stdio.h>

#include<ctype.h>

int main ()

{

  int a, b, c;

    printf ("enter theree numbers to find the largest of all :\n");

    scanf ("%d%d%d", &a, &b, &c);

    if (a == b || b == c || a == c)

    printf ("more than one variable have same value!");

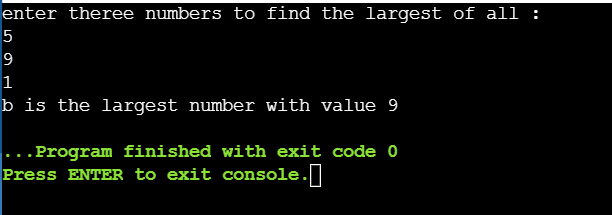
    (a > b && a > c) ? printf ("a is the largest number with value %d", a) :

    ((b > a && b > c) ? printf ("b is the largest number with value %d ", b) :

    printf ("c is the largest number with value %d ", c));

  return 0;

}



**b) WAP to check whether a character is an alphabet, digit. If it is an alphabet then check it is in upper case or lower case. If it is in lower case then check it is vowel or consonant. If it is a digit then check whether it is divisible by 2 and 5 or not.**

#include<stdio.h>

#include<ctype.h>

int main()

{

    char ch;

    printf("enter the character to check it's an digit or alphabat :\n");

    scanf("%c",&ch);

    if((ch>=65&&ch<=90)||(ch>=97&&ch<=122))

    {

        printf("entered character is a alphabet! \n now checking for upper or lower case....\n");

        if(ch>=65&&ch<=90)

            printf("the character is an upper case!\n");

        else

        {

            printf("entered character is an lower case!\n");

            printf("now checking for vowel or consonant....\n");

            ch=tolower(ch);

            if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u')

            printf("the character is a vowel!\n");

            else

            printf("entered character is a consonant!\n");

        };

    };

    if((ch>=48)&&(ch<=57))

    {

        printf("entered character is an digit!\n now checking for divisibality by 2 or 5....\n");

        if((ch%2)==0)

            printf("entered character is divisible by 2!\n");

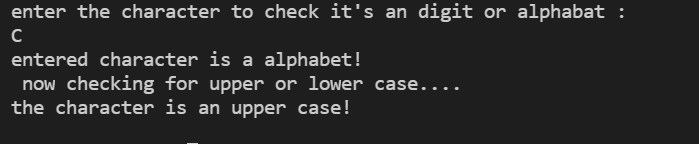
        if((ch%5)==0)

            printf("enetered character is divsible by 5!\n");

    }

    return 0;

}



**4. Write a program to print the month name corresponding to the digit enter from the user using switch statement. For example if user enter 1 then print “January” on 2 print “February”… and so on.**

#include<stdio.h>

void main()

{

    char num;

    printf("enter the number to print corresponding month : \n");

    scanf("%d",&num);

    switch(num)

    {

        case 1:printf("january\n");

        break;

        case 2:printf("february\n");

        break;

        case 3:printf("march\n");

        break;

        case 4:printf("april\n");

        break;

        case 5:printf("may\n");

        break;

        case 6:printf("june\n");

        break;

        case 7:printf("july\n");

        break;

        case 8:printf("august\n");

        break;

        case 9:printf("september\n");

        break;

        case 10:printf("october\n");

        break;

        case 11:printf("november\n");

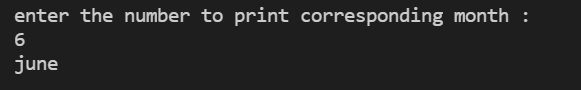
        break;

        case 12:printf("december\n");

        break;

    }

}



**5. WAP a read a three digits number from the keyboard and check whether it is a palindrome or not.**

**(Hint: If we reverse a number and the reverse number is same as the original number then it is palindrome. For example, the number 121 is a palindrome.)**

#include<stdio.h>

void main()

{

    char num[5];

    printf("enter three digit number to check for pelindrome :\n");

    gets(num);

    int i=0,j=2;

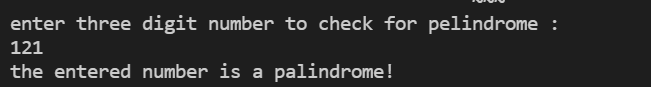
    if(num[i]==num[j])

    printf("the entered number is a palindrome!");

    else

    printf("the entered number is not a palindrome!");

}



**6. WAP to find the eligibility of admission of a student for a professional institute based on the**

**following criteria:**

**Marks in Mathematics >=65,**

**Marks in Physics >=55,**

**Marks in Chemistry >=60,**

**Total in all three subjects >=190 or Total in Mathematics and Chemistry >=130**

**(Hint: Input: marks obtained in Physics: 55, marks obtained in Chemistry: 61 marks obtained in Mathematics: 72, Output: The candidate is eligible for admission.)**

#include<stdio.h>

void main()

{

    int p,c,m;

    printf("enter the marks of students in the order of physics, chemistry and maths:\n");

    scanf("%d%d%d",&p,&c,&m);

    if(m>=65&&p>=55&&c>=60)

    {

        if((p+c+m)>=190||(m+c)>=130)

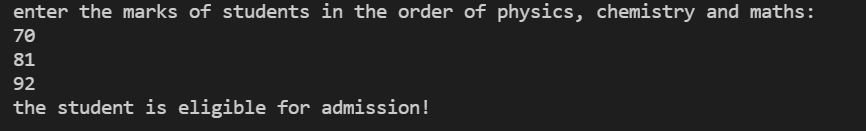
        printf("the student is eligible for admission!\n");

    }

    else

    printf("thre student is not eligible for admission!\n");

}



**7. Write a C program to input marks of five subjects and calculate the percentage. Also**

**Calculate Grade according to the following:**

**Percentage>= 90% A**

**Percentage>=80% B**

**Percentage>=70% C**

**Percentage>=60% D**

**Percentage>=40% E**

**Percentage<40% Fail**

#include<stdio.h>

void main()

{

    int s1,s2,s3,s4,s5;

    float p;

    printf("enter the marks of five subjects :\n");

    scanf("%d %d %d %d %d",&s1,&s2,&s3,&s4,&s5);

    p=((s1+s2+s3+s4+s5)/5);

    if(p>=90)

    printf("grade is A\n");

    else if(p>=80)

    printf("grade is B\n");

    else if(p>=70)

    printf("grade is C\n");

    else if(p>=60)

    printf("grade is D\n");

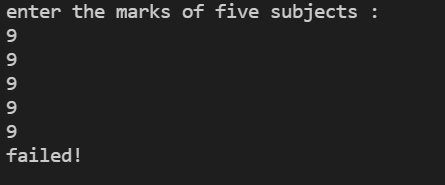
    else if(p>=40)

    printf("grade is E\n");

    else

    printf("failed!\n");

}



**Home Assignment:**

**1.Write a C program to check a given year is leap-year or not.**

#include<stdio.h>

void main()

{

    int year;

    printf("enter the year to check it for leap year :\n");

    scanf("%d",&year);

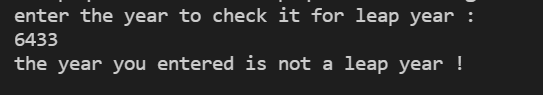
    if((year%4)==0)

    printf("the year you entered is a leap year! \n");

    else

    printf("the year you entered is not a leap year !\n");

}



**2. Write a C program to input the coefficients of a quadratic equation and find the roots of an Equation.**

#include<stdio.h>

#include<math.h>

void main()

{

    float a,b,c,r1,r2;

    printf("assuming the ax^2 + bx + c = 0 be the form of quadratic equation...\n");

    printf("enter the values of a , b and c to find the roots :\n");

    scanf("%f %f %f",&a,&b,&c);

    r1=(-b+sqrt(pow(b,2)-4\*a\*c))/(2\*a);

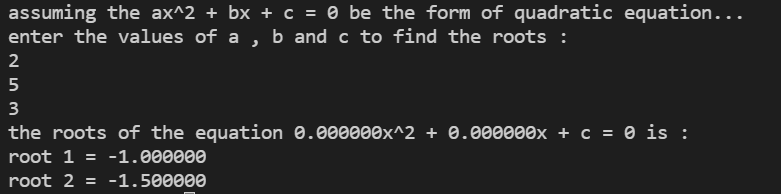
    r2=(-b-sqrt(pow(b,2)-4\*a\*c))/(2\*a);

    printf("the roots of the equation %fx^2 + %fx + c = 0 is :\n");

    printf("root 1 = %f\n",r1);

    printf("root 2 = %f",r2);

}



**3. WAP to do the following tasks on a triangle:**

**a) Input angles of a triangle and check whether triangle is valid or not. (Hint: Input all three angles of triangle in some variable say angle1, angle2 and angle3. Find sum of all three angles, store sum in some variable say sum = angle1 + angle2 + angle3. Check if (sum = = 180) then, triangle can be formed otherwise not. In addition, make sure angles are greater than 0 i.e., check condition for angles if (angle1!= 0 && angle2 != 0 && angle3 != 0).**

#include<stdio.h>

void main()

{

    float a1,a2,a3;

    printf("enter the angles of the triangle :\n");

    scanf("%f %f %f",&a1,&a2,&a3);

    if((a1!=0)&&(a2!=0)&&(a3!=0))

    {

        if((a1+a2+a3)==180)

        printf("the triangle formed by the angles is valid!\n");

        else

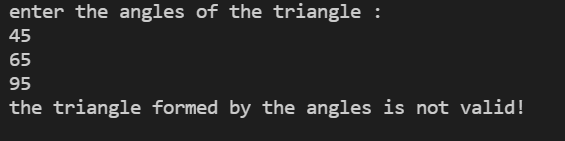
        printf("the triangle formed by the angles is not valid!\n");

    }

    else

    printf("the triangle formed by the angles is not valid!\n");

}



**b) Input all sides of a triangle and check whether triangle is valid or not. (Hint: A triangle is valid if sum of its two sides is greater than the third side. Means if a, b, c are three sides of a triangle. Then the triangle is valid if all three conditions are satisfied. a + b > c, a + c > b and b + c > a)**

#include<stdio.h>

void main()

{

    float a,b,c;

    printf("enter the sides of the triangle :\n");

    scanf("%f %f %f",&a,&b,&c);

    if(((a+b)>c)&&((a+c)>b)&&((b+c)>a))

    {

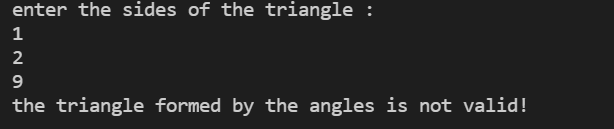
        printf("the triangle formed by the sides is valid!\n");

    }

    else

        printf("the triangle formed by the angles is not valid!\n");

}



**c) Whether the triangle is equilateral, isosceles or scalene triangle. (Hint: Input sides of a triangle from user. Store it in some variables say side1, side2 and side3. Check if (side1 = = side2 && side2 = = side3), then the triangle is equilateral. If it is not an equilateral triangle then it may be isosceles. Check if (side1 = = side2 || side1 = = side3 || side2 = = side3), then triangle is isosceles. If it is neither equilateral nor isosceles then it scalene triangle.**

#include<stdio.h>

void main()

{

    float a,b,c;

    printf("enter the sides of the triangle :\n");

    scanf("%f %f %f",&a,&b,&c);

    if(((a+b)>c)&&((a+c)>b)&&((b+c)>a))

    {

        printf("the triangle formed by the sides is valid!\n");

    }

    else

        printf("the triangle formed by the angles is not valid!\n");

    if((a==b)&&(a==c))

        printf("the triangle formed by the sides is an equilateral!\n");

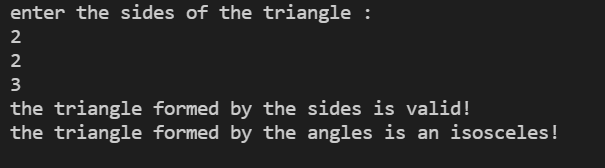
    else if((a==b)||(a==c)||b==c)

        printf("the triangle formed by the angles is an isosceles!\n");

    else

        printf("the triangle formed by the angles is an scalene!\n");

}



**CP ASSIGNMENT -3**

**Q1. WAP to accept a number and print all its digits in words using Switch and loop**

**statement.**

**(Hint: Input: 149, Output: One Four Nine)**

#include<stdio.h>

void main()

{

    char num[10];

    printf("enter the number to print in words :\n");

    gets(num);

    for(int n=0;num[n]!='\0';n++)

    {

        switch(num[n])

        {

            case '0':printf("zero\t");

            break;

            case '1':printf("one\t");

            break;

            case '2':printf("two\t");

            break;

            case '3':printf("tree\t");

            break;

            case '4':printf("four\t");

            break;

            case '5':printf("five\t");

            break;

            case '6':printf("six\t");

            break;

            case '7':printf("seven\t");

            break;

            case '8':printf("eight\t");

            break;

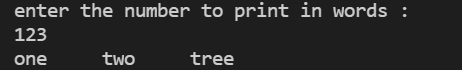
            case '9':printf("nine\t");

            break;

        };

    };

}



**Q2. WAP to print Fibonacci series up to ‘n’ terms where n is entered from user.**

#include<stdio.h>

void main()

{

    int n,a=0,b=1,temp=1;

    printf("enter the number upto which you want to display fibonacci series :\n");

    scanf("%d",&n);

    printf("the series is as followed ...\n");

    printf("the first two  constant numbers are :\n %d  %d  \n",a,b);

    printf("the rest of the series is :\n ");

    for(int i=0;i<=(n-3);i++)

    {

        printf("%d  ",a+b);

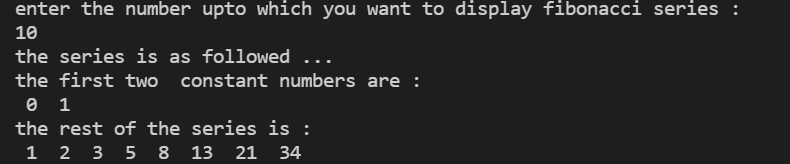
        temp=b;

        b=a+b;

        a=temp;

    }

}



**Q3 WAP to check whether the entered integer number is palindrome or not.**

#include<stdio.h>

void main()

{

    char num[10];

    int i=0,tag,j,t;

    printf("enter number to check for pelindrome :\n");

    gets(num);

    for(j=0;num[j]!='\0';j++);

    if(j%2==0)

        t=j/2;

    else

        t=(j+3)/2;

    while(j>=t)

    {

        if(num[i]==num[j-1])

        tag = 1;

        else

        {

            tag = 0;

            break;

        }

        j--;

        i++;

    }

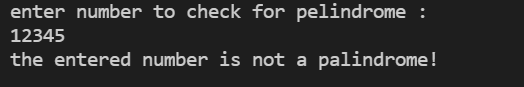
    if(tag==1)

    printf("the entered number is a palindrome!\n");

    else

    printf("the entered number is not a palindrome!\n");

}



**Q4 WAP to check whether a given number is prime or not. Also print all prime numbers**

**between 1 to n where n is entered from user.**

4a

#include <stdio.h>

int main()

{

    int i,j,num,tag;

    printf("enter the num:\n");

    scanf("%d",&num);

    for(i=2;i\*i<=num;i++)

    {

        if(num%i==0)

        {

            tag=1;

            break;

        }

        else

        {

            tag=0;

        }

    };

    if(tag==0)

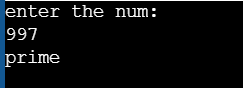
    printf("prime");

    else

    printf("its not prime");

    return 0;

}



4b

#include<stdio.h>

void main()

{

    int num,i,j,tag;

    printf("enter the number:\n");

    scanf("%d",&num);

    for(i=2;i<=num;i++)

    {

        tag=1;

        for(j=2;j<=i/2;j++)

        {

            if(i%j==0)

            {

                tag=0;

                break;

            }

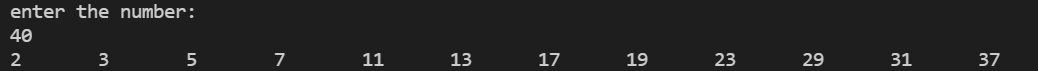
        }

        if(tag==1)

        printf("%d\t",i);

    }

}



**Q5. a) WAP to print the following pyramids using numbers**

**1**

**2 3 2**

**3 4 5 4 3**

**4 5 6 7 6 5 4**

**5 6 7 8 9 8 7 6 5**

#include<stdio.h>

void main()

{

    int n,countspace=0,aftermid=0;

    printf("enter the row number to make pyrimid :\n");

    scanf("%d",&n);

    for(int i=0;i<n;i++)                    // this loop is for rows

    {

        for(int j=1;j<(n-i);j++)        // this loop is for spaces

        {

            printf(" ");

            countspace++;

        }

        for(int j=0;j<((2\*i)+1);j++)        // this loop is for enrting numbers or stars

        {

            if(countspace<=(n-1))

            {

                printf("%d",(j+i+1));       // for printing number till middle

                countspace++;

            }

            else

            {

                aftermid++;

                printf("%d",(i+1+j-(2\*aftermid)));

            }

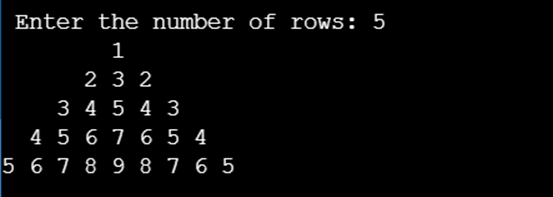
        }

        printf("\n");

        countspace=aftermid=0;

    }

}



**b) WAP to print the following pascal’s triangle using numbers.**

**1**

**1 1**

**1 2 1**

**1 3 3 1**

**1 4 6 4 1**

**1 5 10 10 5 1**

#include <stdio.h>

int factorial(int);

int ncr(int,int);

void main()

{

  int num,i,j;

  printf("Enter the number of rows you want to print in pascal triangle :\n");

  scanf("%d",&num);

  for (i=0;i<num;i++)

  {

    //loop for spaces

    for (j=0;j<=(num-i-2);j++)

      printf(" ");

    //loop for numbers

    for (j=0;j<=i;j++)

      printf("%d ",ncr(i,j));

    printf("\n");

  }

}

int factorial(int n)

{

  int i;

  int fact=1;

  for (i= 1;i<=n;i++)

    fact\*=i;

  return fact;

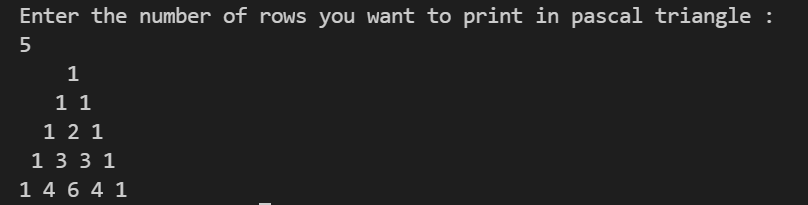
}

int ncr(int i,int j)

{

    return factorial(i)/(factorial(j)\*factorial(i-j));

}



**c) WAP to print the following Floyd’s triangle using numbers.**

**1**

**2 3**

**4 5 6**

**7 8 9 10**

#include<stdio.h>

void main()

{

    int num,i,j,k=1;

    printf("enter the number of rows :\n");

    scanf("%d",&num);

    for(i=1;i<=num;i++)

    {

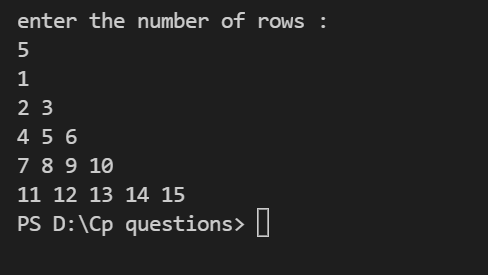
        for(j=1;j<=i;j++,k++)

        printf("%d ",k);

        printf("\n");

    }

}



**Practice Assignment:**

**1. WAP that reads 10 different numbers and compute the sum of all odd and even numbers separately. Use for loop, while loop and do-while loop each separately for this program.**

**2. WAP to find the binary equivalent of an integer number entered from user also count the number of 1’s and number of 0’s in the resultant binary number.**

**3. Write programs to print following patterns-**

**\***

**\* \* \***

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

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

#include<stdio.h>

void main()

{

    int n;

    printf("enter the number of rows to make pyramids :");

    scanf("%d",&n);

    for(int i=0;i<n;i++)

    {

        for(int j=0;j<(n-i);j++)

        printf(" ");

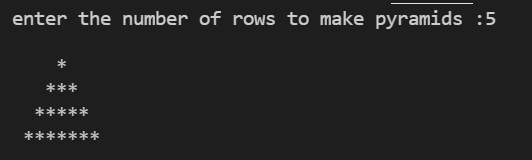
        for(int j=0;j<((2\*i)-1);j++)

        printf("\*");

        printf("\n");

    }

}



**0**

**1 0 1**

**2 1 0 1 2**

**3 2 1 0 1 2 3**

#include <stdio.h>

int main()

{ int l;

   for(int i=1;i<=5;i++)

   {

   for(int j=5;j>i;j--)

   printf(" "); l=i-1;

   for( int k=1;k<=i;k++,l--)

   printf("%d ",l);

   l++;

   for(int k=1;k<=i-1;k++)

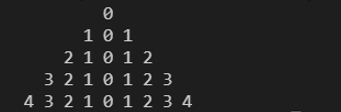
   printf("%d ",++l);

   printf("\n");

}

return 0;

}

****

**1**

**2 2**

**3 3 3**

**4 4 4 4**

**5 5 5 5 5**

#include<stdio.h>

void main()

{

    int num,i,j,k;

    printf("enter the number of rows :\n");

    scanf("%d",&num);

    for(i=1;i<=num;i++)

    {

        //loop for spaces

        for(k=num-1;k>=i;k--)

        printf(" ");

        //loop for numbers in rows

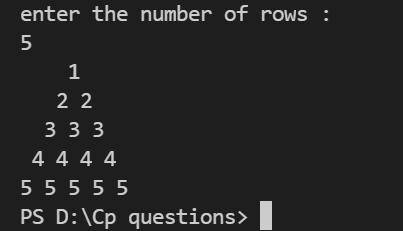
        for(j=1;j<=i;j++)

        printf("%d ",i);

        printf("\n");

    }

}



**1**

**232**

**34543**

**4567654**

#include<stdio.h>

void main()

{

    int n,countspace=0,aftermid=0;

    printf("enter the row number to make pyrimid :\n");

    scanf("%d",&n);

    for(int i=0;i<n;i++)                    // this loop is for rows

    {

        for(int j=1;j<(n-i);j++)        // this loop is for spaces

        {

            printf(" ");

            countspace++;

        }

        for(int j=0;j<((2\*i)+1);j++)        // this loop is for enrting numbers or stars

        {

            if(countspace<=(n-1))

            {

                printf("%d",(j+i+1));       // for printing number till middle

                countspace++;

            }

            else

            {

                aftermid++;

                printf("%d",(i+1+j-(2\*aftermid)));

            }

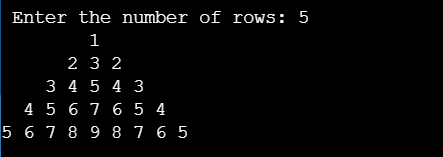
        }

        printf("\n");

        countspace=aftermid=0;

    }

}



**\***

**\* \***

**\* \* \***

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

**\* \* \***

**\* \***

**\***

#include<stdio.h>

void main()

{

    int num,i,j,k;

    printf("enter the number of columns :\n");

    scanf("%d",&num);

    for(i=1;i<=num;i++)

    {

        for(k=1;k<=i;k++)

        printf("\*");

        printf("\n");

    }

    for(i=2;i<=num;i++)

    {

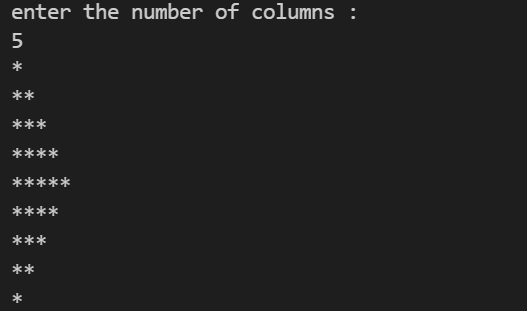
        for(j=num;j>=i;j--)

        printf("\*");

        printf("\n");

    }

}



**NIT-J CSED**

**Computer Programming Lab**

**Lab 4: Functions**

**Q1. Write a C program to input radius of circle from user and find diameter, circumference**

**and area of the given circle using function.**

#include<stdio.h>

void diameter(float n)

{

    printf("the diameter of circle is %f \n",n\*2);

}

void circumference(float n)

{

    float pi=3.14159265359;

    printf("the circumference of the circle is %f \n",2\*pi\*n);

}

void area(float n)

{

    float pi=3.14159265359;

    printf("the area of the circumference of the circle is %f \n",pi\*n\*n);

}

void main()

{

    float r;

    printf("enter the radius of the circle : \n");

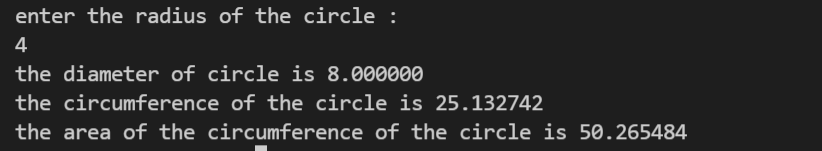
    scanf("%f",&r);

    diameter(r);

    circumference(r);

    area(r);

}

****

**Q2. WAP to find the sum of the following series from 1 to n using user-defined function.**

**1!/1 + 2!/2 + 3!/3 + ….. + n!/n. For example, 1!/1+2!/2+3!/3+4!/4+5!/5 = 1+1+2+6+24=34.**

**[Hint: use a separate function to compute the factorial of the corresponding number.]**

#include<stdio.h>

int factorial(int);

int series(int);

void main()

{

    int n,s;

    printf("enter the number up to which you want to print sum of series :\n");

    scanf("%d",&n);

    s=series(n);

    printf("%d",s);

}

int factorial(int a)

{

    if((a-1)==0){return 1;}

    else

    a=a\*factorial(a-1);

    return (a);

}

int series(int i)

{

    int ssum;

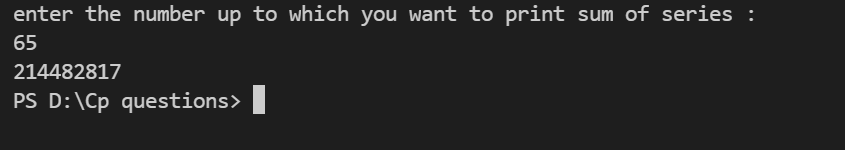
    if((i-1)==0)return 1;

    else

    ssum=(factorial(i)/i)+series(i-1);

    return ssum;

}

****

**Q3. WAP that accepts ‘n’ numbers from keyboard and count how many of those numbers are**

**even numbers using user-defined function. The main() function reads all ‘n’ numbers one by**

**one and at the same time sending them by reference to a called function named ‘int count**

**even()’. The job of the ‘count even’ function is to check the number received by it as even or**

**odd. For every even number received, it should increase a count variable and finally send the**

**value of the count variable to the main() function where it gets printed.**

#include<stdio.h>

int counteven(char n[]);

void main()

{

    char a[50],count;

    printf("enter the numbers one by one , to check for total even numbers :\n");

    gets(a);

    count=counteven(a);

    printf("the number of evens are : %d \n",count);

}

int counteven(char n[])

{

    int i=0,c=0;

    while(n[i]!='\0')

    {

        if(n[i]%2==0)

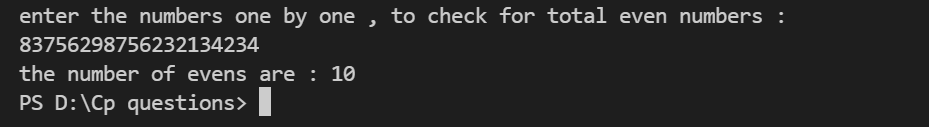
        c++;

        i++;

    }

return c;

}

****

**Q4. WAP to check whether a number is prime, Armstrong, perfect number or not using**

**functions. Kindly check the previous assignments (Assignments I to VI) to find the definitions**

**of prime, Armstrong and perfect number.**

**The function prototype is as follows:**

**/\* Function declarations \*/**

**int Prime(int num);**

**int Armstrong(int num);**

**int Perfect(int num);**

**Example: Input**

**Input any number: 11**

**Output**

**11 is prime number**

**11 is not a Armstrong number**

**11 is not a perfect number**

/\*

#include<stdio.h>

#include<math.h>

int prime(int);

int armstrong(int);

int perfect(int);

void main()

{

    int n,t1,t2,t3;

    printf("enter the number to check whether it is prime, Armstrong, perfect number or not :\n");

    scanf("%d",&n);

    t1=prime(n);

    t2=armstrong(n);

    t3=perfect(n);

}

int prime(int num)

{

    int tag=0;

    for(int i=2;i\*i<=num;i++)

    {

        if(num%i==0)

        {

            tag=1;

            break;

        }

    }

    if(tag==0)

    printf("the entered number is prime!\n");

    else

    printf("the entered number is not prime!\n");

    return 0;

}

int armstrong(int num)

{

    int a,b=num,n=0,i=0;

    while(b!=0)

    {

        a=b-(b/10)\*10;

        b=b/10;

        i++;

    }

    a=0;b=num;

    for(int j=0;j<i;j++)

    {

        a=b-(b/10)\*10;

        b=b/10;

        n+=pow(a,i);

    }

    if(n==num)

    printf("the entered number is a armstrong!\n");

    else

    printf("the entered mumber is not a armstrong!\n");

    return 0;

}

int perfect(int num)

{

    int a=0,b,c;

    for(int i=1;i<num;i++)

    {

        if(num%i==0)

        a+=i;

    }

    if(a==num)

    printf("the entered number is a perfect!\n");

    else

    printf("the entered number is not a perfect!\n");

    return 0;

}

\*/

#include<stdio.h>

#include<math.h>

int prime(int);

int armstrong(int);

int perfect(int);

void main()

{

    int n,t1,t2,t3;

    printf("enter the number to check whether it is prime, Armstrong, perfect number or not :\n");

    scanf("%d",&n);

    t1=prime(n);

    t2=armstrong(n);

    t3=perfect(n);

}

int prime(int num)

{

    int tag=0;

    for(int i=2;i\*i<=num;i++)

    {

        if(num%i==0)

        {

            tag=1;

            break;

        }

    }

    if(tag==0)

    printf("the entered number is prime!\n");

    else

    printf("the entered number is not prime!\n");

    return 0;

}

int armstrong(int num)

{

    int a[10],b=num,n=0,i=0;

    while(b!=0)

    {

        a[i]=b-(b/10)\*10;

        b=b/10;

        i++;

    }

    for(int j=0;j<i;j++)

    {

        n+=pow(a[j],i);

    }

    if(n==num)

    printf("the entered number is a armstrong!\n");

    else

    printf("the entered mumber is not a armstrong!\n");

    return 0;

}

int perfect(int num)

{

    int a=0,b,c;

    for(int i=1;i<num;i++)

    {

        if(num%i==0)

        a+=i;

    }

    if(a==num)

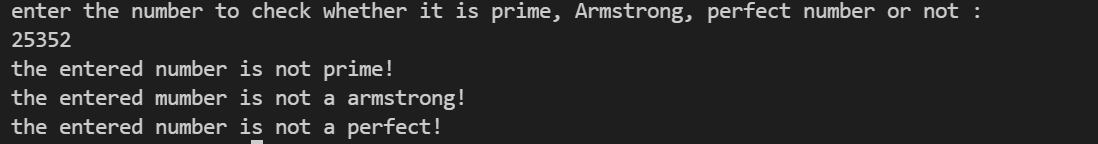
    printf("the entered number is a perfect!\n");

    else

    printf("the entered number is not a perfect!\n");

    return 0;

}

****

**Assignment**

**1. WAP using user-defined function to find all strong numbers between 1 to n. Use a**

**separate function to compute the factorial of the corresponding number.**

**[Hint: A number in which the sum of factorial of individual digits is equal to the number is**

**called strong number.**

**For example, 145 is a strong number because 145= 1! + 4! + 5!**

**Example: Input**

**Input lower limit: 1**

**Input upper limit: 1000**

**Output**

**Strong numbers between 1 and 1000: 1, 2, 145]**

#include<stdio.h>

#include<stdio.h>

void strong(int \*,int,int);

void numtoarr(int);

int factorial(int);

int arr[10];

void main()

{

    int num,n,i=0;

    printf("\nenter the number to check for strong number :\n");

    scanf("%d",&num);

    n=num;

    for(int c=0;c<=num;c++)

    {

        numtoarr(c);

    }

}

int factorial(int n)

{

    int fact=1,i;

    for(i=1;i<=n;i++)

    fact\*=i;

    return fact;

}

void strong(int a[],int i,int n)

{

    int sum=0,t;

    for(t=0;t<i;t++)

    {

        sum=sum+factorial(a[t]);

    }

    if(sum==n)

    printf("the strong number is : %d\n",n);

}

void numtoarr(int c)

{

    int n=c,i=0;

  do

    {

        arr[i]=n-((n/10)\*10);

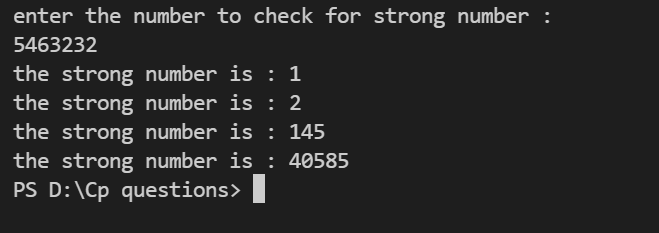
        n=n/10;

        i++;

    }while(n!=0);

    strong(arr,i,c);

}

****

**2. Write a function to compute the distance between two points and use it to develop**

**another function that will compute the area of the triangle whose vertices are A(x1, y1),**

**B(x2, y2), and C(x3, y3). Use these functions to develop a function which returns a value 1**

**if the point (x, y) lines inside the triangle ABC, otherwise a value 0.**

#include<stdio.h>

#include<math.h>

#include<stdlib.h>

float fdistance(float x1,float y1,float x2,float y2)

{

    float dis;

    dis=sqrt(pow((x1-x2),2)+pow((y1-y2),2));

    return dis;

};

float farea(float x1,float x2,float x3,float y1,float y2,float y3)

{

    float a,b,c,s,area;

    a=fdistance(x1,y1,x2,y2);

    b=fdistance(x1,y1,x3,y3);

    c=fdistance(x2,y2,x3,y3);

    s=(a+b+c)/2;

    area=sqrt(s\*(s-a)\*(s-b)\*(s-c));

    return area;

}

int flocation(float x1,float x2,float x3,float y1,float y2,float y3,float p,float q)

{

    float a,b,c,i,j,k;

    a=fdistance(x1,y1,x2,y2);

    b=fdistance(x1,y1,x3,y3);

    c=fdistance(x2,y2,x3,y3);

    i=fdistance(p,q,x1,y1);

    j=fdistance(p,q,x2,y2);

    k=fdistance(p,q,x3,y3);

    if(( (j<=a)&&(j<=b)&&(j<=c)) && ((i<=a)&&(i<=b)&&(i<=c)) && ((k<=a)&&(k<=b)&&(k<=c)) )

    return 1;

    else

    return 0;

}

void main()

{

    float x1,x2,x3,y1,y2,y3,dis,a,b,AREA;

    int choice,n;

    do

    {

        printf("choose what you want to do...\n");

        printf("  1   - for finding the distance between two points\n");

        printf("  2   - for finding the area of triangle from its vertices\n");

        printf("  3   - for finding the location of pint wrt triangle\n");

        printf("  0   - to exit\n");

        scanf("%d",&choice);

        switch(choice)

        {

            case 1: printf("you have choosen to find the distance between two points...\n");

                    printf("enter the co-ordinates (x1,y1) of the first point :\t");

                    scanf("%f %f",&x1,&y1);

                    printf("enter the co-ordinates (x2,y2) of the second point :\t");

                    scanf("%f %f",&x2,&y2);

                    dis=fdistance(x1,y1,x2,y2);

                    printf("the distance between two points is %f\n\n",dis);

            break;

            case 2: printf("you have choosen to find the area of triangle from vertices ...\n");

                    printf("enter the co-ordinates (x1,y1) of the first point :\t");

                    scanf("%f %f",&x1,&y1);

                    printf("enter the co-ordinates (x2,y2) of the second point :\t");

                    scanf("%f %f",&x2,&y2);

                    printf("enter the co-ordinates (x2,y2) of the third point :\t");

                    scanf("%f %f",&x3,&y3);

                    AREA=farea(x1,x2,x3,y1,y2,y3);

                    printf("the area of the triangle is %f\n\n",AREA);

            break;

            case 3: printf("you have choosen to find the position of point wrt triangle...\n");

                    printf("enter the co-ordinates (x1,y1) of the first point of triangle :\t");

                    scanf("%f %f",&x1,&y1);

                    printf("enter the co-ordinates (x2,y2) of the second point of triangle :\t");

                    scanf("%f %f",&x2,&y2);

                    printf("enter the co-ordinates (x2,y2) of the third point of triangle :\t");

                    scanf("%f %f",&x3,&y3);

                    printf("enter the co-ordinates (a,b) of the point to locate :\t");

                    scanf("%f %f",&x3,&y3);

                    n=flocation(x1,x2,x3,y1,y2,y3,a,b);

                    if(n==1)

                    printf("the pint lies inside the triangle !\n\n");

                    else

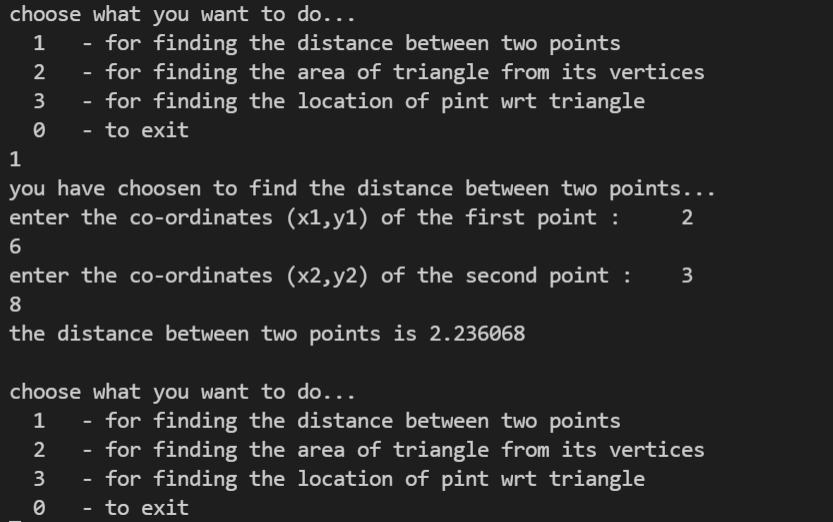
                    printf("the point lies outside the the triangle !\n\n");

        }

    }

    while(choice!=0);

}

****

**3. Write a menu driven program which has following options:**

**1. Find the p**

**th power of a number n using recursive function**

**2. Find the binary equivalent of a number n and count all 1’s using recursive function.**

**3. Find the reverse of the number using recursive function.**

**4. Exit**

**Once a menu item is selected the appropriate action should be taken and once this action is**

**finished, the menu should reappear. Unless the user selects the Exit option the program**

**should continue to work.**

#include<stdio.h>

float multi=1;

float power(float num,int p)

{

    if(p==1)

    return num;

    else

    {

        multi=num\*power(num,p-1);

        return multi;

    }

};

int binary(int num)

{

int count =0,i,j;

printf("the binary equivalent of the number is...\n");

do

{

    i=num%2;

    num=num/2;

    printf("%d",i);

    if(i==1)

    count++;

} while (num!=0);

return count;

};

void reverse(int num)

{

    int i;

    printf("the number in reverse order is ....\n");

    do

    {

        i=num-((num/10)\*10);

        num=num/10;

        printf("%d",i);

    } while (num!=0);

    printf("\n\n");

};

void main()

{

    int choice,n,p,count;

    float num,pow;

    do

    {

        printf("choose what you want to do...\n");

        printf("  1   - for Find the pth power of a number n\n");

        printf("  2   - for Find the binary equivalent of a number n and count all 1's \n");

        printf("  3   - for Find the reverse of the number  \n");

        printf("  0   - to exit \n");

        scanf("%d",&choice);

        switch(choice)

        {

            case 1: printf("you have choosen to Find the pth power of a number n...\n");

                    printf("enter the number and the power :\t");

                    scanf("%f %d",&num,&p);

                    pow=power(num,p);

                    printf("the %dth power of a number %f is %f  \n\n",p,num,pow);

            break;

            case 2: printf("you have choosen to Find the binary equivalent of a number n and count all 1's ...\n");

                    printf("enter the number :\t");

                    scanf("%d",&n);

                    count=binary(n);

                    printf("\nthe number of 1's in the binary equalent is = %d\n\n",count);

            break;

            case 3: printf("you have choosen to for Find the reverse of the number ...\n");

                    printf("enter the number :\t");

                    scanf("%d",&n);

                    reverse(n);

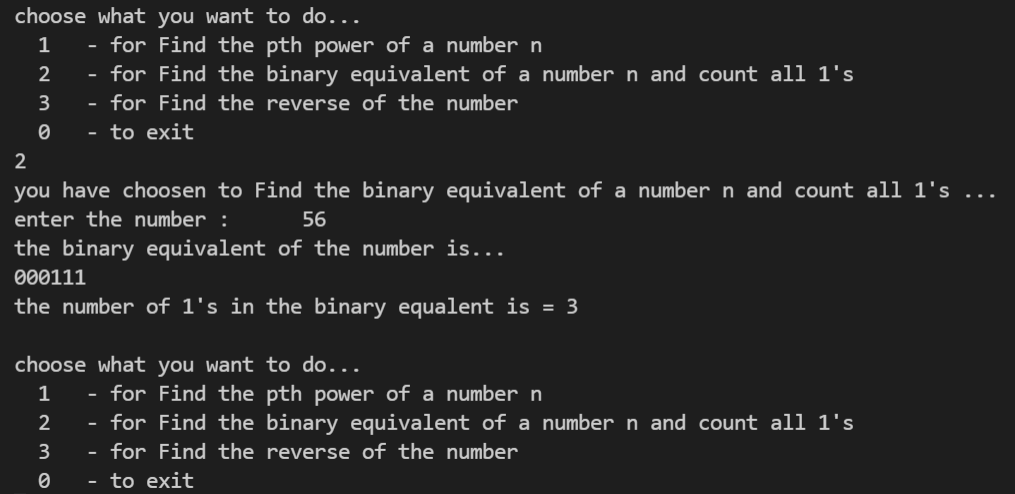
            break;

        }

    }

    while(choice!=0);

};

****

**NIT-J CSED**

**Computer Programming Lab**

**Lab 5: Recursion and Pointers**

**1. Write a program to calculate the factorial of a given number using recursion.**

#include<stdio.h>

int factorial(int num)

{

    int fact;

    if(fact == 1)

    return fact;

    else

    {

        fact=num\*factorial(num-1);

        return fact;

    }

}

void main()

{

    int n,fac;

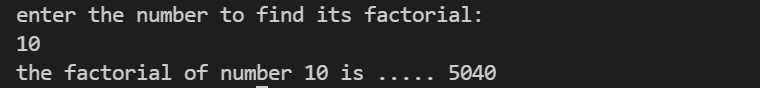
    printf("enter the number to find its factorial:\n");

    scanf("%d",&n);

    fac=factorial(n);

    printf("the factorial of number %d is ..... %d \n",n,fac);

}

****

**2. Write a C program to swap two numbers using pointers and functions. How to swap two**

**numbers using call by value and call by reference method.**

#include<stdio.h>

void swap\_by\_reference(int \*x,int \*y)

{

    int temp;

    temp=\*x;

    \*x=\*y;

    \*y=temp;

    printf("the swaped numbers ( in swap\_by\_reference ) is...\n");

    printf("x = %d \n",\*x);

    printf("y = %d \n",\*y);

}

void swap\_by\_value(int x,int y)

{

    int temp;

    temp=x;

    x=y;

    y=temp;

    printf("the swaped numbers ( in swap\_by\_reference ) is...\n");

    printf("x = %d \n",x);

    printf("y = %d \n",y);

}

void main()

{

    int a,b,choice;

    restart:

    printf("enter the numbers to swap ...\n");

    printf("a = ");

    scanf("%d",&a);

    printf("b = ");

    scanf("%d",&b);

    printf("enter 1 for call by value method ");

    printf("or enter 2 for call ny reference method\n");

    scanf("%d",&choice);

    if(choice==1)

    {

        swap\_by\_value(a,b);

        printf("the original values are... \n");

        printf("a = %d \n b = %d \n",a,b);

    }

    else if (choice==2)

    {

        swap\_by\_reference(&a,&b);

        printf("the original values are... \n");

        printf("a = %d \nb = %d \n",a,b);

    }

    else

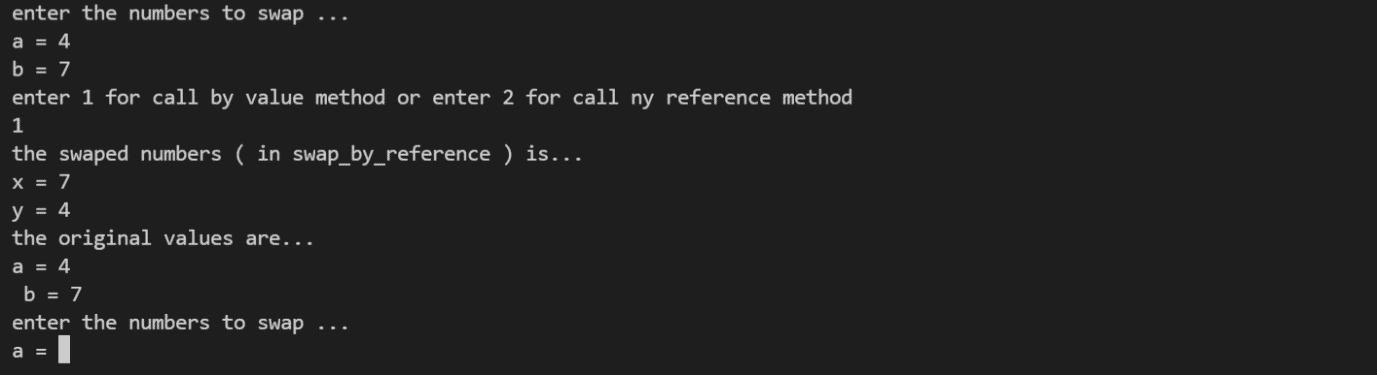
    {

        printf("entered choice is wrong !\n");

    }

    goto restart;

}

****

**3. A 5-digit positive integer is entered through the keyboard, write a function to calculate sum**

**of digits of the 5-digit number: (1) Without using recursion (2) Using recursion**

#include<stdio.h>

int recursion(int num)

{

    int sum=0,n;

    if (num==0)

    return num;

    else

    {

        n=num-((num/10)\*10);

        sum=n+recursion(num/10);

        return sum;

    }

}

void without\_recursion(int num)

{

    int sum=0,n;

    do

    {

        n=num-((num/10)\*10);

        num=num/10;

        sum+=n;

    } while (num!=0);

    printf("the sum of the digits of number is : %d \n\n",sum);

}

void main()

{

    int num,choice,r\_sum=0;

    printf("enter 5 digit number :\n");

    scanf("%d",&num);

    printf("enter 1 for adding the digits by Using recursion \n");

    printf("enter 2 for adding digits by without Using recursion\n");

    scanf("%d",&choice);

    if(choice==1)

    {

        r\_sum=recursion(num);

        printf("the sum of digits of entered number is = %d \n\n",r\_sum);

    }

    else if (choice==2)

    {

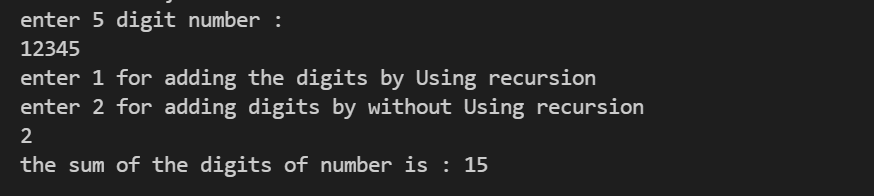
        without\_recursion(num);

    }

    else

    printf("entered choice is wrong !\n\n");

}

****

**4. A positive integer is entered through the keyboard, write a program to obtain the prime**

**factors of the number. Modify the function suitably to obtain the prime factors recursively.**

/\*

#include<stdio.h>

void prime\_factors(int num)

{

    int n;

    for(n=2;n<=num;n++)

    {

        if(num%n==0)

        {

            printf("%d, ",n);

            prime\_factors(num/n);

            break;

        }

    }

}

void main()

{

    int num;

    printf("Enter an number to find its prime factors : ");

    scanf("%d", &num);

    printf("The Prime Factors of %d are: ",num);

    prime\_factors(num);

}

\*/

#include<stdio.h>

void recursion(int num)

{

    int n;

    for(n=2;n<=num;n++)

    {

        if(num%n==0)

        {

            printf("%d\t",n);

            recursion(num/n);

            break;

        }

    }

}

void without\_recursion(int num)

{

    int i=1,j;

    printf("the prime factors are...\n");

    for(i=2;i<=num;i++)

    {

        int tag;

        if(num%i==0)

        {

            tag=0;

            for(j=2;j\*j<=i;j++)

            {

                if(i%j==0)

                {

                    tag=1;

                    break;

                }

            }

            if(tag==0)

            printf("%d\t",i);

        }

    };

}

int prime;

void main()

{

    int num,choice,r;

    printf("enter number to find its prime factors:\n");

    scanf("%d",&num);

    printf("enter 1 for finding prime factors by Using recursion \n");

    printf("enter 2 for finding prime fectors by without Using recursion\n");

    scanf("%d",&choice);

    if(choice==1)

    {

        recursion(num);

        printf("\n\n");

    }

    else if (choice==2)

    {

        without\_recursion(num);

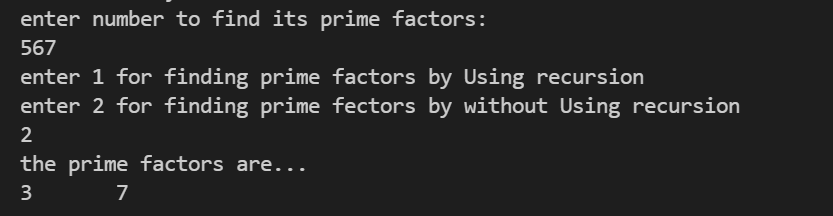
        printf("\n\n");

    }

    else

    printf("entered choice is wrong !\n\n");

}

****

**5. Write a recursive function to generate nth Fibonacci term in C programming. How to**

**generate nth Fibonacci term in C programming using recursion.**

/\*#include<stdio.h>

int Fibonacci(int num)

{

    if (num == 0)

    {

        return 0;

    }

    else if (num == 1)

    {

        return 1;

    }

    else

    {

        return(Fibonacci(num - 1) + Fibonacci(num - 2));

    }

}

void main()

{

    int num;

    int result;

    printf("Enter the nth number in fibonacci series: ");

    scanf("%d", &num);

    if (num < 0)

    {

        printf("Fibonacci of negative number is not possible.\n");

    }

    else

    {

        result = Fibonacci(num);

        printf("The %d number in Fibonacci series is %d\n", num, result);

    }

}

\*/

#include<stdio.h>

int input;

int i=0,j=1,num=0;

void Fibonacci(int n)

{

    int sum=0,temp;

        sum=i+j;

        i=j;

        j=sum;

        num++;

        if(num==n+1)

        printf("the corrosponding fibonacci term to the value entered = %d\n",sum);

        else

        Fibonacci(n);

 }

void main()

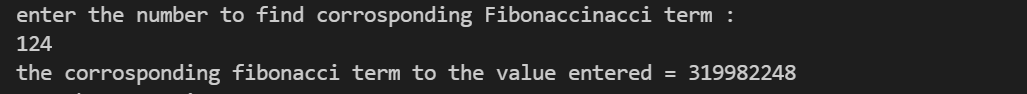
{

    printf("enter the number to find corrosponding Fibonaccinacci term :\n");

    scanf("%d",&input);

    Fibonacci(input);

}

****

**NIT-J CSED**

**Computer Programming Lab**

**Lab 6: Array, Pointer and Functions with Array**

**1. Consider a class having ‘n’ students. You need to read all the student marks for a particular**

**subject and store only marks in a one dimensional array named as student\_array and then**

**have to do the following things one-by-one using recursion. Marks should be between 0 to**

**100.**

**i) display all the marks on the screen and find the total and average marks scored in**

**that subject.**

**ii) find the minimum and maximum among all the marks.**

**iii) counting the number of marks scored by the students which are below the average**

**marks, total number of marks which are similar, and frequency of each mark stored in**

**the array.**

**[Hint: The main() function need to call three sub-functions by passing the array and the**

**number of**

**students using the following prototype.**

**void diplay(int \*n) // call by reference**

**min\_max(int n) // call by value**

**count(int \*n)// call by reference**

**In order to extend the accessibility of student\_array by all the programs of this assignment,**

**declared student\_array as ‘extern int student\_array ‘in a header file. Subsequently, include**

**this header file in all the programs. This is how we can share a common variable or function**

**in different files. A sample example is shown at the end which will help you to use the extern**

**variable.]**

**2. Extend the program by separating the odd and even marks into two separate arrays**

**named as odd\_marks and even\_marks respectively and finally copied these two arrays into a**

**third arrays named as odd\_even\_marks so that all the odds marks are stored before all the**

**marks which are even.**

**3. Extend the above program such that if any new student has taken late admission then**

**your code should able to accommodate such new student mark in the student\_array. To do**

**this, your code should ask the position (i.e., index) where the new mark will be entered in**

**the student\_array. If any student is leaving the class then your program should handle this**

**by deleting the corresponding mark from the student\_array.**

**[Hint: Call two functions insert\_mark (int position) and delete\_mark (int mark) from**

**main() function.**

**Assignment**

**1. Extend the program such that it can return the nth highest mark from the student\_array.**

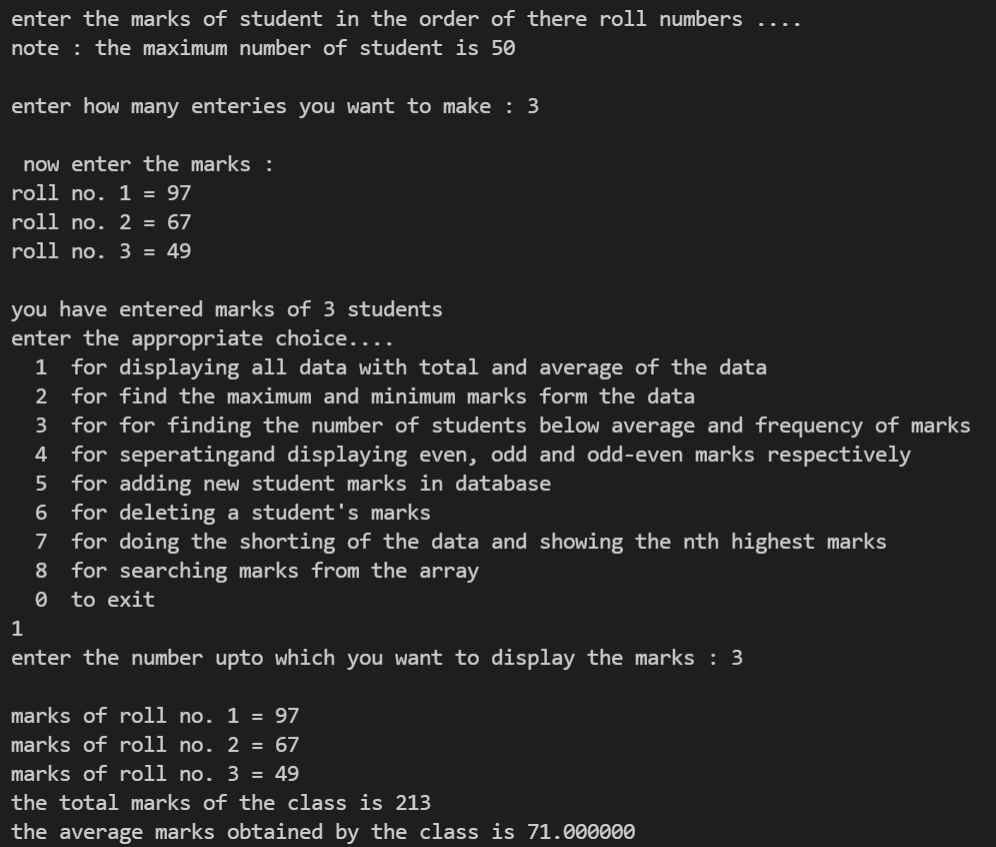
**To do this first sort all the marks of student\_array.**

**2. Extend the program using function which will take the mark of a student as the input and**

**search in the student\_array. It will display appropriate message ‘Found’ or ‘Not Found’. In**

**the case of ‘Found’, return the index of the student\_array where the mark was found.**

1. #include<stdio.h>
2. #include"student.h"
3. int max\_num;
4. float avg=0;
5. extern int student\_array[];
6. extern int input\_marks(int \*);
7. void diplay(int \*array,int n)
8. {
9. int sum=0;
11. if(n>max\_num)
12. {   printf("\nthe maximum number of student exceeded ....\n");
13. printf("showing marks of all students ....\n");
14. n=max\_num;
15. }
16. for(int i=0;i<n;i++)
17. {
18. printf("\nmarks of roll no. %d = %d",i+1,array[i]);
19. sum+=array[i];
20. }
21. avg=(sum/n);
22. printf("\nthe total marks of the class is %d ",sum);
23. printf("\nthe average marks obtained by the class is %f",avg);
24. }
25. void min\_max(int array[],int n)
26. {
27. int min=array[0],max=array[0],max\_n,min\_n;
28. for(int i=0;i<n;i++)
29. {
30. if(min>=array[i])
31. {
32. min=array[i];
33. min\_n=i;
34. }
35. };
36. for(int i=0;i<n;i++)
37. {
38. if(max<=array[i])
39. {
40. max=array[i];
41. max\_n=i;
42. }
43. };
44. printf("the maximum marks is obtained by roll no. %d = %d",max\_n,max);
45. printf("\nthe minimum marks is obtained by roll no. %d = %d\n",min\_n,min);
46. }
47. void count(int \*array,int n)
48. {
49. int count=0,element;
50. for(int i=0;i<n;i++)
51. {
52. if(array[i]<avg)
53. count++;
54. };
55. printf("the number of students below average is = %d\n",count);
56. for(int i=0;i<max\_num;i++)
57. {
58. count=0;
59. element=array[i];
60. for(int j=0;j<max\_num;j++)
61. {
62. if(element==array[j])
63. count++;
64. }
65. printf("\nthe number of students got %d marks = %d ",element,count);
66. }
68. }
69. void seperate(int \*array)
70. {
71. int even[50],j=0,k=0,t,odd[50],odd\_even[50];
72. for(int i=0;i<max\_num;i++)
73. {
74. if(array[i]%2==0)
75. even[j++]=array[i];
76. else
77. odd[k++]=array[i];
78. }
79. printf("the marks in 'odd' array are...\n");
80. for(int i=0;i<k;i++)
81. printf("%d\t",odd[i]);
83. printf("\nthe marks in 'even' array are...\n");
84. for(int i=0;i<j;i++)
85. printf("%d\t",even[i]);
87. for(int i=0;i<k;i++)
88. odd\_even[i]=odd[i];
89. for(int i=k,t=0;i<max\_num;i++,t++)
90. odd\_even[i]=even[t];
91. printf("\nthe marks in 'odd\_even' array are...\n");
92. for(int i=0;i<max\_num;i++)
93. printf("%d\t",odd\_even[i]);
95. }
96. void insert\_marks(int \*position)
97. {
98. int temp,value;
99. printf("enter the marks of new student....\n");
100. scanf("%d",&value);
101. for(int i=max\_num;i>=\*position;i--)
102. {
103. student\_array[i]=student\_array[i-1];
104. }
105. student\_array[\*position-1]=value;
106. max\_num++;
107. printf("\nthe new marks of students are...\n");
108. for(int i=0;i<max\_num;i++)
109. printf("\nmarks of roll no. %d = %d",i+1,student\_array[i]);
110. }
111. void delete\_marks(int \*position)
112. {
113. int temp;
115. for(int i=\*position-1;i<max\_num-1;i++)
116. {
118. student\_array[i]=student\_array[i+1];
119. }
120. max\_num--;
121. printf("\nthe new marks of students are...\n");
122. for(int i=0;i<max\_num;i++)
123. printf("\nmarks of roll no. %d = %d",i+1,student\_array[i]);
124. }
125. void shorting(int \*array)
126. {
127. int small,pos,temp;
128. printf("\n the shorted list is ....\n");
129. for(int i=0;i<max\_num-1;i++)
130. {
131. small=array[i];
132. pos=i;
133. for(int j=i+1;j<max\_num;j++)
134. {
135. if(array[j]<small)
136. {
137. small=array[j];
138. pos=j;
139. }
140. }
141. temp=array[i];
142. array[i]=array[pos];
143. array[pos]=temp;
144. }
145. for(int i=0;i<max\_num;i++)
146. printf("%d\t",array[i]);
147. int n,t;
148. printf("\nenter the position to dispaly its marks : ");
149. scanf("%d",&n);
150. t=max\_num-n;
151. printf("\nthe marks at position %d is : %d\n",n, array[t]);
152. }
153. void searching(int \*array)
154. {
155. int value;
156. printf("\nenter the marks to search in the list : ");
157. scanf("%d",&value);
158. for(int i=0;i<max\_num;i++)
159. {
160. if(array[i]==value)
161. {
162. printf("the marks was found at postion %d",i+1);
163. };
164. };
165. }
166. void main()
167. {
168. int choice,num;
169. printf("firstly create a database \n");
170. max\_num=input\_marks(student\_array);
171. do
172. {
173. printf("\nenter the appropriate choice....\n");
174. printf("  1  for displaying all data with total and average of the data\n");
175. printf("  2  for find the maximum and minimum marks form the data\n");
176. printf("  3  for for finding the number of students below average and frequency of marks\n");
177. printf("  4  for seperatingand displaying even, odd and odd-even marks respectively\n");
178. printf("  5  for adding new student marks in database\n");
179. printf("  6  for deleting a student's marks\n");
180. printf("  7  for doing the shorting of the data and showing the nth highest marks\n");
181. printf("  8  for searching marks from the array\n");
182. printf("  0  to exit\n");
184. scanf("%d",&choice);
186. switch(choice)
187. {
188. case 1 :    printf("enter the number upto which you want to display the marks : ");
189. scanf("%d",&num);
190. diplay(student\_array,num);
191. break;
192. case 2 :    min\_max(student\_array,max\_num);
193. break;
194. case 3 :    count(student\_array,max\_num);
195. break;
196. case 4 :    seperate(student\_array);
197. break;
198. case 5 :    printf("\nenter the role no. at which you want to add marks of new student : ");
199. scanf("%d",&num);
200. insert\_marks(&num);
201. break;
202. case 6 :    printf("\nenter the roll no. of student you want to delete : ");
203. scanf("%d",&num);
204. delete\_marks(&num);
206. break;
207. case 7 :    shorting(student\_array);
208. break;
209. case 8 :    searching(student\_array);
210. break;
211. case 0 :
212. break;
214. }
215. }while(choice!=0);
216. }



**NIT-J CSED**

**Computer Programming Lab**

**Lab 7 : Multi-dimensional Arrays, pointer and Functions**

1. WAP to add, subtract & perform scalar multiplication on 2D matrices.

1. WAP to find the multiplication of matrices A and B into matrix C where the initialization of matrices A and B should be done by a function initMat(), printing of matrices should be done by printMat() and multiplication should be done by mulMat(). You need to pass the matrices into functions according to need.

Answer for q1,q2,hw q 1,hw q 2.

#include<stdio.h>

void initMat(int n,int m,int a[][10])

{

    for(int i=0;i<n;i++)

    {

        for(int j=0;j<m;j++)

        {

            printf("enter the value at [%d][%d] : ",i,j);

            scanf("%d",&a[i][j]);

        }

    }

}

void printMat(int n,int m,int a[][10])

{

    for(int i=0;i<n;i++)

    {

        for(int j=0;j<m;j++)

        {

            printf("%d  ",a[i][j]);

        }

        printf("\n");

    }

}

void mulMat(int n,int m,int m1,int a[][10],int b[][10])

{

    static int multi\_m[10][10];

    for(int i=0;i<n;i++)

    for(int j=0;j<m;j++)

    for(int t=0;t<m1;t++)

    {

        multi\_m[i][j]=0;

        multi\_m[i][j]=a[i][j]\*b[i][j];

    }

    printMat(n,m,multi\_m);

}

void addMat(int n,int m,int a[][10],int b[][10])

{

    static int sum\_m[10][10];

    for(int i=0;i<n;i++)

    for(int j=0;j<m;j++)

    sum\_m[i][j]=a[i][j]+b[i][j];

    printMat(n,m,sum\_m);

}

void subMat(int n,int m,int a[][10],int b[][10])

{

    static int sub\_m[10][10];

    for(int i=0;i<n;i++)

    for(int j=0;j<m;j++)

    sub\_m[i][j]=a[i][j]-b[i][j];

    printMat(n,m,sub\_m);

}

void transMat(int n,int m,int a[][10])

{

    int trans\_m[10][10];

    for(int i=0;i<n;i++)

    for(int j=0;j<m;j++)

    trans\_m[j][i]=a[i][j];

    printMat(n,m,trans\_m);

}

void symMat(int n,int m,int a[][10])

{

    int tag=0;

    for(int i=0;i<n;i++)

    for(int j=0;j<m;j++)

    {

        if(a[i][j]!=a[j][i])

        {

            tag=1;

            break;

        }

    if(tag==1)

    break;

    }

    if(tag==1)

    printf("\nthe matrix is not a symmetric !\n");

    else

    printf("\nthe matrix is symmetric !\n");

}

int main()

{

    int a[10][10],b[10][10],r1,r2,c1,c2,choice,num=0;

    printf("enter the choice wisely.....\n");

    printf("  1  for doing scalar multiplication on matrix\n");

    printf("  2  for doing addition on matrixes\n");

    printf("  3  for doing subtraction on matrixes\n");

    printf("  4  for mutiplying two matrixes\n");

    printf("  5  for finding transpose of a matric\n");

    printf("  6  for checking if matrix is symmetric or not\n");

    scanf("%d",&choice);

    switch(choice)

    {

        case 1 :    printf("enter the no. of rows and columns of first matrix :\n");

                    scanf("%d %d",&r1,&c1);

                    printf("\nenter the scalar you want to multiply :\n");

                    scanf("%d",&num);

                    initMat(r1,c1,a);

                    for(int i=0;i<r1;i++)

                    {

                        for(int j=0;j<c1;j++)

                        {

                            a[i][j] = a[i][j] \* num;

                        }

                    }

                    for(int i=0;i<r1;i++)

                    {

                        for(int j=0;j<c1;j++)

                        {

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

                        }

                        printf("\n");

                    }

                    printMat(r1,c1,a);

        break;

        case 2 :    start1:

                    printf("enter the no. of rows and columns of first matrix :\n");

                    scanf("%d%d",&r1,&c1);

                    printf("\nenter the no. of rows and columns of second matrix :\n");

                    scanf("%d%d",&r2,&c2);

                    if(r1==r2&&c1==c2)

                    printf("the matrixes can be added !\n");

                    else

                    {

                        printf("the matrixes can't be added !\n re-enter the values \n");

                        goto start1;

                    }

                    printf("\nenter the values in first matrix\n");

                    initMat(r1,c1,a);

                    printf("\nenter the values in second matrix\n");

                    initMat(r2,c2,b);

                    addMat(r1,c1,a,b);

        break;

        case 3 :    start2:

                    printf("enter the no. of rows and columns of first matrix :\n");

                    scanf("%d%d",&r1,&c1);

                    printf("\nenter the no. of rows and columns of second matrix :\n");

                    scanf("%d%d",&r2,&c2);

                    if(r1==r2&&c1==c2)

                    printf("the matrixes can be substracted !\n");

                    else

                    {

                        printf("the matrixes can't be substracted !\n re-enter the values \n");

                        goto start2;

                    }

                    printf("\nenter the values in first matrix\n");

                    initMat(r1,c1,a);

                    printf("\nenter the values in second matrix\n");

                    initMat(r2,c2,b);

                    subMat(r1,c1,a,b);

        break;

        case 4 :    start3:

                    printf("enter the no. of rows and columns of first matrix :\n");

                    scanf("%d%d",&r1,&c1);

                    printf("\nenter the no. of rows and columns of second matrix :\n");

                    scanf("%d%d",&r2,&c2);

                    if(r1==c2)

                    printf("the matrixes can be multiplied !\n");

                    else

                    {

                        printf("the matrixes can't be multiplied !\n re-enter the values \n");

                        goto start3;

                    }

                    printf("\nenter the values in first matrix\n");

                    initMat(r1,c1,a);

                    printf("\nenter the values in second matrix\n");

                    initMat(r2,c2,b);

                    mulMat(r1,c2,c1,a,b);

        break;

        case 5 :    printf("enter the no. of rows and columns of first matrix :\n");

                    scanf("%d%d",&r1,&c1);

                    initMat(r1,c1,a);

                    transMat(r1,c1,a);

        break;

        case 6 :    printf("enter the no. of rows and columns of first matrix :\n");

                    scanf("%d%d",&r1,&c1);

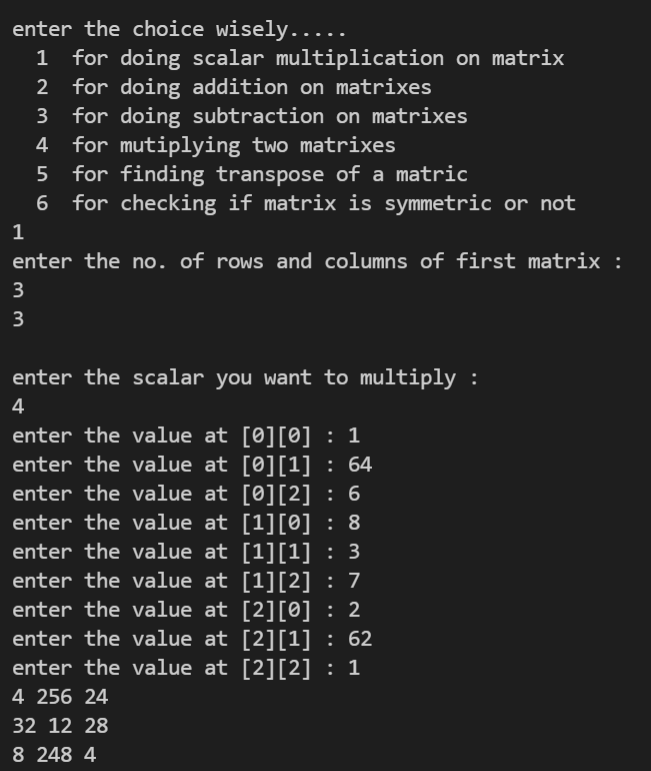
                    initMat(r1,c1,a);

                    symMat(r1,c1,a);

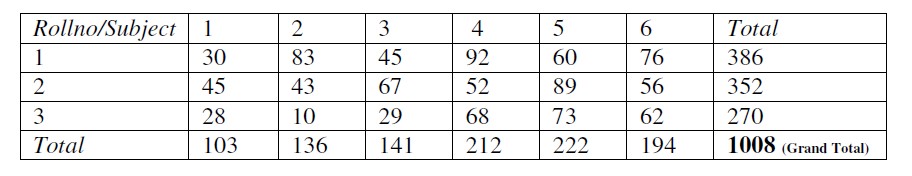
        break;

    }

}



1. Assume that you now planned to stored subject wise student marks in the same *student\_array* as discussed in Assignment-VIII by modifying it to a two dimensional array as shown below for three students and six subject marks.



Assuming ‘n’ students and ‘m’ subjects, design a menu driven program using function which will do the followings where all the students’ rollno and subject name are integers as shown in the above table.

i) Read those student marks and display them on the screen along with their rollno using pointer. ii) Display the subject-wise and student-wise total marks obtained and the grand total.

iii) Display the student details who obtained the highest mark.

#include<stdio.h>

#include"student2d.h"

extern int student\_array[50][50];

extern void input(int [][50],int,int);

void display(int a[][50],int n,int m)

{

    printf("the marks of students in different subjects are...\n");

    printf("rollno./subject\t");

    for(int i=0;i<m;i++)

    printf("%d\t",i+1);

    printf("\n");

    for(int i=0;i<n;i++)

    {

        printf("%d\t\t",i+1);

        for(int j=0;j<m;j++)

        {

            printf("%d\t",a[i][j]);

        }

        printf("\n");

    }

}

void sub\_stud\_marks(int a[][50],int n,int m)

{

    int sum=0;

    printf("\nthe subject-wise total marks are...\n");

    for(int i=0;i<n;i++)

    {

        printf("the total marks in subject %d is : ",i+1);

        for(int j=0;j<m;j++)

        {

            sum+=a[j][i];

        }

        printf("%d\n",sum);

        sum=0;

    }

    printf("\n\nthe student-wise total marks are....\n");

    for(int i=0;i<n;i++)

    {

        printf("the total marks of roll no. %d is : ",i+1);

        for(int j=0;j<m;j++)

        {

            sum+=a[i][j];

        }

        printf("%d\n",sum);

        sum=0;

    }

    printf("\n\nthe grand total marks obtained by the class are....   ");

    for(int i=0;i<n;i++)

    {

        for(int j=0;j<n;j++)

        {

            sum+=a[i][j];

        }

    }

    printf("%d\n",sum);

}

void highest\_mark(int a[][50],int n,int m)

{

    printf("\nthe student that gained the highest marks in the class is ....\n");

    int stud\_sum[50],high,r\_num;

    for(int i=0;i<n;i++)

    {

        stud\_sum[i]=0;

        for(int j=0;j<m;j++)

        {

            stud\_sum[i]+=a[i][j];

        }

    }

    high=stud\_sum[0];

    r\_num=0;

    for(int i=0;i<n;i++)

    {

        if(high<stud\_sum[i])

        {

            r\_num=i;

            high=stud\_sum[i];

        }

    }

    printf("the highest marks is gained by roll no. : %d \n",r\_num+1);

    printf("the marks in different subjects are...\n");

    for(int i=0;i<m;i++)

    {

        printf("\tsubject %d : %d\n",i+1,a[r\_num][i]);

    }

}

void main()

{

    int choice,n,m;

    printf("enter the no. of students you want to assign marks....\n");

    scanf("%d",&n);

    printf("enter the no of subjects you wants to assign marks....\n");

    scanf("%d",&m);

    input(student\_array,n,m);

    start:

    printf("enter the choice wisely.....\n");

    printf("  1  for Reading student marks and display them on the screen along with their rollno\n");

    printf("  2  for Displaying the subject-wise and student-wise total marks obtained and the grand total\n");

    printf("  3  for Displaying the student details who obtained the highest mark\n");

    scanf("%d",&choice);

    switch(choice)

    {

        case 1 :    display(student\_array,n,m);

                    // goto start;

        break;

        case 2 :    sub\_stud\_marks(student\_array,n,m);

                    // goto start;

        break;

        case 3 :    highest\_mark(student\_array,n,m);

                    goto start;

        break;

    }

}

**Assignment**

1. WAP to perform following operation on matrix (3\*3) using function and pointer

1. transpose of matrix

2.check weather symmetric or not.

1. Write a function which takes to sorted arrays as parameters and merge then into an another shorted array. Write a program to implement above function.

#include<stdio.h>

int\* shorting\_2\_array(int n,int m,int a[10],int b[10],int c[20])

{

    int p=0,q=0;

    for(int i=0;i<n+m;i++)

    {

        if(a[p]<b[q])

        {

            c[i]=a[p];

            p++;

        }

        else

        {

            c[i]=b[q];

            q++;

        }

    }

    return c;

}

void main()

{

    int n,m,a[10],b[10],c[20],\*ptr;

    printf("enter the number of elements in shorted manner inthe first array : \n");

    scanf("%d",&n);

    printf("enter the element in the first array.....\n");

    for(int i=0;i<n;i++)

    {

        printf("element %d : ",i);

        scanf("%d",&a[i]);

    }

    printf("\nenter the number of elements in shorted manner in the second array : \n");

    scanf("%d",&m);

    printf("enter the element in the second array.....\n");

    for(int i=0;i<m;i++)

    {

        printf("element %d : ",i);

        scanf("%d",&b[i]);

    }

    ptr=shorting\_2\_array(n,m,a,b,c);

    printf("\nthe new array is....\n");

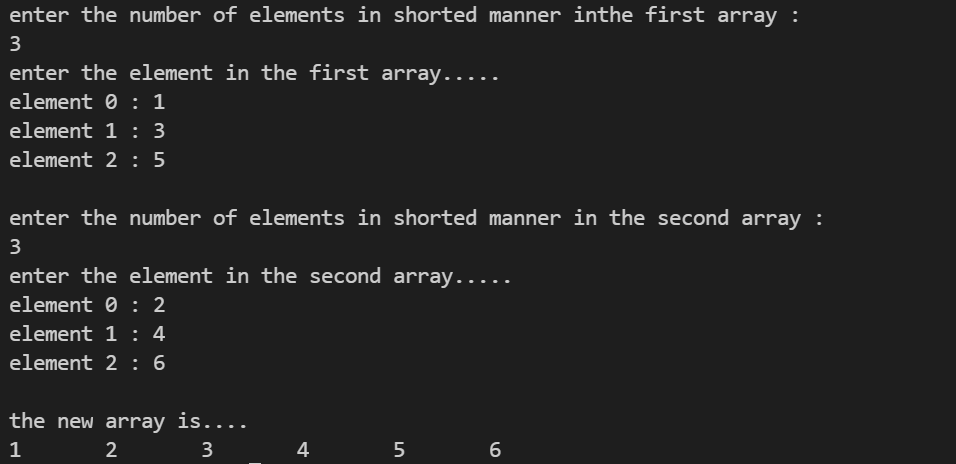
    for(int i=0;i<n+m;i++)

    {

        printf("%d\t",ptr[i]);

    }

}



**Lab 8 : Stings, Pre-processor directives and Macros**

**1. Write a program to demonstrate the use of following functions- strlen(), strlwr(),**

**strupr(), strcpy(), strncpy(), strcmp(), strcmpi(), strchr(), strrchr().**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int main()

{

    char str[50],str1[50],str2[50],str3[50],s,\*ret;

    int \*n,num,i,j,k,choice;

    start:

    printf("enter the choice wisely...\n");

    printf(" 1 to find the string length\n");

    printf(" 2 to convert the characters to lower case\n");

    printf(" 3 to convert the characters to upper case\n");

    printf(" 4 to copy and replace the sring with some other string\n");

    printf(" 5 to comapre two strings\n");

    printf(" 6 to compare two strings upto first n characters\n");

    printf(" 7 to compare two strings without case sensitivety\n");

    printf(" 8 to search the string for the first occerance of character\n");

    printf(" 9 to search for the last occurrence of the character in string\n");

    scanf("%d",&choice);

    switch(choice)

    {

        case 1 :

                    printf("\nenter the string : \n");

                    getchar();

                    gets(str);

                    num=strlen(str);

                    printf("the length of the entered string is : %d\n",num);

        break;

        case 2 :    printf("enter the string : \n");

                    getchar();

                    gets(str);

                    strlwr(str);

                    printf("the string in lower cases is : \n");

                    puts(str);

        break;

        case 3 :    printf("enter the string : \n");

                    getchar();

                    gets(str);

                    printf("the string in upper case is : \n");

                    strupr(str);

                    puts(str);

        break;

        case 4 :    printf("enter the firat string : \n");

                    getchar();

                    gets(str1);

                    printf("enter the second string :\n");

                    gets(str2);

                    printf("the first string is : %s\nthe second string is : %s\n",str1,str2);

                    strcpy(str1,str2);

                    printf("the first string after copying : %s\nthe second string after copying : %s\n",str1,str2);

        break;

        case 5 :    printf("enter the first string : \n");

                    getchar();

                    gets(str1);

                    printf("enter the second string :\n");

                    gets(str2);

                    printf("enter the number upto which you want to copy string : ");

                    scanf("%d",&num);

                    printf("the first string is : %s\nthe second string is : %s\n",str1,str2);

                    strncmp(str1,str2,num);

                    printf("the first string after copying : %s\nthe second string after copying : %s\n",str1,str2);

        break;

        case 6 :    printf("enter the first string : \n");

                    getchar();

                    gets(str1);

                    printf("enter the second string to compare :\n");

                    gets(str2);

                    num=strcmp(str1,str2);

                    if(num==0)

                    printf("\nthe strings are identical !\n");

                    else

                    printf("\nthe strings are not identical !\n");

        break;

        case 7 :    printf("enter the first string : \n");

                    getchar();

                    gets(str1);

                    printf("enter the second string :\n");

                    gets(str2);

                    num=strcmpi(str,str3);

                    if(num==0)

                    printf("\nthe strings are equal !\n");

                    else

                    printf("\nthe strings are not equal !\n");

        break;

        case 8 :    printf("enter the string : \n");

                    getchar();

                    gets(str);

                    printf("enter the character to search : ");

                    scanf("%c",&s);

                    getchar();

                    ret=strchr(str,s);

                    if(ret==NULL)

                    printf("\nthe entered character is not in the string !\n");

                    else

                    printf("\nthe string start from the charcter %c is  : %s\n",s,ret);

        break;

        case 9 :    printf("enter the string : \n");

                    getchar();

                    gets(str);

                    printf("enter the character to search : ");

                    scanf("%c",&s);

                    ret=strchr(str,s);

                    if(ret!=NULL)

                    printf("\nthe string starting from character %c is: %s\n",s,ret);

                    else

                    printf("\nthe entered character is not in the string !\n");

        break;

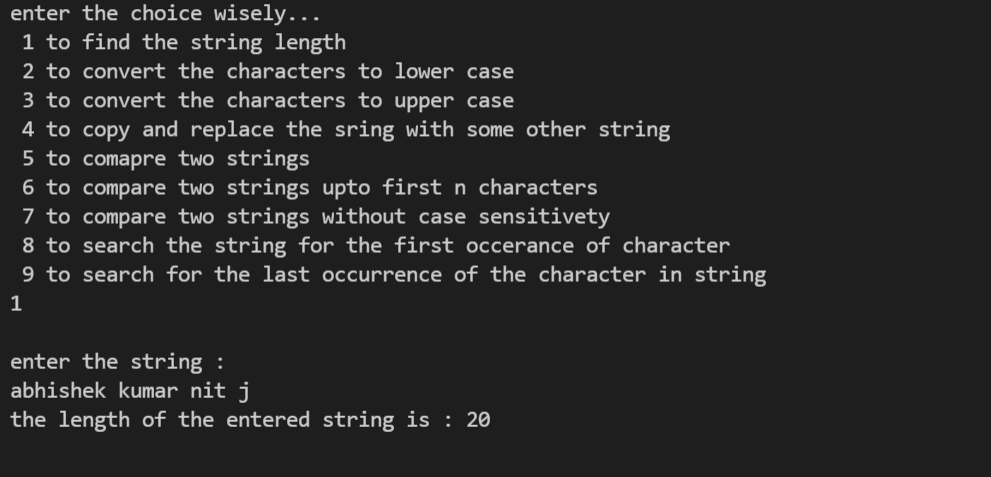
        case 0 :    return 0;

    }

    printf("\n\n");

    goto start;

}

****

**2. You are asked to store all the students name those who have enrolled in the class. You designed a two**

**dimensional array (std\_name\_array[ ][ ]) where the first dimension stores the number of the students**

**name and the second dimension stores the maximum length of each name. Do the following:**

**i) Read those students name and display them on the screen using pointer.**

**ii) Search a particular student name is present in the std\_name\_array[ ][ ] or not using binary**

**search.**

**iii) Append the word ‘NIT-J’ at the end of each name.**

**iv) List out the name(s) which has maximum characters.**

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

char std\_name\_array[50][20];

void input(char (\*name)[20],int n)

{

    for(int i=0;i<n;i++)

    {

        printf("enter the name of roll no. %d :",i+1);

        gets(name[i]);

    }

}

void display(char (\*name)[20],int n)

{

    for(int i=0;i<n;i++)

    {

        printf("the name of roll no. %d is : ",i+1);

        printf("%s\n",\*(name+i));

    }

}

void search(char name[][20],int n,char s[20])

{

    int tag=0;

    for(int i=0;i<n;i++)

    {

        if(tag==strcmp(name[i],s))

        {

            printf("the entered name is, of roll no. : %d\n",i+1);

            tag++;

        }

    }

    if(tag!=0)

        {

           printf("name was not found in database\n");

        }

}

void append(char name[][20],int n)

{

    for(int i=0;i<n;i++)

    {

        strcat(name[i]," NIT-J");

    }

}

void enlist(char name[][20],int n)

{

    int len[50];

    for(int i=0;i<n;i++)

    {

        len[i]=strlen(name[i]);

    }

    int max\_l=len[0],num=0;

    for(int i=0;i<n;i++)

    {

        if(max\_l<len[i])

        {

            max\_l=len[i];

            num=i;

        }

    }

    printf("the name of student(s) with longest name is...\n");

    for(int i=0;i<n;i++)

    {

        if(len[i]==max\_l)

        {

            printf("roll no. = %d\tname = %s\n",i+1,name[i]);

        }

    }

}

int main()

{

    int n,num,choice;

    char c[20];

    printf("enter the numbers of students of which you want to add name : ");

    scanf("%d",&n);

    getchar();

    printf("enter the names of students roll no. wise....\n");

    input(std\_name\_array,n);

    printf("\nenter the choice wisely...\n");

    printf(" 1 for displaying the content of database\n");

    printf(" 2 for searching student name in database\n");

    printf(" 3 for appending NIT-J at the end of the data base\n");

    printf(" 4 for listing the name(s) having the highest length\n");

    scanf("%d",&choice);

    switch(choice)

    {

        case 1 :    display(std\_name\_array,n);

        break;

        case 2 :    printf("enter the name to search : ");

                    fflush(stdin);

                    gets(c);

                    search(std\_name\_array,n,c);

        break;

        case 3 :    append(std\_name\_array,n);

                    printf("\nthe new database is .....\n");

                    display(std\_name\_array,n);

        break;

        case 4 :    enlist(std\_name\_array,n);

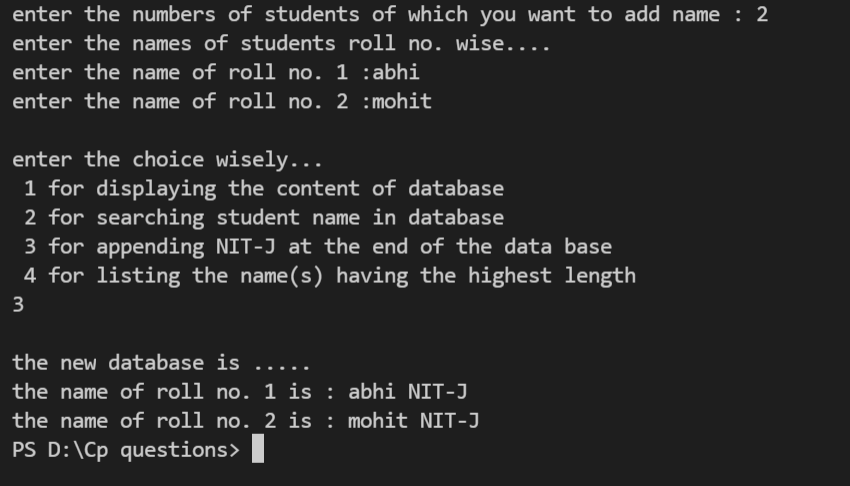
        break;

        case 0 :    return 0;

        break;

    }

}

****

**1. Write a C program to find total number of alphabets, digits or special character in a string.**

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

void main()

{

    char s[50];

    int c=0,n=0,t=0,len;

    printf("enter the string to know more about its character sets : ");

    gets(s);

    len=strlen(s);

    for(int i=0;i<len;i++)

    {

        if(s[i]<=57&&s[i]>=48)

        n++;

        else if ((s[i]<=90&&s[i]>=65)||(s[i]<=122&&s[i]>=97))

        {

            c++;

        }

        else

        t++;

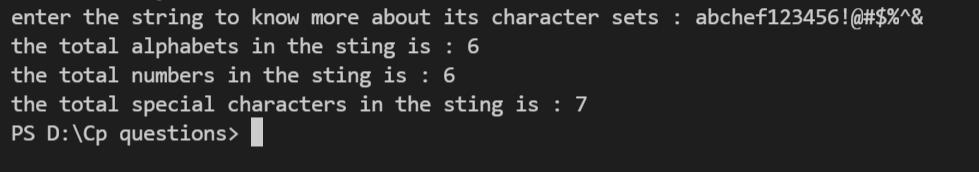
    }

        printf("the total alphabets in the sting is : %d\n",c);

        printf("the total numbers in the sting is : %d\n",n);

        printf("the total special characters in the sting is : %d\n",t);

}

****

**2. Calculate the area of circle by using function call macros.**

#include<stdio.h>

#define area(r) 3.14\*r\*r

void main()

{

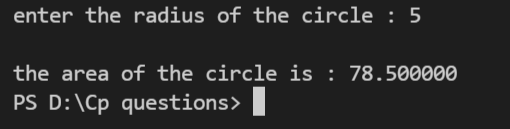
    float r;

    printf("enter the radius of the circle : ");

    scanf("%f",&r);

    printf("\nthe area of the circle is : %f",area(r));

}

****

**3. Write a C program using function to check whether a string is palindrome or not.**

#include<stdio.h>

#include<string.h>

void main()

{

    char s[50];

    int n,q,t=0;

    printf("enter the string : ");

    gets(s);

    n=strlen(s);

    q=n;

    for(int i=0;i<n/2;i++)

    {

        if(s[i]==s[q])

        {

         t=0;

         q--;

        }

        else

        {

            t=1;

            break;

        }

    }

    if(t!=0)

    {

        printf("the entered string is a palindrome !\n");

    }

    else

    printf("the entered string is not a palindrome !\n");

}

****

**4. Find the greater no among three nos. using function call macros.**

**5. Write a program to implement predefines macros**

**\_DATE\_**

**\_FILE\_**

**\_LINE\_**

**\_STDC\_**

**\_TIME\_**

#include<stdio.h>

void main()

{

    printf("the use of diferent pedefined macros .....\n");

    printf("currrent File :%s\n", \_\_FILE\_\_ );

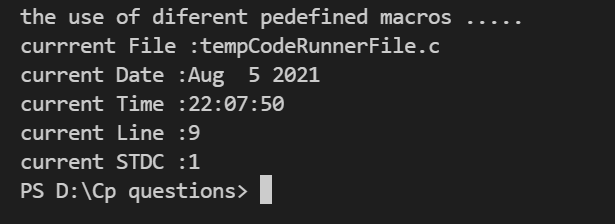
    printf("current Date :%s\n", \_\_DATE\_\_ );

    printf("current Time :%s\n", \_\_TIME\_\_ );

    printf("current Line :%d\n", \_\_LINE\_\_ );

    printf("current STDC :%d\n", \_\_STDC\_\_ );

}

****

**Assignment**

**1. Write a C program to find last occurrence of a character in a given string.**

#include<stdio.h>

#include<string.h>

void main()

{

    char s[50],c,\*w;

    printf("enter the string : ");

    gets(s);

    printf("\nenter the character to search : ");

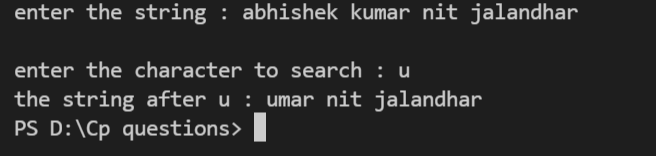
    scanf("%c",&c);

    fflush(stdin);

    w=strrchr(s,c);

    printf("the string after %c : %s",c,w);

}

****

**2. Write a program to print error message using #error directive**

#include <stdio.h>

#include <limits.h>

#define MILLISECONDS(n) (n \* 365 \* 24 \* 60 \* 60 \* 1000)

int main()

{

    int a;

    #if INT\_MAX < MILLISECONDS(12)

    #error Integer size cannot hold our age in milliseconds

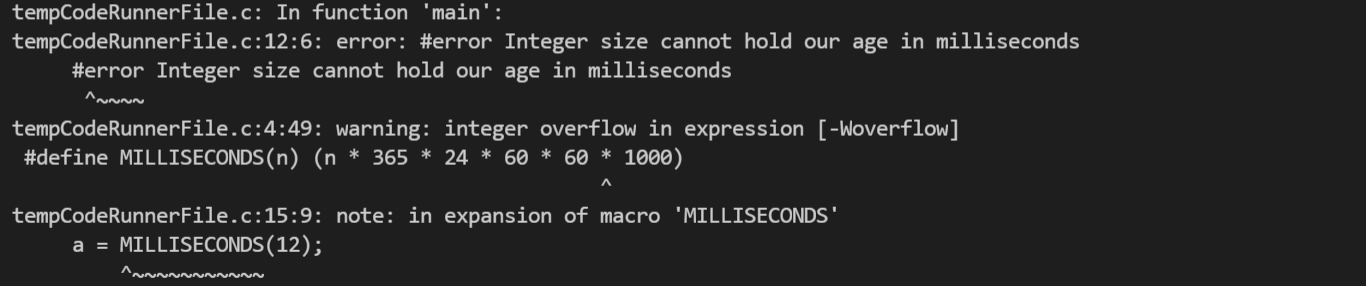
    #endif

    a = MILLISECONDS(12);

    printf("total milliseconds in 12 years are : %d\n",a);

    return 0;

}

****

**Lab 9 : Structure and union**

**1. You are now asked to extend the before assignment program by storing student’s rollno, name along**

**with their marks of six subjects for which you created a structure of five students as follows:**

**struct student**

**{**

**int rollno;**

**char \*name;**

**int sub-marks[6];**

**}s[5];**

**Do the followings on the above structure.**

**i) Display all the student details by passing the structure to a function.**

**ii) Find the total mark obtained by each student in all the six subjects.**

**iii) Print the details of the student who got highest and who got lowest marks.**

#include<stdio.h>

#include<stdlib.h>

struct student

{

    int rollno;

    char \*name;

    int sub\_marks[6];

} s[5];

void input(struct student s[])

{

    for(int i=0;i<5;i++)

    {

        printf("the data for student %d .....\n",i+1);

        printf("enter the rollno. : ");

        scanf("%d",&s[i].rollno);

        printf("\nenter the name of the student : ");

        fflush(stdin);

        getchar();

        gets(s[i].name);

        printf("\nnow enter the markes of the student...\n");

        for(int j=0;j<6;j++)

        {

            printf("enter the marks for subject %d : ",j+1);

            scanf("%d",&s[i].sub\_marks[j]);

            printf("\n");

        }

    }

}

void display(struct student s[])

{

    for(int i=0;i<5;i++)

    {

        printf("the data for student %d .....\n",i+1);

        printf("the rollno. is : %d",s[i].rollno);

        printf("\n the name of the student is : ");

        fflush(stdin);

        puts(s[i].name);

        printf("\nnow the markes of the student are...\n");

        for(int j=0;j<6;j++)

        {

            printf("the marks for subject %d is : %d",j+1,s[i].sub\_marks[j]);

            printf("\n");

        }

    }

}

void total\_marks(struct student s[],int sum[])

{

    printf("\nthe total marks of each students are..... \n");

    for(int i=0;i<5;i++)

    {

        for(int j=0;j<6;j++)

        {

            sum[i]+=s[i].sub\_marks[j];

        }

    }

    for(int i=0;i<5;i++)

    {

        printf("the total marks of student %d : %d\n",i+1,sum[i]);

    }

}

void max\_min\_marks(struct student s[],int sum[])

{

    int max=sum[0],min=sum[0],n=0,m=0;

    for(int i=0;i<5;i++)

    {

        if(max<=sum[i])

        {

            max=sum[i];

            n=i;

        }

        if(min>=sum[i])

        {

            min=sum[i];

            m=i;

        }

    }

    printf("the student with maximun marks is ....\n");

    printf("the rollno. is : %d",s[n].rollno);

    printf("\n the name of the student is : ");

    fflush(stdin);

    puts(s[n].name);

    printf("\nnow the markes of the student are...\n");

    for(int j=0;j<6;j++)

    {

        printf("the marks for subject %d is : %d",j+1,s[n].sub\_marks[j]);

        printf("\n");

    }

    printf("the student with the least marks is...\n");

    printf("the rollno. is : %d",s[m].rollno);

    printf("\n the name of the student is : ");

    fflush(stdin);

    puts(s[m].name);

    printf("\nnow the markes of the student are...\n");

    for(int j=0;j<6;j++)

    {

        printf("the marks for subject %d is : %d",j+1,s[m].sub\_marks[j]);

        printf("\n");

    }

}

void main()

{

    int sum[5]={0,0,0,0,0};

    char std\_name[5][20];

    for(int i=0;i<5;i++)

    s[i].name=std\_name[i];

    printf("enter the data for 5 students....\n");

    input(s);

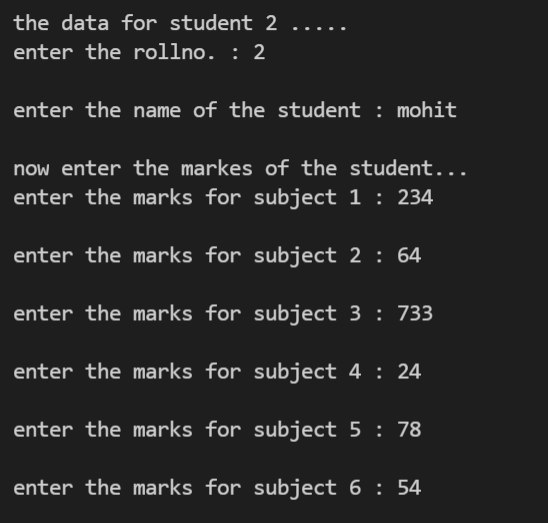
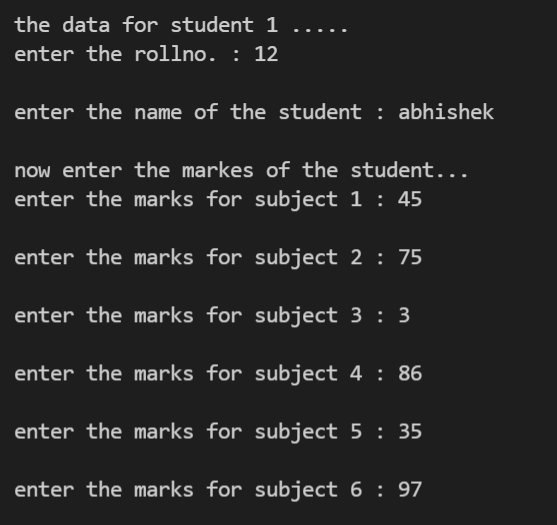
    display(s);

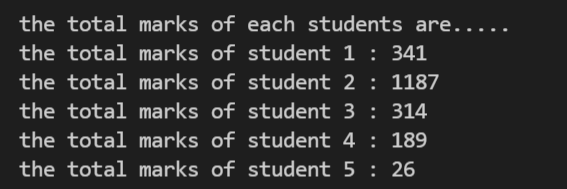
    total\_marks(s,sum);

    max\_min\_marks(s,sum);

}

**The output was very long so ,below is the parts of that output……….**

****

****

**2. Instead of a structure, now you are asked to create a union that stores a student’s rollno, mark for one**

**subject and his name. WAP to create such union and read a student details, and finally print it on the**

**screen. [Hint: Remember union in C, shares the same memory location for all the members and it**

**occupies maximum memory whose size is equal to the memory occupied by the largest member**

#include <stdio.h>

#include <string.h>

union student

{

   char name[15];

   int mark;

   int rollno;

} s[5];

void main()

{

   int m,r;char a[15];

   printf("Enter data of students\n");

   for (int i=0;i<5;i++)

   {

      printf("Enter student's name\n");

      fflush(stdin);

      scanf("%[^\n]s", &s[i].name);

      strcpy(a,s[i].name);

      printf("Enter Roll No\n");

      scanf("%d", &s[i].rollno);

      r=s[i].rollno;

      printf("Enter marks\n");

      scanf("%d", &s[i].mark);

      m=s[i].mark;

      printf("RollNo\tName\tMarks\n");

      printf("%d\t%s\t%d\n",r,a,m);

      a[0]='\0';

   }

   return 0;

}

**1. Define a data type for storing complex number and implement addition, subtraction,**

**multiplication and modulus operations for the defined data type.**

#include <stdio.h>

#include <math.h>

struct complex

{

   int r;

   int i;

};

void main()

{

   struct complex a, b;

   printf("enter both of the complex numbers one by one....\n");

   printf("Enter real and imagnary part of 1st complex number : \n");

   scanf("%d %d", &a.r, &a.i);

   printf("Enter real and imagnary part of second complex number : \n");

   scanf("%d %d", &b.r, &b.i);

   printf("sum of entered complex numbers is %d + %di\n", a.r + b.r,a.i + b.i);

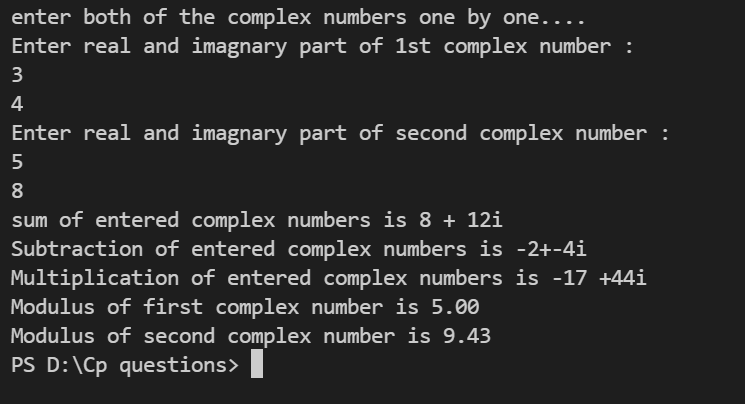
   printf("Subtraction of entered complex numbers is %d+%di\n", a.r - b.r,a.i - b.i);

   printf("Multiplication of entered complex numbers is %d +%di\n", a.r \* b.r -a.i \* b.i, a.r \* b.i + a.i \* b.r);

   printf("Modulus of first complex number is %.2f\n", sqrt(a.r \* a.r + a.i \*a.i));

   printf("Modulus of second complex number is %.2f\n", sqrt(b.r \* b.r + b.i \*b.i));

}

****

**2. Create a structure to specify data of customers in a bank. The data to be stored is:**

**Account number, Name, Balance in account.**

**i. Write a function to print the Account number and name of each customer with**

**balance below Rs. 100**

**ii. If a customer requests for withdrawal or deposit, it is given in the form:**

**Account Number, Amount, Code (1 for deposit, 0 for withdrawal)**

**If on withdrawal the balance falls below Rs. 100 then program give a message**

**” Insufficient Balance”.**

#include <stdio.h>

struct account

{

   int bal;

   int accno;

   char name[15];

} s[5];

void below\_100(struct account s[])

{

   printf("Details of accounts with balance less than 100 are as follows\n");

   printf("Account Number\t\tName\tBalance\n");

   for (int i = 0; i < 5; i++)

   {

      if (s[i].bal < 100)

      {

         printf("%10d\t%10s\t%d\n", s[i].accno, s[i].name, s[i].bal);

      }

   }

}

void main()

{

   printf("Enter details of five Account Holders ........\n");

   for (int i = 0; i < 5; i++)

   {

      printf("Enter name of %dth account holder : \n", i + 1);

      fflush(stdin);

      scanf("%[^\n]s", s[i].name);

      printf("Enter account number : \n");

      scanf("%d", &s[i].accno);

      printf("Enter balance : \n");

      scanf("%d", &s[i].bal);

   }

   below\_100(s);

   printf("Enter 1 for deposit and 0 for withdrawal .....\n");

   int k;

   int l, p;

   scanf("%d", &k);

   if (k == 0)

   {

      printf("Enter account number : \n");

      scanf("%d", &l);

      for (int i = 0; i < 5; i++)

      {

         if (s[i].accno == l)

         {

            printf("Enter amount of withdrawal : \n");

            scanf("%d", &p);

            if (p > s[i].bal)

            {

               printf("Insufficient Balance !\n");

               break;

            }

            else

            {

               printf("Updated Records is : \n");

               printf("Account Number\t\tName\tBalance\n");

               printf("%10d\t%10s\t%d\n", s[i].accno, s[i].name, s[i].bal - p);

               break;

            }

         }

      }

   }

   else if (k == 1)

   {

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

      scanf("%d",&l);

      printf("Enter amount to deposit :\n");

      fflush(stdin);

      scanf("%d",&p);

      for(int i=0;i<5;i++)

      {

         if(s[i].accno==l)

         {

            printf("Updated records are ....\n");

            printf("Account Number\t\tName\tBalance\n");

            printf("%10d\t%10s\t%d\n", s[i].accno, s[i].name, s[i].bal + p);

            break;

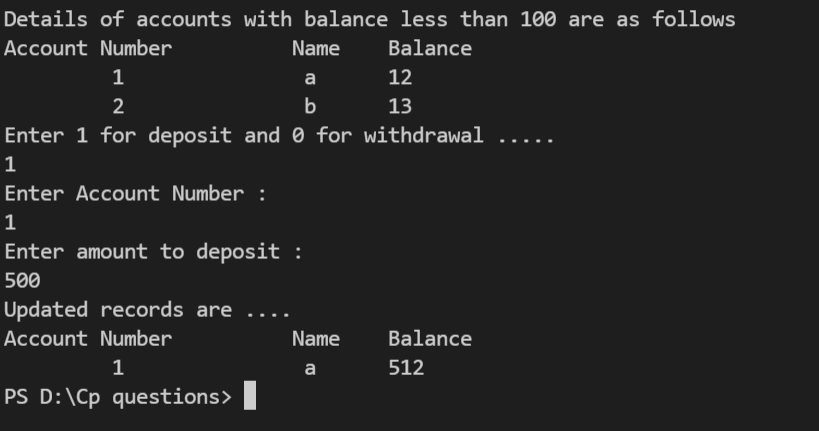
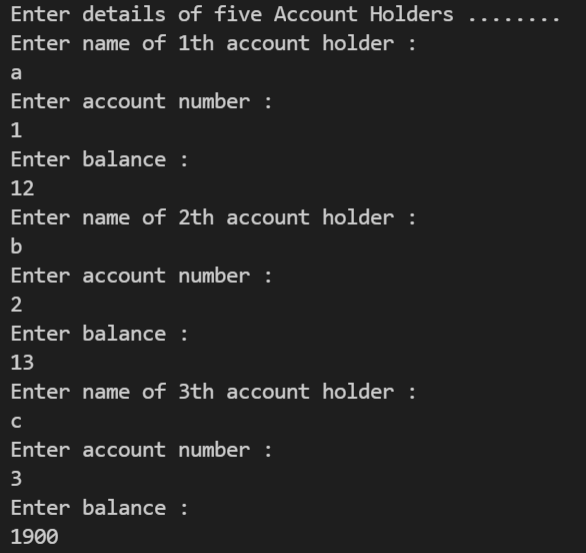
         }

      }

   }

}

**The output was very long so ,below is the parts of that output……….**

****

**3. Develop a phonebook application. It should be able to store, modify and list entries present**

**in the phonebook. A phone book entry consists of the name of the person and his contact**

**information. The name of the person consists of his first name and family name. The contact**

**information consists of the landline number and the mobile number of the person.**

#include <stdio.h>

struct directory

{

long long int mobnumber;

long long int landline;

char first[15];

char last[15];

}s[10];

int main()

{   int k;long long int tempmob;

    int j=0;

    do

    {

        printf("Enter 1 to add a new contact\n");

        printf("Enter 2 to modify an existing contact\n");

        printf("Enter 3 to display all contacts\n");

        printf("Enter 0 to exit\n");

        scanf("%d",&k);

        switch (k)

        {

        case 1: printf("Enter first name of new contact\n");

                fflush(stdin);

                scanf("%s",&s[j].first);

                printf("Enter last name of new contact\n");

                fflush(stdin);

                scanf("%s",&s[j].last);

                printf("Enter mobile number of new contact\n");

                scanf("%lld",&s[j].mobnumber);

                printf("Enter landline number of new conatct\n");

                scanf("%lld",&s[j].landline);

                j++;

            break;

        case 2: printf("Enter mobile number to modify it");

                scanf("%lld",&tempmob);

                for(int i=0;i<j;i++)

                {

                    if(tempmob==s[i].mobnumber)

                    {

                        printf("Enter modified first name of contact\n");

                        fflush(stdin);

                        scanf("%s",&s[i].first);

                        printf("Enter modified last name of contact\n");

                        fflush(stdin);

                        scanf("%s",&s[i].last);

                        printf("Enter modified mobile number\n");

                        scanf("%lld",&s[i].mobnumber);

                        printf("Enter modified landline number\n");

                        scanf("%lld",&s[i].landline);

                        break;

                    }

                }

                break;

        case 3: printf("The numbers currently present in the directory are\n");

                printf("FirstName\tLastName\tMobile\t\tLandline\n");

                for(int i=0;i<j;i++)

                {

                    printf("%9s\t%8s\t%lld\t%lld\n",s[i].first,s[i].last,s[i].mobnumber,s[i].landline);

                }

                break;

        case 0: printf("EXIT\n");

                break;

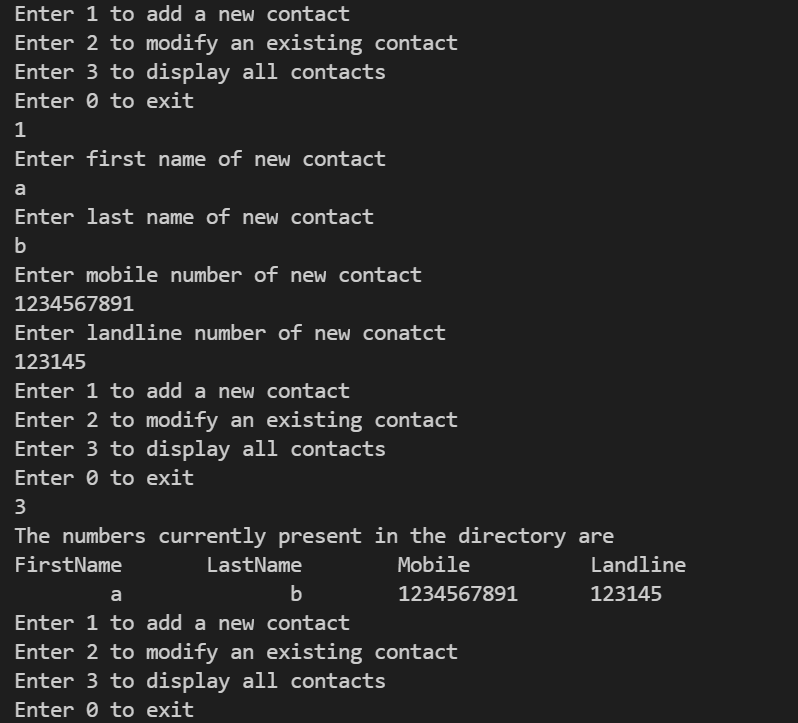
        default: printf("Enter correct response");

            break;

        }

    } while (k!=0);

    return 0;}

****

**4. Create a structure Employee1 that holds the information- Id (integer), Name (character []),**

**Salary (long).Create a union Employee2 that also holds the same information as well as**

**same data type as that of structure Employee1. Write a program in C to find and display**

**the size of structure Employee1 and union Employee2.**

#include <stdio.h>

struct emp\_1

{

   int id;

   char name[20];

   long salary;

};

union emp\_2

{

   int id;

   char name[20];

   long salary;

};

void main ()

{

   struct emp\_1 a;

   union emp\_2 b;

   printf("The memory occupied by structure of Employee 1 is %d\n",sizeof(a));

   printf("The memory occupied by union of Employee 2 is %d\n",sizeof(b));

}

****

**5. Create a structure called Library to hold accession number, title of the book, author name,**

**price of the book and flag indicating whether book is issued or not. Write a menu driven**

**program that implements the working of a library. The menu options should be:**

**i. Add book information**

**ii. Display book information**

**iii. List the count of books in a library**

**iv. List all books of given author**

**v. Exit**

#include <stdio.h>

#include<string.h>

struct library

{

   int access\_num;

   char title[30];

   char author[30];

   int price;

   int flag;

} s[10];

void main()

{

   int k, j = 0,acc;

   do

   {

      printf("The menu options are as follows ....\n");

      printf(" 1 to add new book information\n");

      printf(" 2 to display book information\n");

      printf(" 3 to List the count of books in library\n");

      printf(" 4 to list all the books of a particular author\n");

      printf(" 5 to exit\n");

      scanf("%d", &k);

      switch (k)

      {

         case 1:  printf("Enter book's title : \n");

                  fflush(stdin);

                  scanf("%[^\n]s",&s[j].title);

                  printf("Enter books Accession Number : \n");

                  scanf("%d",&s[j].access\_num);

                  printf("Enter Author's Name : \n");

                  fflush(stdin);

                  scanf("%[^\n]s",&s[j].author);

                  printf("Enter price of Book :\n");

                  scanf("%d",&s[j].price);

                  printf("Enter 1 if it's issued and 0 if it's not : \n");

                  scanf("%d",&s[j].flag);

                  j++;

         break;

         case 2:  printf("Enter the Accession Number of book to display its information : \n");

                  scanf("%d",&acc);

                  for(int i=0;i<j;i++)

                  {

                     if(acc=s[i].access\_num)

                     {

                        printf("The details of book are as follows....\n");

                        printf("Accession Number : %d\n",s[i].access\_num);

                        printf("Title of book is : %s\n",s[i].title);

                        printf("Author of book is : %s\n",s[i].author);

                        printf("Price of book is : %d\n",s[i].price);

                        if(s[i].flag==1)

                        printf("The book is already issued !\n");

                        else

                        printf("The book can be issued !\n");

                     }

                  }

         break;

         case 3:  printf("Books currently available in Library are as follows......\n");

                  printf("Acc Number\tTitle\t\tAuthor\t\tPrice\tFlag\n");

                  for(int i=0;i<j;i++)

                  {

                     printf("%d\t\t%10s\t%10s\t%d\t%d",s[i].access\_num,s[i].title,s[i].author,s[i].price,s[i].flag);

                     printf("\n");

                  }

         break;

         case 4:  printf("Enter the author's name to find its books : \n");

                  char str[15];

                  fflush(stdin);

                  scanf("%[^\n]s",&str);

                  printf("Acc Number\tTitle\t\tAuthor\t\tPrice\tFlag\n");

                  for(int i=0;i<j;i++)

                  {

                     if(strcmp(str,s[i].author)==0)

                     {

                        printf("%d\t\t%10s\t%10s\t%d\t%d\n",s[i].access\_num,s[i].title,s[i].author,s[i].price,s[i].flag);

                     }

                  }

         break;

         case 5:  printf("EXIT ! \n");

         break;

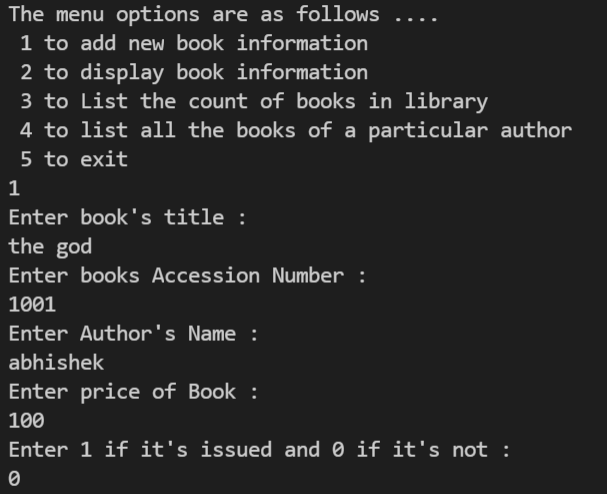
         default: printf("Please Enter Correct Response !\n");

         break;

      }

   } while (k != 5);

}

****

**Assignment**

**1. Define a structure data type called DATE for storing dates. The type contains three integer**

**members: day, month and year. Implement the following operations for the defined data**

**type:**

**i) Isvalid: Check whether the entered date is valid or not, e.g. 31-2-2018 is not valid**

**date since February does not have 31 days.**

**ii) Nextdate: Finds the next date e.g. if the current date is 31-1-2018, then the result of**

**Nextdate operation is 1-2-2018.**

**iii) Datediff: Finds the difference between two dates.**

#include <stdio.h>

struct DATE

{

   int day;

   int month;

   int year;

};

int isValid(struct DATE d)

{

   if (d.day >= 1 && d.day <= 31 && d.month >= 1 && d.month <= 12)

   {

      if (d.month == 2 && d.day > 28)

      {

         if ((d.day == 29) && (d.year % 4 == 0))

         return 1;

         else

         return 0;

      }

      if (d.month == 4 || d.month == 6 || d.month == 9 || d.month == 11)

      {

         if (d.day > 30)

         return 0;

      }

      else

      return 1;

   }

   else

   return 0;

}

void NextDate(struct DATE d)

{

   if(d.day==30&&(d.month==4||d.month==6||d.month==9||d.month==11))

   {

      printf("Next date is : %d/%d/%d",1,d.month+1,d.year);

      return;

   }

      else if (d.month==12&&d.day==31)

   {

      printf("Next date is : %d/%d/%d\n",1,1,d.year+1);

   }

   else

   if(d.day==31&&(d.month==1||d.month==3||d.month==5||d.month==7||d.month==8||d.

   month==10))

   {

      printf("Next date is %d/%d/%d\n",1,d.month+1,d.year);

   }

   else if(d.month==2&&d.day==28&&d.year%4==0)

   {

      printf("Next date is : %d/%d/%d\n",29,d.month,d.year);

   }

   else if(d.month==2&&d.day==28&&d.year%4!=0)

   {

       printf("Next Date is : %d/%d/%d\n",1,d.month+1,d.year);

   }

   else if(d.month==2&&d.day==29&&d.year%4==0)

   {

      printf("Next Date is : %d/%d/%d\n",1,d.month+1,d.year);

   }

   else if(isValid(d))

   {

      printf("Next Date is : %d/%d/%d\n",d.day+1,d.month,d.year);

   }

   else

      printf("Invalid Date Entered !\n");

}

int Difference(struct DATE d1, struct DATE d2)

{

   int arr[12]={31,28,31,30,31,30,31,31,30,31,30,31};

   int sum=0;

   for(int i=0;i<d1.month;i++)

      sum=sum+arr[i];

      int ly=d1.year/4;

      int n1=d1.year\*365+sum+d1.day+ly;

      sum=0;

   for(int i=0;i<d2.month;i++)

      sum=sum+arr[i];

      ly=d2.year/4;

      int n2=d2.year\*365+sum+d2.day+ly;

   return n2-n1>0?n2-n1:n1-n2;

}

void main()

{

   struct DATE d;

   printf("Enter a date to check if its valid or not ......\n");

   printf("Enter day : \n");

   scanf("%d",&d.day);

   printf("Enter month : \n");

   scanf("%d",&d.month);

   printf("Enter year : \n");

   scanf("%d",&d.year);

   if(isValid(d))

   printf("Its a Valid Date !\n");

   else

   printf("It's an invalid date!\n");

   NextDate(d);

   struct DATE d1,d2;

   printf("Enter two dates to find difference between them\n");

   printf("Enter Date 1\n");

   printf("Enter day : \n");

   scanf("%d",&d1.day);

   printf("Enter month : \n");

   scanf("%d",&d1.month);

   printf("Enter year : \n");

   scanf("%d",&d1.year);

   printf("Enter Date 2\n");

   printf("Enter day : \n");

   scanf("%d",&d2.day);

   printf("Enter month : \n");

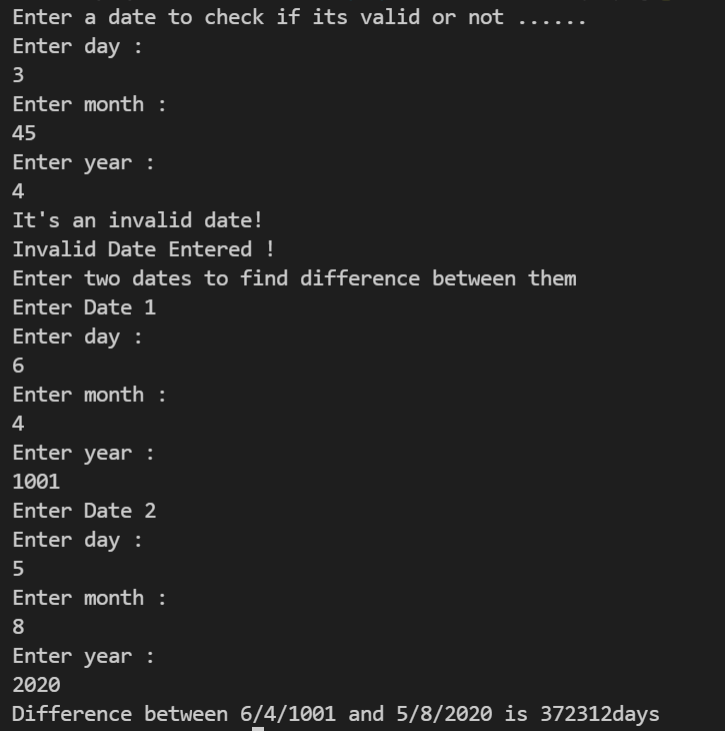
   scanf("%d",&d2.month);

   printf("Enter year : \n");

   scanf("%d",&d2.year);

   printf("Difference between %d/%d/%d and %d/%d/%d is %ddays",d1.day,d1.month,d1.year,d2.day,d2.month,d2.year,Difference(d1,d2));

}

****

**2. Define structure data type TRAIN\_INFO. The type contains:**

**i. Train No: interger type**

**ii. Train name: string**

**iii. Departure time: aggregate type TIME**

**iv. Arrival time: aggregate type TIME**

**v. Start station: string**

**vi. End station: string**

**The structure type TIME contains two integer members: hours and minute. Maintain a train**

**timetable and implement the following operations:**

**i) List all the trains (sorted according to train number) that depart from particular**

**station.**

**ii) List all the trains that depart from a particular station at particular point.**

**iii) List all the trains that depart from a particular station within next one hour of a given**

**time.**

**iv) List all the trains between a pair of start station and end station.**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

struct Time

{

int hours;

int minutes;

};

struct Train\_Info

{

int train\_no;

char train\_name[50];

struct Time departure;

struct Time arrival;

char start[50];

char end[50];

};

void sort(struct Train\_Info t[], int size)

{

for (int i = 0; i < size; i++)

{

int minindex = i;

for (int j = i + 1; j < size; j++)

{

if (t[minindex].train\_no > t[j].train\_no)

{

minindex = j;

}

}

struct Train\_Info temp = t[minindex];

t[minindex] = t[i];

t[i] = temp;

}

}

void print\_train(struct Train\_Info t[], int size)

{

sort(t, size);

printf("S.no\t\tTrainNo\t\tTrain Name\t\tdepTime\t\tArrTime\t\tStart\t\tEnd\n");

for (int i = 0; i < size; i++)

{

printf("%d\t\t%d\t\t%s\t\t%d:%d\t\t%d:%d\t\t%s\t\t%s\n", i + 1,

t[i].train\_no, t[i].train name, t[i].departure.hours, t[i].departure.minutes,

t[i].arrival.hours, t[i].arrival.minutes, t[i].start, t[i].end);

};

}

void train\_station(struct Train\_Info t[], int size)

{

char str[30];

printf("Enter the depart station for which you want to search for the trains: ");

scanf("%s", str);

getchar();

int found = 0;

printf("S.no\t\tTrainNo\t\tTrain Name\t\tdepTime\t\tArrTime\t\tStart\t\tEnd\n");

for (int i = 0; i < size; i++)

{

if (strcmp(str, t[i].start) == 0)

{

found = 1;

printf("%d\t\t%d\t\t%s\t\t%d:%d\t\t%d:%d\t\t%s\t\t%s\n", i + 1,

t[i].train\_no, t[i].train\_name, t[i].departure.hours,

t[i].departure.minutes, t[i].arrival.hours, t[i].arrival.minutes, t[i].start,

t[i].end);

}

}

if (found == 0)

{

printf("No Data Available!\n");

};

}

struct Time next\_time(struct Time t)

{

if(t.hours==23)

{

 t.hours=0;

}

else

t.hours++;

return t;

}

void train\_time(struct Train\_Info t[], int size)

{

char str[30];

struct Time t1, t2;

printf("Enter the depart station for which you want to search for the trains: ");

fflush(stdin);

scanf("%s", str);

fflush(stdin);

printf("Enter the hours and minutes for the time respectively: ");

scanf("%d %d", &t1.hours, &t1.minutes);

int found = 0;

printf("S.no\t\tTrainNo\t\tTrain Name\t\tdepTime\t\tArrTime\t\tStart\t\tEnd\n");

for (int i = 0; i < size; i++)

{

if (strcmp(str, t[i].start) == 0)

{

if ((t[i].departure.minutes-t1.minutes<=60&&t[i].departure.minutest1.minutes>=0)&&(t[i].departure.hours-t1.hours==1||t[i].departure.hours-t1.hours==0))

{

found = 1;

printf("%d\t\t%d\t\t%s\t\t%d:%d\t\t%d:%d\t\t%s\t\t%s\n", i + 1, t[i].train\_no,

t[i].train\_name, t[i].departure.hours, t[i].departure.minutes, t[i].arrival.hours,

t[i].arrival.minutes, t[i].start, t[i].end);

}

}

}

if (found == 0)

{

printf("No Data Available!\n");

}

}

void train\_pair(struct Train\_Info t[], int size)

{

printf("S.no\t\tTrainNo\t\tTrain Name\t\tStart-End\n");

for (int i = 0; i < size; i++)

{

printf("%d\t\t%d\t\t%s\t\t%s-%s\n", i + 1, t[i].train\_no,

t[i].train\_name, t[i].start, t[i].end);

}

}

int main()

{

struct Train\_Info t[3] = {2647, "Punjab Express", {11, 15}, {23, 15}, "Mohali", "Delhi", 7456,

"Bombay Express", {7, 15}, {15, 6}, "Mumbai", "indore", 4516, "Dakshin Express", {12, 25},

{22, 45},

"chennai", "Bangalore"};

while (1)

{

int n;

printf("-------Menu------\n");

printf("Enter:\n1 to List all the trains that depart from particular station.\n2 to List all the

trains that depart from a particular station at particular point.\n3 to List all the trains that

depart from a particular stationwithin next one hour of a given time.\n4 to List all the trains

between a pair of start station and end station.\n5to exit\n");

printf("Enter your choice: ");

scanf("%d", &n);

printf("---\n");

switch (n)

{

case 1:

printf("\t\t\t--Train Details---\n");

print\_train(t, 3);

printf("\t\t\t---\n");

break;

case 2:

printf("\t\t\t--Train Details---\n");

train\_station(t, 3);

printf("\t\t\t---\n");

break;

case 3:

printf("\t\t\t--Train Details--\n");

train\_time(t, 3);

printf("\t\t\t--\n");

break;

case 4:

printf("\t\t\t-Train Details--\n");

train\_pair(t, 3);

printf("\t\t\t-\n");

break;

case 5:

printf("Exiting !\n");

exit(0);

};

}

return 0;

}

