/\* b) Functions without arguments and with return type

• check Armstrong number or not\*/

#include<stdio.h>

#include<math.h>

int checkArmstrongNumber(int n);

int main()

{

     int n, flag;

   printf("enter the number: ");

   scanf("%d",&n);

   flag = checkArmstrongNumber(n);

   if (flag == 1)

   {

      printf("%d is armstrong number",n);

   }

   else

   {

      printf("%d is not armstrong number", n);

   }

   return 0;

}

int checkArmstrongNumber(int num)

{

   int originalNum, remainder,n = 0,flag;

   double result = 0.0;

   // (store the number of digits of num in n)

   for (originalNum = num; originalNum != 0; ++n)

   {

      originalNum /= 10;

   }

   for (originalNum = num; originalNum != 0; originalNum /= 10)

    {

      remainder = originalNum % 10;

      // (store the sum of the power of individual digits in result)

      result += pow(remainder, n);

   }

   // (condition for Armstrong number)

   if (round(result) == num)

      flag = 1;

   else

      flag = 0;

   return flag;

}

Output:-

enter the number: 520

520 is not armstrong number

enter the number: 135

135 is not armstrong number

enter the number: 153

153 is armstrong numbe

/\* e) Function return Multiple values

• simple calculator (add, sub, mul, div, mod)\*/

#include <stdio.h>

//Function declarations for calculator

float add(float num1, float num2);

float sub(float num1, float num2);

float mult(float num1, float num2);

float div(float num1, float num2);

int mod(int num1, int num2);

int main()

{

     char op;

    float num1, num2,result=0.0f;

    printf("Enter number1,operator,number2:\n");

    scanf("%f %c %f", &num1, &op, &num2);

    switch(op)

    {

        case '+':

            result = add(num1, num2);

            break;

        case '-':

            result = sub(num1, num2);

            break;

        case '\*':

            result = mult(num1, num2);

            break;

        case '/':

            result = div(num1, num2);

            break;

        case ' % ':

             result = mod(num1,num2);

             break;

        default:

            printf("Invalid operator");

    }

    printf("%.2f %c %.2f = %.2f", num1, op, num2, result);

    return 0;

}

float add(float num1, float num2)

{

    return num1 + num2;

}

float sub(float num1, float num2)

{

    return num1 - num2;

}

float mult(float num1, float num2)

{

    return num1 \* num2;

}

float div(float num1, float num2)

{

    return num1 / num2;

}

int mod(int num1, int num2)

{

    return  num1 % num2;

}

Output:-

Enter number1,operator,number2:

25

+

25

25.00 + 25.00 = 50.00