# Lab instructions
# **Week 05**

# Introduction to Programming
# ECS 102, 2018-19 Semester II
# IISER Bhopal

# square_root_2.c

In week 3, you have written an iterative program using ***while*** loop to calculate the square root of a given number $x$. Use data type ***float*** for both

$$\text{guess } g \text{ and } \text{error} = \text{fabsf}(g * g - x).$$

a) Use input $x = 32767$, and see whether the program is converging, i.e. check whether the error is becoming less than 0.0001. Print error in each iteration to see what is happening.

b) Terminate your program inside the ***while*** loop if number of iterations exceeds 100 using ***break*** statement.

c) Use ***do-while*** loop to implement your program.

# square_root_2.c (continued)

d)  Use *for* loop to implement your program.

e)  You can check both the iteration count (when it becomes 100) and error in the condition checking for all the three types of loops and avoid break statement. Implement it for all the three loops.

f)  Use data type *double* for both

   guess $g$ and  error $= \mathrm{fabs}(g * g - x),$

   (Note: use **fabs** instead of **fabsf**).

   Use input $x = 32767$, and see whether the program is converging.

d)  Use *goto* statement to keep reading a number until and unless you get a positive number.

# use_function.c

```c
#include <stdio.h>

int multiplication(int m, int n);

void main()
{
        int a, b, c;
        a = 3;
        b = 2;
        c = multiplication(a,b);
        printf("multiplication (%d,%d): %d\n", a, b, c);
}

int multiplication(int m, int n)
{
        int c = m*n;
        return c;
}
```

**You can use different variable names in the function i.e., it does not need to be (int a, int b)**

**Declare another function multiplicationf(float a, float b)**

**You can declare the same variable 'c' inside the function, as in the main function**

# use_function.c (continued)

(a) Write the square root algorithm inside a function.

(b) Read 10 numbers from user input in a loop and call the square root function to calculate the square root of a number in each iteration.

(c) Use **continue** statement to avoid execution of the square root function if the number is negative.

(d) If the number is negative, calculate the square root of the corresponding positive number multiplied by $i$, e.g., square root of -16 is $4i$.

# use_switch.c

Use *switch* statement to determine the grade given the input as marks between 0 and 100 following the rules given in the table below.

| marks | grade |
|-------|-------|
| 90-100 | O |
| 80-89 | A |
| 60-79 | B |
| 50-59 | C |
| 40-49 | D |
| <40 | F |

# linear_equation.c

Say we have a certain numbers of chickens and cows. User will give inputs of number of heads and number of legs. There are no amputees.

(a) You need to determine the numbers of chickens and cows.

(b) If there is no solution, you need to print that there is no solution.

# linear_equation_2.c

Say we have a certain numbers of chickens, cows, and spiders. User will give inputs of number of heads and number of legs. There are no amputees.

(a) You need to print the numbers of chickens, cows, and spiders. Print all the possible solutions.

(b) If there is no solution, you need to print that there is no solution.