



LAB 2: Introduction to Programming

```
+++++
Instructions:
A) Save your lab.doc as LAB_no_RollNo.doc. At the end of lab you need to submit your all programs along with the output.
-- LAB_No_Roll_No_2hr.doc for lab task executed during the lab
-- LAB_No_Roll_No_complete.doc for Full solution of the Lab assignment ( It should contain all lab assignment/problems)
B) Use/paste the snapshot of the steps followed along with result/s.
C) Mention your observation/comment after results in the doc.
D) Along with the doc/pdf file you need to upload your c program files with following nomenclature.
-- LAB_No_Prob_No.c
+++++
```

Objective(s):

- To be familiar with syntax and structure of C-programming.

+++++

Sample Problem

Write a Program to calculate and display the area of a RECTANGLE having its Length (l=12cm) and width (w=10cm) .

Problem Analysis:

The problem is to calculate the volume of a RECTANGLE having its input parameters identified as: **length** (integer type) and **width** (integer type) and depth (integer type).

The output of the program is to display the area; hence the output parameter is identified as area (integer type).

During the processing or calculation phase, we don't need any extra parameters (variables) for this problem.

The area of the RECTANGLE is the multiplication of its length and width hence the mathematical formula to calculate area is:

area = Length * width. (area = len * w)

Input Variable	Processing variable/ Calculations	Output variables	Necessary header files/functions/macros
len (int) w(int)	Area=len*w	Area(int)	Stdio.h printf()

Algorithm:

Start

Define variables: len(int), w(int), Area(int)

Assign value to variables: len= 12, w=10

Calculate the Area as: Area=len*w

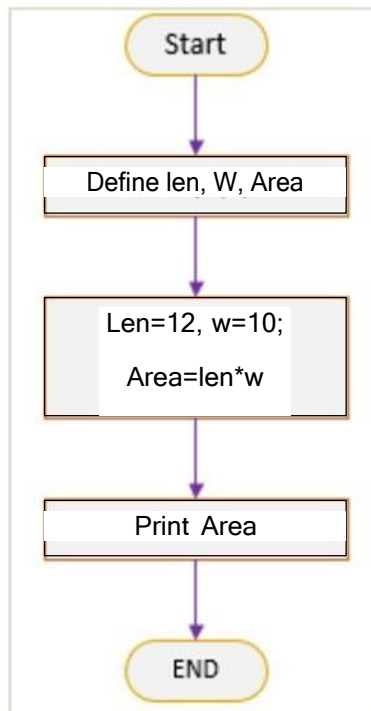
Display the Area (Area)

Stop



LAB 2: Introduction to Programming

Flowchart:



Code:

//Following code is written and compiled in Linux Vi editor

```
#include<stdio.h>

int main(void)
{
    //start the program
    Int len,w,Area;           //variables declaration
    len=12;w=108;             //assign value to variables
    Area=len*w;               //calculation using mathematical formula
    printf("The Area of the RECTANGLE is: %d",Area);
    //display the Area

    return 0;                 //end the main program
}
```

Output (Compilation, Debugging & Testing)

The Area of the RECTANGLE is: 120

Discussion and Conclusion

This is the first code written in C program. The program is focused on the calculation of area of a RECTANGLE for the given length and width. From this lab, I understood the basic structure of C programming including the meaning of header files & steps of problem solving. Hence, Area of a RECTANGLE is calculated and displayed.

+++++



CSB101: Problem Solving and Computer Programming

LAB 2: Introduction to Programming LAB 1 Exercise

Daksh Verma
231210036

PART A : Compulsory Questions

1. The program discussed in the class “*demo.c*” is given. Download and save it on your local system.
 - a) Compile and run the demo program

Problem Analysis:

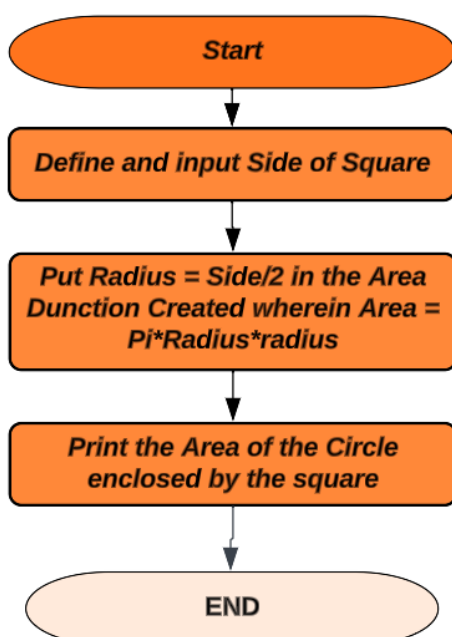
The problem is to calculate and print the area of a circle that is enclosed by a square with a given side length. To solve this problem, we'll need to use the formula for the area of a circle and the area of a square.

Input Variable	Processing variable/ Calculations	Output variables	Necessary header files/functions/macros
Side (float)	AreaofCircle = $\text{Pi} * \text{side} / 2 * \text{side} / 2$	Area (float)	Stdio.h printf()

Algorithm:

1. Start
2. Input the side length of the square.
3. Calculate the radius of the circle as half of the side length: **radius = side_length / 2**.
4. Calculate the area of the circle using the formula: **area_circle = π * radius * radius**, where π is a constant (approximately 3.1416).
5. Display the area of the circle.
6. Stop

Flowchart



Code

```
/* Program to compute the Area of circle enclosing square of side */  
  
#include <stdio.h> /* Compulsory, for library files */  
  
#define PI 3.1416  
  
double area_of_circle(float);  
double area_of_circle (float radius)  
{  
    return PI*radius*radius;  
}  
  
int main() /* Function heading */  
{  
    int squareSide; /* Declare variables */  
    double area; /* Declare variables */  
    scanf("%d", &squareSide); /* Read squareSide */  
    area = area_of_circle(squareSide/2);  
    printf("Area of circle enclosing square of side %d is: %f\n", squareSide, area);  
    return 0;  
}
```



CSB101: Problem Solving and Computer Programming

LAB 2: Introduction to Programming

Daksh Verma
231210036

Output:

```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ ./opt
90
Area of circle enclosing square of side 90 is: 6361.740000
```

Discussion and Conclusion:

This C program calculates and prints the area of a circle enclosed by a square with a given side length. It involves basic arithmetic operations, such as squaring and multiplying by π .



CSB101: Problem Solving and Computer Programming

LAB 2: Introduction to Programming

Daksh Verma
231210036

2. Write a program in C to display “Welcome to NIT Delhi”

Problem Analysis:

The problem is to create a simple C program that displays the message "Welcome to NIT Delhi" on the screen. This program does not require any input variables or complex calculations. It involves using the printf() function to print the desired message.

Input Variable	Processing variable/ Calculations	Output variables	Necessary header files/functions/macros
None	None	None	Stdio.h (Not Necessary) printf()

Algorithm:

Start

Display the message "Welcome to NIT Delhi" using the printf() function.

Stop

Flowchart



Code

```
daksh@Ubuntu: ~/Desktop/Daksh/Coding/C
/* Code to Print" Welcome to NIT Delhi"
 * Code by - Daksh Verma
 * Roll No - 231210036 */

//Preprocessing statement to include stdio header file
#include <stdio.h>

//Calling of required main() function
int main(){

    //using printf function to print the required line
    printf("Welcome to NIT Delhi\n");

    //returning 0 value to int main() function
    return 0;}
```

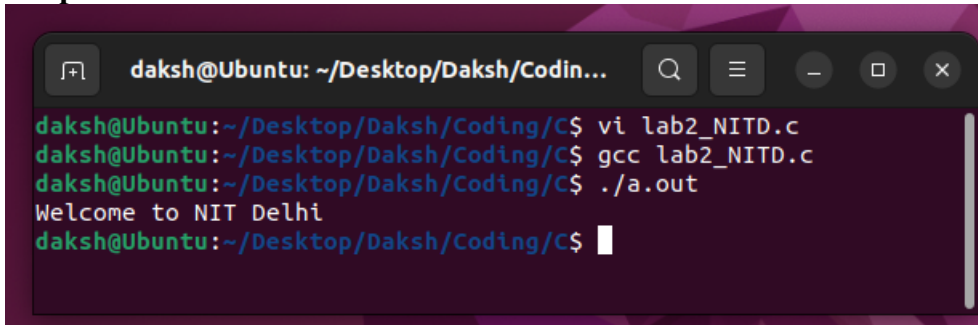


CSB101: Problem Solving and Computer Programming

LAB 2: Introduction to Programming

Daksh Verma
231210036

Output:

A screenshot of a terminal window with a dark purple background. The window title is 'daksh@Ubuntu: ~/Desktop/Daksh/Codin...'. The terminal shows the following commands and output:

```
 daksh@Ubuntu:~/Desktop/Daksh/Coding/C$ vi lab2_NITD.c
 daksh@Ubuntu:~/Desktop/Daksh/Coding/C$ gcc lab2_NITD.c
 daksh@Ubuntu:~/Desktop/Daksh/Coding/C$ ./a.out
Welcome to NIT Delhi
 daksh@Ubuntu:~/Desktop/Daksh/Coding/C$
```

Discussion and Conclusion:

This C program is a basic example of how to use the `printf()` function to display a message on the screen. It doesn't involve any input or complex processing. Such programs are often used for simple greetings or informational messages in the beginning stages of learning C programming.



CSB101: Problem Solving and Computer Programming

LAB 2: Introduction to Programming

Daksh Verma
231210036

3. Write a program in C to display “**Hello X, Welcome to NIT Delhi!!**”, where X is your name. After this display “**Welcome to NIT Delhi**” 4 times.

Problem Analysis:

This problem consists of two parts:

1. Displaying a welcome message that includes inputted name.
2. Displaying the message "Welcome to NIT Delhi" four times using a while loop.

For the first part, we will need to input your name, and for the second part, we will use a while loop to display the message.

Input Variable	Processing variable/ Calculations	Output variables	Necessary header files/functions/macros
Name (character/string)	None	None(int)	Stdio.h printf() scanf()

Algorithm:

1. Start
2. Input name as a string.
3. Display the message "Hello X, Welcome to NIT Delhi!!" where X is replaced with your name.
4. Initialize a variable **count** to 0.
5. While **count** is less than 4, do the following: a. Display the message "Welcome to NIT Delhi" using **printf()**. b. Increment **count** by 1.
6. Stop

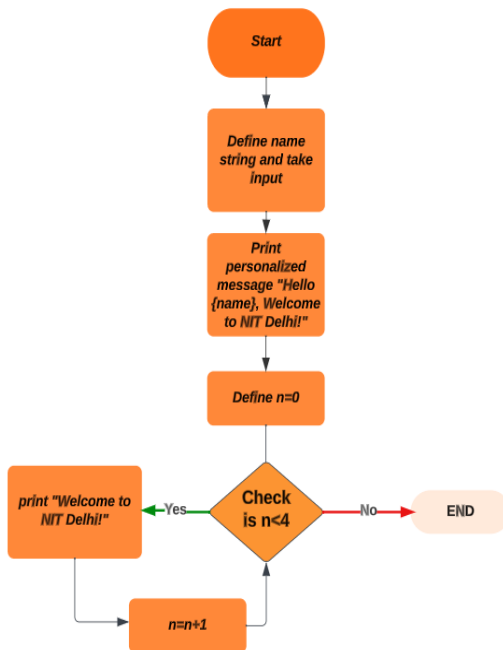


CSB101: Problem Solving and Computer Programming

LAB 2: Introduction to Programming

Daksh Verma
231210036

Flowchart



Code

```
daksh@Ubuntu: ~/Desktop/Daksh/Coding/C
/* Code to display "Hello X, welcome to NIT Delhi!", where X is your name.
 * After this display " welcome to NIT Delhi" 4 time."
 * Code by - Daksh Verma
 * Roll No - 231210036 */

#include <stdio.h>

int main(){
    printf("Enter your name:\n");
    char name[20];           //Variable definition of name variable
    scanf("%[^\n]s", name);  //Takin Input of name

    printf("Hello %s, Welcome to NIT Delhi!!\n",name);
    //Printing statement with input name variable

    int n=0;

    //Using while loop to iterate value of n from 0 to 3 (Loop runs 4 times)
    while (n<4){
        printf("Welcome to NIT Delhi!\n");
        n++;                //using ++ to add 1 to the value of the integer n
    }
    return 0;
}
```

Output:

```
daksh@Ubuntu: ~/Desktop/Daksh/Coding/C
daksh@Ubuntu:~/Desktop/Daksh/Coding/C$ gcc lab2_HelloX.c
daksh@Ubuntu:~/Desktop/Daksh/Coding/C$ ./a.out
Enter your name:
Daksh Verma
Hello Daksh Verma, Welcome to NIT Delhi!!
Welcome to NIT Delhi!
Welcome to NIT Delhi!
Welcome to NIT Delhi!
Welcome to NIT Delhi!
daksh@Ubuntu:~/Desktop/Daksh/Coding/C$
```

Discussion and Conclusion:

This C program first takes your name as input and displays a personalized welcome message. Then, it uses a while loop to display the "Welcome to NIT Delhi" message four times. This example demonstrates input handling, string formatting, and the use of loops in C.

Observation:

When working with strings and using printf() function, the program outputs the value of the string till the first space character in it . To rectify this issue we use “[^\n]%s” instead of “%s” because whenever we write “^\n” in scanf function, we will be able to write string with spaces and also able to store it in a particular variable(here, str variable), until “\n” is encountered.



CSB101: Problem Solving and Computer Programming

LAB 2: Introduction to Programming

Daksh Verma
231210036

- Write a program in C to display the table of 50.

Problem Analysis:

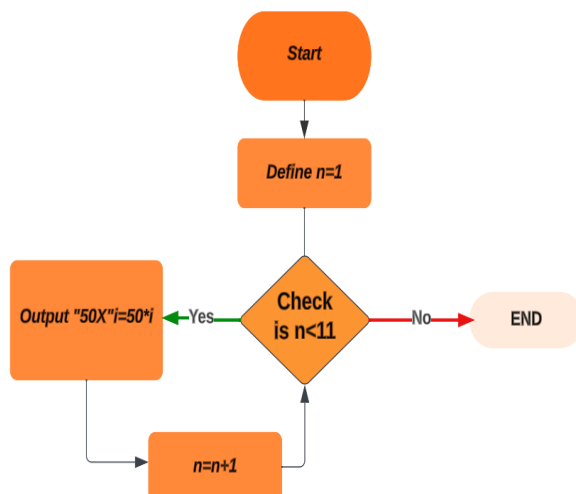
The problem is to create a simple C program that displays the multiplication table of the number 50. In this table, we will display the product of 50 multiplied by each integer from 1 to 10. The output will show the multiplication results for each of these calculations.

Input Variable	Processing variable/ Calculations	Output variables	Necessary header files/functions/macros
None	Multiplication of 50 by integers from 1 to 10	None	Stdio.h printf()

Algorithm:

- Start
- Initialize a variable **number** with the value 50.
- Create a loop that iterates from **i** equal to 1 to 10. a. Calculate the product of **number** and **i**. b. Display the result as "50 x i = product" where **i** is the current integer, and **product** is the calculated product.
- End the loop.
- Stop

Flowchart



Code

```
/* Code to Print table of 50
 * Code by - Daksh Verma
 * Roll No - 231210036*/

#include <stdio.h>

int main(){
    //Declaring variables i
    int i=1;
    //starting loop to iterate i from 1 to 10

    while (i<11){
        //printing table
        printf("50 X %d = %d\n",i,50*i);
        i++;
    }
    return 0;
}
```



CSB101: Problem Solving and Computer Programming

LAB 2: Introduction to Programming

Daksh Verma
231210036

Output:

```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C$ gcc lab2_tableof50.c
daksh@Ubuntu:~/Desktop/Daksh/Coding/C$ ./a.out
50 X 1 = 50
50 X 2 = 100
50 X 3 = 150
50 X 4 = 200
50 X 5 = 250
50 X 6 = 300
50 X 7 = 350
50 X 8 = 400
50 X 9 = 450
50 X 10 = 500
daksh@Ubuntu:~/Desktop/Daksh/Coding/C$
```

Discussion and Conclusion:

This C program demonstrates how to create a simple multiplication table for the number 50. It uses a loop to calculate and display the products of 50 multiplied by integers from 1 to 10.

Observation:

The increment operator (++) is used to increment the value of a variable in an expression by 1. In C, ++ and -- operators are called increment and decrement operators. They are unary operators needing only one operand. Hence ++ as well as -- operator can appear before or after the operand with same effect.



CSB101: Problem Solving and Computer Programming

LAB 2: Introduction to Programming

Daksh Verma
231210036

5. Write a program in C to take two number (40 and 30) and perform following operations
- Print these two numbers
 - Addition,
 - Subtraction and
 - Multiplication

Problem Analysis:

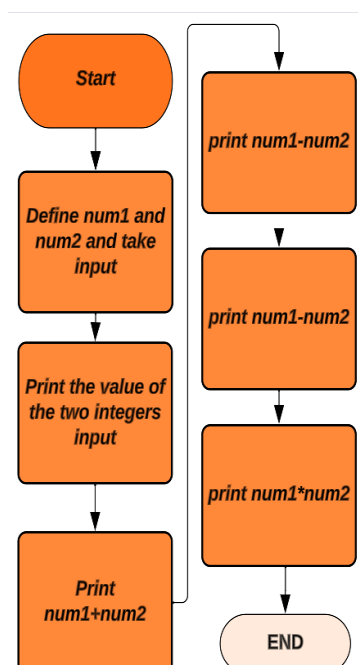
The problem is to create a C program that takes two numbers, 40 and 30, as input and performs various mathematical operations on them. The operations to be performed include addition, subtraction and multiplication.

Input Variable	Processing variable/Calculations	Output variables	Necessary header files/functions/macros
First and Second number (integer)	None (Processing done in print statement for lower memory consumption) Calculations: Addition, Subtraction and multiplication of Two numbers	None	Stdio.h printf() scanf()

Algorithm:

1. Start
2. Input the first number (40).
3. Input the second number (30).
4. Display the two numbers.
5. Display the Addition of the two numbers.
6. Display the subtraction result of number 2 from number 1
7. Display the multiplication result of the two numbers.
8. Stop

Flowchart



Code

```
/* Program to take in two numbers and print,add,subtract and multiply them.
   Code by - Daksh Verma
   * Roll No - 231210036 */

#include <stdio.h>
// using void instead of int Main and returning value 0

void main(){

    //Defining and inputing 2 numbers
    int num1, num2;
    printf("Enter the two numbers \n");
    scanf("%d %d",&num1,&num2);

    //Printing the 2 numbers

    printf("Number 1: %d \nNumber 2: %d\n",num1,num2);

    //Adding the 2 numbers
    printf("Addition of the two numbers is %d \n",num1+num2);

    //Subtracting the second number from the first
    printf("Subtraction of the two numbers is %d \n",num1-num2);

    //Multiplying the two numbers
    printf("Multiplication of the two numbers is %d \n",num1*num2);

}
```



CSB101: Problem Solving and Computer Programming

LAB 2: Introduction to Programming

Daksh Verma
231210036

Output:

```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C$ gcc lab2_2numbers.c
daksh@Ubuntu:~/Desktop/Daksh/Coding/C$ ./a.out
Enter the two numbers
40
30
Number 1: 40
Number 2: 30
Addition of the two numbers is 70
Subtraction of the two numbers is 10
Multiplication of the two numbers is 1200
daksh@Ubuntu:~/Desktop/Daksh/Coding/C$
```

Discussion and Conclusion:

This C program takes two predefined numbers, 40 and 30, and performs basic mathematical operations on them, including addition, subtraction and multiplication. It serves as a simple example of performing arithmetic operations in C and displaying the results.

Observation:

In this code I used a non standard void main() function for experimentation. The void main() function returns no return value when the program is successfully terminated. There is an empty data type. It is used when one doesn't want to return any value to the main function. In C, void main() is one of the ways to define the entry point but in a non-standard way.



LAB 2: Introduction to Programming

PART B : Exploratory Problem

6. Explore the different compilation methods in Unix (optional like a warning, debugging, linking libraries, object file etc) using the 1st problem program.

[*Hint* : use this [LINK](#) for the option required while compiling a source code]

```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ gcc demo.c
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ ls
```

- -o opt: This option compiles the source.c file, but instead of providing the default output name, it generates an executable named opt that can be executed using ./opt. The -o flag is utilized for specifying the output file.

```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ gcc demo.c -o opt
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ ls
a.out  demo.c  opt
```

- -Werror: When employed, this flag compiles the source file and displays warnings if any errors are present in the program. The -W flag is employed to handle warnings.

Code without error

```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ gcc demo.c -Werror -o opt
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ ls
a.out  demo.c  opt
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ ./opt
5
Area of circle enclosing square of side 5 is: 12.566400
```

Code with error

```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ cat source.c
//code for Q2 lab2
#include <stdio.h>

int main(){
    printf("Hello World");
    int n;
    printf(n);
    return 0
}
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ gcc source.c -Werror -o opt
source.c: In function 'main':
source.c:5:29: error: expected ')' before ';' token
    5 |         printf("Hello World");
      |                        ~         ^
source.c:8:17: error: expected ';' before '}' token
    8 |         return 0
      |                ^
    9 |     }
      |     ~
```



CSB101: Problem Solving and Computer Programming

Daksh Verma
231210036

LAB 2: Introduction to Programming

- -Wall: Enabling this option not only checks for errors but also scrutinizes all types of warnings, including those related to unused variables. It is considered a good practice to include this flag while compiling code.

```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ gcc demo.c -Wall -o opt
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ ls
a.out demo.c opt
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ ./opt
7
Area of circle enclosing square of side 7 is: 28.274400
```

Code with unused variable

```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ cat source.c
//code for Q2 lab2
#include <stdio.h>

int main(){
    printf("Hello World");
    int n;
    return 0;
}
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ gcc source.c -Wall -o opt
source.c: In function 'main':
source.c:6:13: warning: unused variable 'n' [-Wunused-variable]
   6 |         int n;
     |         ^
```

- -g: This command grants permission to debug the program using the GNU Debugger (gdb), which will be explained later. The -g option is used for debugging purposes.

```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ gcc -g demo.c -Wall -o opt
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ ./opt
7
Area of circle enclosing square of side 7 is: 28.274400
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$
```

- -lm: By using this command, you link the math.h library to your source file. The -l option is employed to specify the library to link, and for math.h, -lm is used.

```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ gcc demo.c -Wall -o opt -lm
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ ./opt
9
Area of circle enclosing square of side 9 is: 50.265600
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$
```

(I couldn't figure out how run the code using lm without including the math header)

```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ cat source.c
//code for Q2 lab2
#include <stdio.h>

int main(){
    printf("Hello World");
    int n=100;
    float sqroot=sqrt(n);
    printf("Square root of %d is %lf",n,sqroot);
    return 0;
}
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ gcc source.c -o opt -lm
source.c: In function 'main':
source.c:7:22: warning: implicit declaration of function 'sqrt' [-Wimplicit-function-declaration]
   7 |     float sqroot=sqrt(n);
     |                      ^~~~~
source.c:3:1: note: include '<math.h>' or provide a declaration of 'sqrt'
   3 |
     | #include <stdio.h>
+++ |+#include <math.h>
   3 |
source.c:7:22: warning: incompatible implicit declaration of built-in function 'sqrt' [-Wbuiltin-declaration-mismatch]
   7 |     float sqroot=sqrt(n);
     |                      ^~~~~
source.c:7:22: note: include '<math.h>' or provide a declaration of 'sqrt'
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$
```



CSB101: Problem Solving and Computer Programming

Daksh Verma
231210036

LAB 2: Introduction to Programming

- -std=c11: This command specifies the usage of the C11 version of standards when compiling the source.c program. This version permits variable definitions within loop initializations, and employing newer standards is recommended.

```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ gcc -Wall -std=c11 demo.c -o opt
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ ./opt
96
Area of circle enclosing square of side 96 is: 7238.246400
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$
```

- -c: When this option is utilized, the program is compiled, resulting in an object file as output. Object files are typically used in the creation of libraries.

```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ gcc -c demo.c
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ ls
a.out demo.c demo.o opt opy
```

- -v: This option serves a verbose purpose, providing additional details and information during the compilation process.

```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab2_Q6$ gcc -v demo.c
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/11/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none:amdgc- amdhsa
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 11.4.0-1ubuntu
prefix=/usr --with-gcc-major-version-only --program-suffix=-11 --program-pref
osix --libdir=/usr/lib --enable-nls --enable-bootstrap --enable-clocale=gnu --
.....
```

Observation /Comments:

These problems provide a progressive learning experience in C programming. They start with simple output display tasks, gradually introduce user input and arithmetic operations, and culminate in a program that combines various elements. Having knowledge of Python programming, these problems helped me become familiarize myself with the syntax and structure of C programming, including variable handling, loops, conditional statements, and the use of library functions like printf() and scanf() with various data types.

