LAB 4

Subject: programming fundamental

Submitted to: Dr. Kashif Bilal

***Submitted by:* KASHIF KHAN**

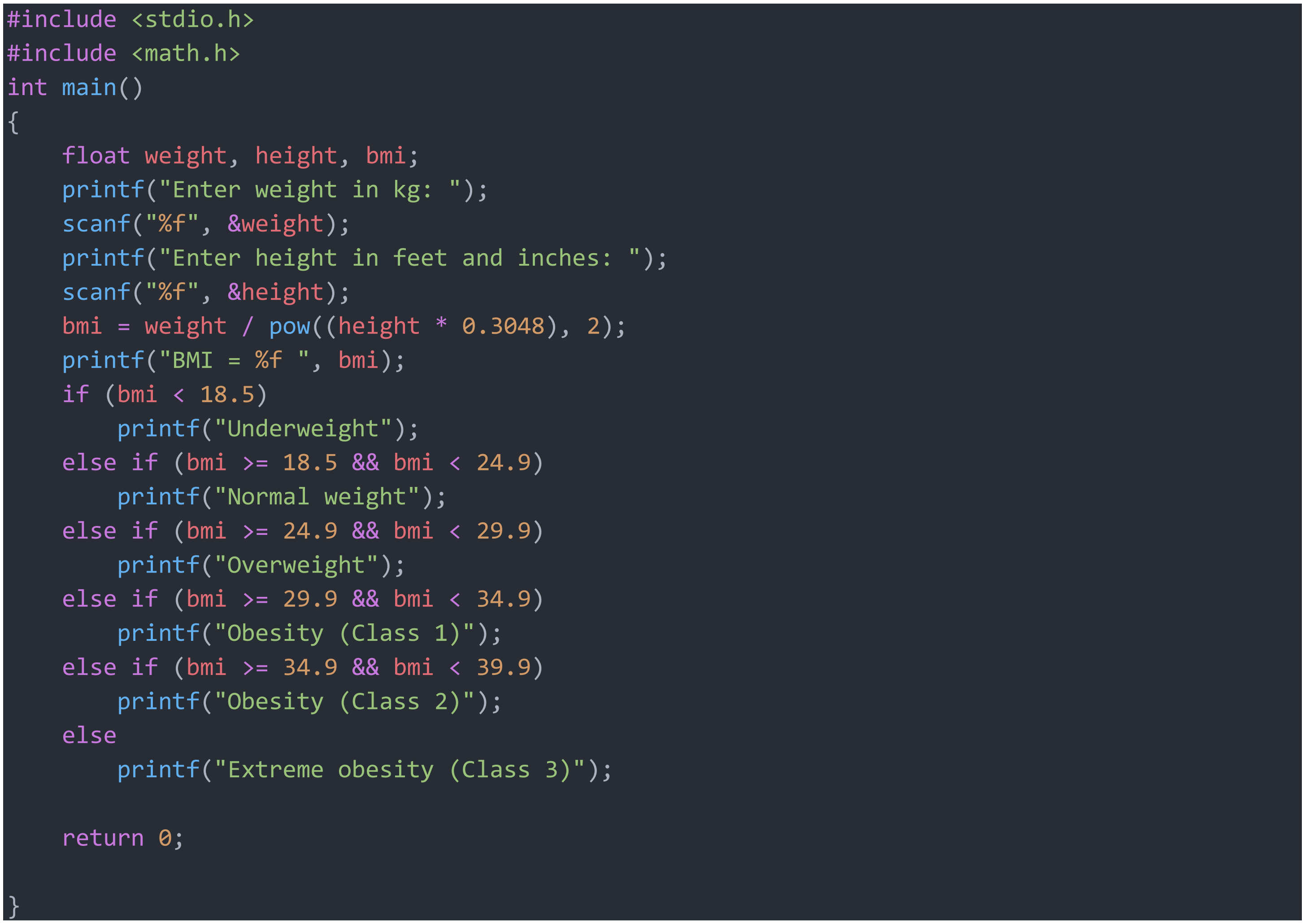
***Student Reg#:* FA22-BSE-O68**

Date: 29.03.2023

Q1: **Body Mass Index (BMI) Calculator and Category:** Write a program that takes two float inputs, weight (in kilograms) and height (in feet and inches), and calculates the BMI (Body Mass Index) of a person. The program should then output the BMI value and the BMI category according to the following classifications: • Underweight: BMI < 18.5

- Normal weight: 18.5 <= BMI < 24.9
- Overweight: 24.9 <= BMI < 29.9
- Obesity (Class 1): 29.9 <= BMI < 34.9
- Obesity (Class 2): 34.9 <= BMI < 39.9
- Extreme obesity (Class 3): BMI >= 39.9

BMI Formula: weight (kg) / [height (m)]2
1 meter = 39. 37 inches.

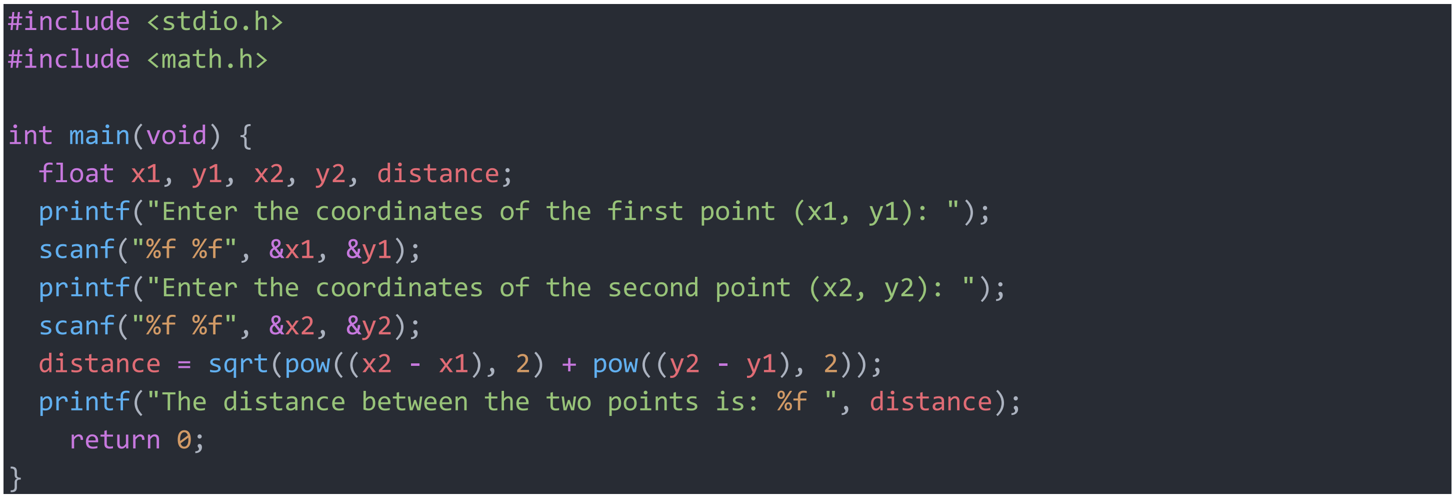


```c
#include <stdio.h>
#include <math.h>
int main()
{
    float weight, height, bmi;
    printf("Enter weight in kg: ");
    scanf("%f", &weight);
    printf("Enter height in feet and inches: ");
    scanf("%f", &height);
    bmi = weight / pow((height * 0.3048), 2);
    printf("BMI = %f ", bmi);
    if (bmi < 18.5)
        printf("Underweight");
    else if (bmi >= 18.5 && bmi < 24.9)
        printf("Normal weight");
    else if (bmi >= 24.9 && bmi < 29.9)
        printf("Overweight");
    else if (bmi >= 29.9 && bmi < 34.9)
        printf("Obesity (Class 1)");
    else if (bmi >= 34.9 && bmi < 39.9)
        printf("Obesity (Class 2)");
    else
        printf("Extreme obesity (Class 3)");

    return 0;

}
```

Q2 : **Calculate the distance between two points in a 2D plane:** Write a program that takes four float inputs (x1, y1, x2, y2) representing the coordinates of two points (P1 and P2) in a 2D plane. The program should calculate and output the Euclidean distance between these two points.

Formula: **distance = sqrt((x2 - x1)^2 + (y2 - y1)^2)**

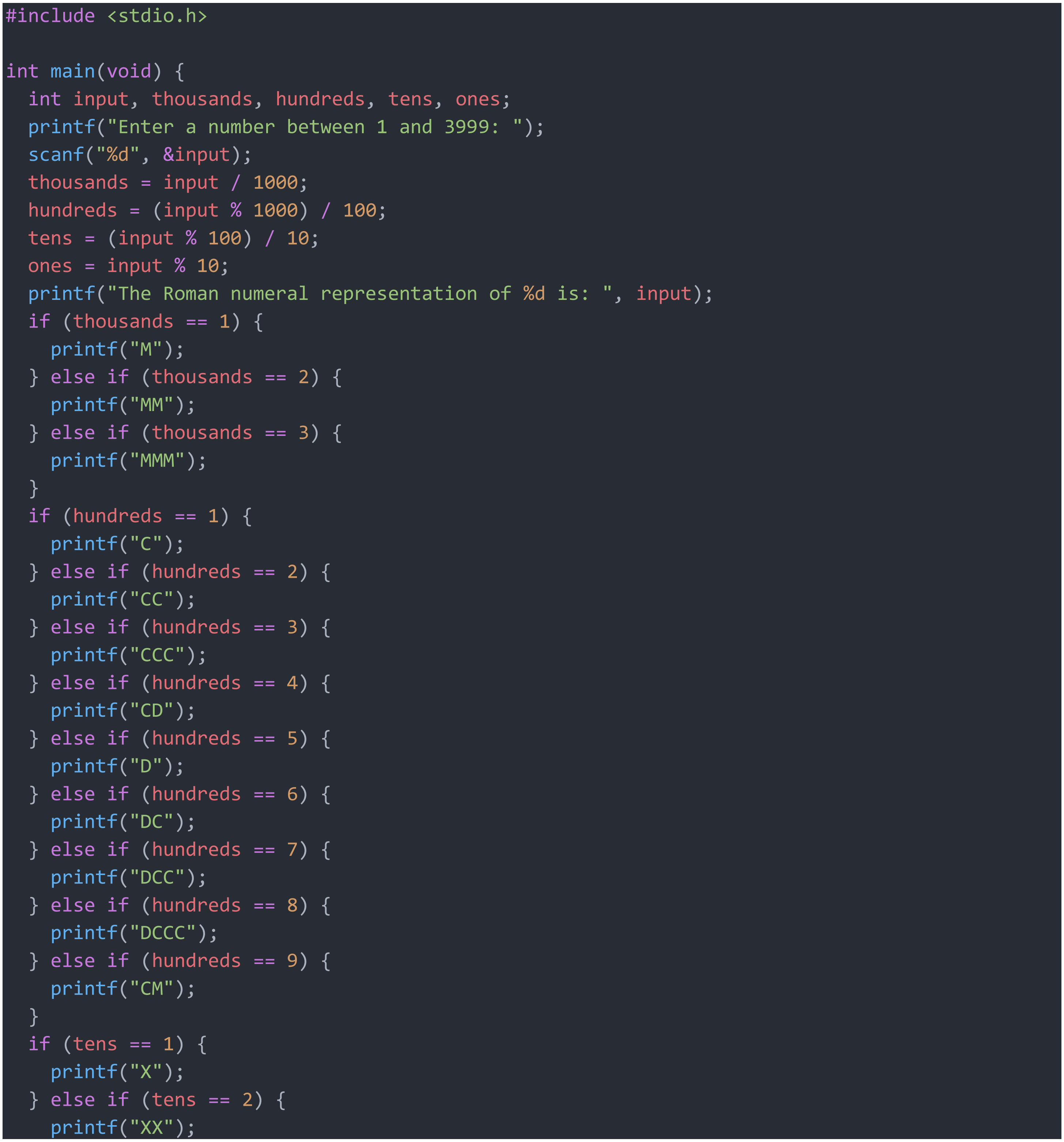
Note: You will need to use the **sqrt()** function from the **math.h** library.



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

int main(void) {
  float x1, y1, x2, y2, distance;
  printf("Enter the coordinates of the first point (x1, y1): ");
  scanf("%f %f", &x1, &y1);
  printf("Enter the coordinates of the second point (x2, y2): ");
  scanf("%f %f", &x2, &y2);
  distance = sqrt(pow((x2 - x1), 2) + pow((y2 - y1), 2));
  printf("The distance between the two points is: %f ", distance);
    return 0;
}
```

Q3: **Roman Numeral Converter:** Write a program that takes an integer input (between 1 and 3999) and converts it to its Roman numeral representation. The program should output the Roman numeral as a string.
Note: You will need to use a series of conditional statements to break down the input number into components that correspond to Roman numeral symbols

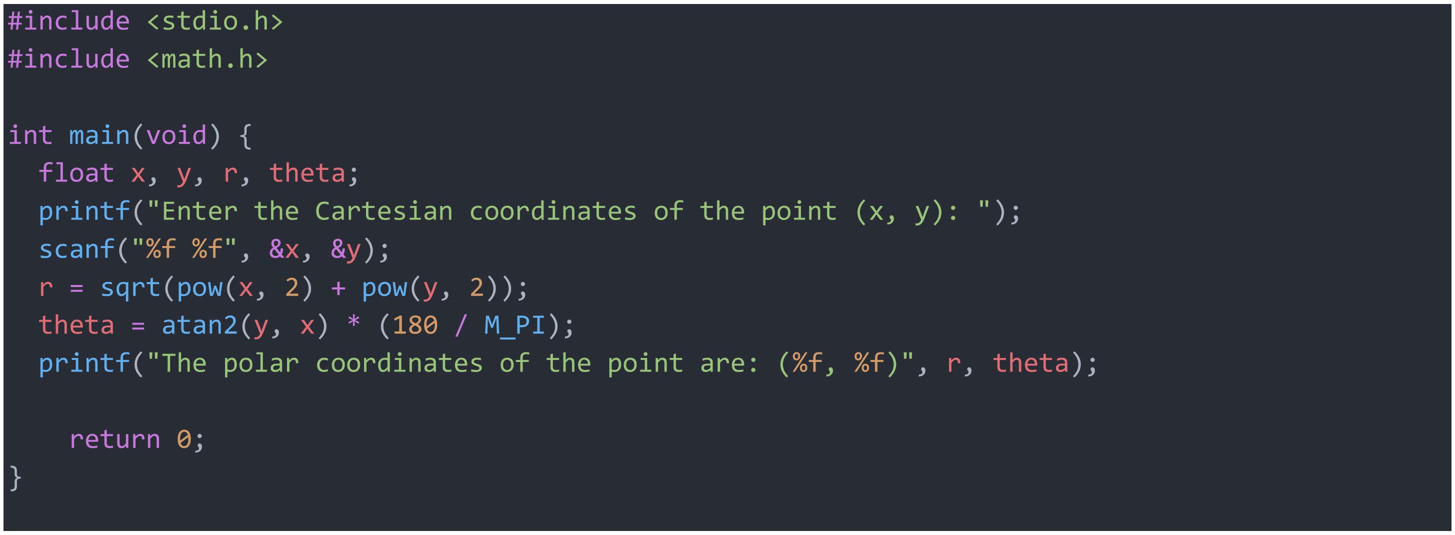
I = 1 , V = 5, X = 10 L = 50 , C = 100, D = 500, M = 1000



```
#include <stdio.h>

int main(void) {
  int input, thousands, hundreds, tens, ones;
  printf("Enter a number between 1 and 3999: ");
  scanf("%d", &input);
  thousands = input / 1000;
  hundreds = (input % 1000) / 100;
  tens = (input % 100) / 10;
  ones = input % 10;
  printf("The Roman numeral representation of %d is: ", input);
  if (thousands == 1) {
    printf("M");
  } else if (thousands == 2) {
    printf("MM");
  } else if (thousands == 3) {
    printf("MMM");
  }
  if (hundreds == 1) {
    printf("C");
  } else if (hundreds == 2) {
    printf("CC");
  } else if (hundreds == 3) {
    printf("CCC");
  } else if (hundreds == 4) {
    printf("CD");
  } else if (hundreds == 5) {
    printf("D");
  } else if (hundreds == 6) {
    printf("DC");
  } else if (hundreds == 7) {
    printf("DCC");
  } else if (hundreds == 8) {
    printf("DCCC");
  } else if (hundreds == 9) {
    printf("CM");
  }
  if (tens == 1) {
    printf("X");
  } else if (tens == 2) {
    printf("XX");
  } else if (tens == 3) {
    printf("XXX");
  } else if (tens == 4) {
    printf("XL");
  } else if (tens == 5) {
    printf("L");
  } else if (tens == 6) {
    printf("LX");
  } else if (tens == 7) {
    printf("LXX");
  } else if (tens == 8) {
    printf("LXXX");
  } else if (tens == 9) {
    printf("XC");
  }
  if (ones == 1) {
    printf("I");
  } else if (ones == 2) {
    printf("II");
  } else if (ones == 3) {
    printf("III");
  } else if (ones == 4) {
    printf("IV");
  } else if (ones == 5) {
    printf("V");
  } else if (ones == 6) {
    printf("VI");
  } else if (ones == 7) {
    printf("VII");
  } else if (ones == 8) {
    printf("VIII");
  } else if (ones == 9) {
    printf("IX");
  }

    return 0;
}
```

Q4 : **Coordinate System Conversion - Cartesian to Polar: a.** Radius: **r = sqrt(x^2 + y^2)**
b. Angle: **θ = atan2(y, x) * (180 / PI)**

Write a program that takes two float inputs (x, y) representing the Cartesian coordinates of a point in a 2D plane. The program should convert these coordinates to polar coordinates (r, θ) and output the radius (r) and angle (θ) in degrees.
Formulas:
Note: You will need to use the **sqrt()**, **atan2()**, and other functions from the **math.h** library. Also, be sure to handle different quadrants and edge cases properly.



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

int main(void) {
  float x, y, r, theta;
  printf("Enter the Cartesian coordinates of the point (x, y): ");
  scanf("%f %f", &x, &y);
  r = sqrt(pow(x, 2) + pow(y, 2));
  theta = atan2(y, x) * (180 / M_PI);
  printf("The polar coordinates of the point are: (%f, %f)", r, theta);

    return 0;
}
```