C programming 1

1.

Create and execute the following program:

```
#include <stdio.h>           // header for standard input/output

int main()                   // definition of the main function
{                            // main function starting bracket
  printf("Hello!\n");
  return 0;
}                            // main function ending bracket
```

Declare two integer variables (a and b) and print results of: sum, subtraction, multiplication and division. The results should be printed in a single line.

2.

Write a program which reads from the keyboard two numbers of double type and prints results of the following operations: multiplication, division, square root, cube root, fourth power, sine and cosine. Print information on what kind of input is expected before the program starts waiting for the input:

```
printf("Input the first number:\n");
scanf("%lf", &a);
```

For example, the printout should be formated as follows:

```
First number     : 12.4
Second number    : 32.22
Multiplication   : 399.5
Division         : 3.85e-001
Square roots     : |3.52| |5.68|
Cube roots       : 2.314589 3.182061
Fourth powers    : 23642 1077711
```

Hint: use tabulator.

3.

Write a program which reads from a keyboard coordinates of three arbitrary points in 2D. Calculate distances between those points. Calculate also an area and a perimeter of a triangle defined by these points. Use Heron formula $P = \sqrt{s(s-a)(s-b)(s-c)}$, where $s = \frac{1}{2}(a+b+c)$, and a, b, c are the lengths of the triangle sides. Format a printout of the results in a clear and readable way.

4.

Modify the program from the previous point by adding code calculating angles of the triangle according to the following formulas:

$$\tan\left(\frac{\gamma}{2}\right) = \sqrt{\frac{(p-a)(p-b)}{p(p-c)}}, \quad \tan\left(\frac{\beta}{2}\right) = \sqrt{\frac{(p-a)(p-c)}{p(p-b)}}, \quad \tan\left(\frac{\alpha}{2}\right) = \sqrt{\frac{(p-b)(p-c)}{p(p-a)}},$$

Print the angles in degrees with precision to three decimal places. Check also if the calculated angles fulfill the condition: $\pi = \alpha + \beta + \gamma$. The number π should be defined using a preprocessor command:

```
#define PI (4.*atan(1))
```

(remember not to use a semicolon at the end of the preprocessor commands). Do you know what exactly this command means? If the condition is not satisfied print the error.

5.

Execute the following code:

```
double a, b;
double w1 = a*a - b*b;
double w2 = (a + b)*(a - b);
printf("w1 = %lf, w2 = %lf\n", w1, w2);
```

This time you can assign values to the variables a and b directly inside the code. Run the program for following values:

```
100000    and 100001;
1000000   and 1000001;
10000000  and 10000001;
100000000 and 100000001;
```

What are the analytical results? Can you continue to increase the input values ad infinity? Repeat the same calculations for floats. Can you explain the results?

6. (optional)

Modify the definition of the π:

```
#define PIa 4.*atan(1) // without brackets
#define PIb 4*atan(1)  // without a dot after 4
```

Compare results of the following operation for all three definitions of π:

```
printf("%lf\n", 3 / PI);
printf("%lf\n", 3 / PIa);
printf("%lf\n", 3 / PIb);
```

Which results are correct? What is a reason for the errors?