COMPUTER PROGRAMMING I

GROUP (   )   LAB (   )

BY

NAME:

Student ID:

DATE:

LAB INSTRUCTION

- LAB Mode: **INDIVIDUAL**.
- Date of Submission: **NOVMBER 10, 2021.**
-

- The font must be **ARIAL** and the size must be **12. JUSTIFY. SPACING 1.5**.
- Use **YOUR NAME, SURNAME**, and **DATE** as a variable in your answer
- Complete the cover sheet and attach it to your lab as (first page).
- Answer lab with own work (***NO PLAGIARISM***).
- Your marks will be deducted in the case of:
  - Late submission.
  - Plagiarism.
  - Different instruction.

[P/S: There may be more than one error in each piece of code]

LAB QUESTIONS

1. Write statements that assign random integers to the variable *n* in the following ranges:

   a) $1 \le n \le 6$

   b) $100 \le n \le 1000$

   c) $0 \le n \le 19$

   d) $1000 \le n \le 2222$

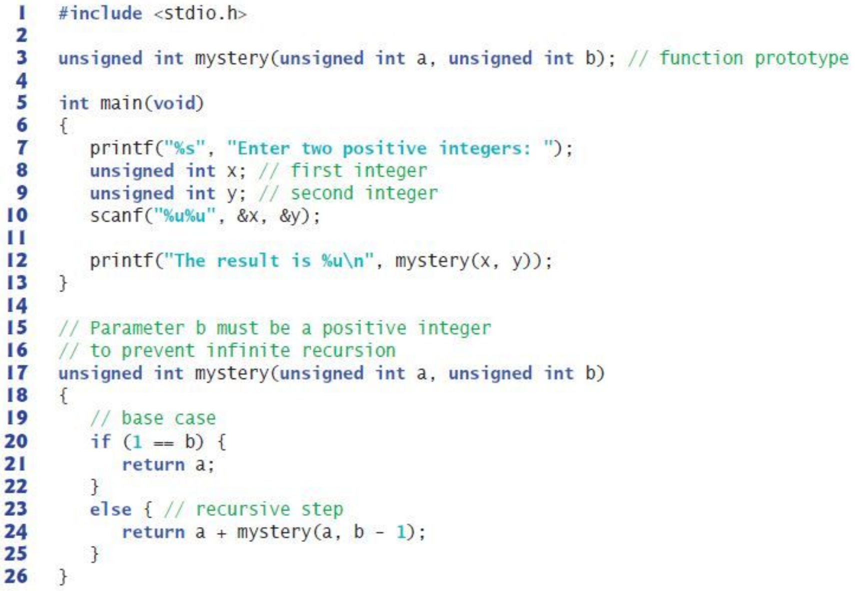
   e) $-1 \le n \le 1$

   f) $-3 \le n \le 11$

2. What does the following program do? What happens if you exchange lines 5 and 6?

```
1. #include <stdio.h>
2. int main(void)
3. {int c;
4. if ((c = getchar()) != EOF) {
5. main();
6. printf("%c", c);
7. }}
```

3. What does the floowing program do?



```
#include <stdio.h>

unsigned int mystery(unsigned int a, unsigned int b); // function prototype

int main(void)
{
   printf("%s", "Enter two positive integers: ");
   unsigned int x; // first integer
   unsigned int y; // second integer
   scanf("%u%u", &x, &y);

   printf("The result is %u\n", mystery(x, y));
}

// Parameter b must be a positive integer
// to prevent infinite recursion
unsigned int mystery(unsigned int a, unsigned int b)
{
   // base case
   if (1 == b) {
      return a;
   }
   else { // recursive step
      return a + mystery(a, b - 1);
   }
}
```

4. Find the error in each of the following program segments and explain how to correct it:

a)

```
double cube(float);
cube(float number)
return number * number * number;
}
```

b)

```
double y = 123.45678;
int x;
x = y;
printf("%f\n", (double) x);
```

c)

```
double square(double number)
double number;
return number * number;
}
```

d)

```
int sum(int n)
{
if (0 == n) {
return 0;
  else {
return n + sum(n);}
```