CSCI1121 **Spring 2018**

Problems to solve

1. Create a C program to read a natural number and to convert the natural number to multiples of 50, 20, 10, and 1.

2. Write a function that takes an integer parameter and returns the result of 1^4 + 2 ^4 + 3^4 + ... + n^4 . Write a C program that reads a positive integer up to1000 and uses the function to display the result of the expression.

3. Write a function that searches for a number in an array of integers. If the number is found, the function should return its position, otherwise -1. Write a program that initializes an array of integers with values that are already sorted in ascending order. The program should read an integer and use the function to display its position in the array.

4. Write a C program to read an IPv4 address and to check its validity.

5. Use a recursive function to calculate the sum and the product of two numbers.

6. Build a calculator that can add, multiply, divide, and find the square root of two numbers.

7. Write a function that takes as parameters two strings and returns a pointer to the longer string. If the strings have the same number of characters, it should return NULL. Write a C program that reads two strings of less than 20 characters and uses the function to display the longer one.

8. Write a C program to evaluate the mathematical function: **`f(x) = 0.5 x^3 - x ln x`**

9. Write a function that takes as parameters an integer and a character and displays the character as many times as the value of the integer. Write a C program that reads an integer and a character and uses the function to display the character.

10. Write a C program that reads a two-digit positive integer and duplicates its digits. For example, if the user enters 12, the program should display 1122.

11. Enter the current PIN code of a mobile phone to change the PIN code of it and the device checks if that code is equal to the one stored in the SIM card. If they are the same, the user is asked to enter the new PIN code once more for verification, and if it is entered correctly, the PIN is stored in the SIM card. Write a C program that simulates this process. Assume that the current code stored in SIM is `5678`.

12. In a C Programming exam, each exam is graded by two GTAs. If the difference of their grades is less than `z`, the final grade is their average. Otherwise, the test is reviewed by a third GTA such as: i) If the grade of the third GTA is equal to the average of the first two grades, that is the final grade. ii) If the grade is less than the minimum of the first two grades, the final grade is `minimal`. iii) Otherwise, the final grade is the average of the grade of the third reviewer and the one of the first two grades closest to it. Write a C program to read the two grades and the difference `z` and to display the final grade according to that method.

13. Create a C program to read an integer and to display a message to indicate whether it is positive or negative. If it is `0`, the program is to display `Zero`. ***Use the conditional operator***.

14. Create a C program to display the area of a square or a circle based on the user's choice. If the user enters `0`, the program is to read the side of the square and to display its area. If the user enters `1`, the program is to read the radius of the circle and to display its area.

15. A machine shop produces small and big metal discs. A small one costs $0.007 and a big one $0.03. For orders of more than $310 or more than 3200 discs in total, a discount of 5% is offered. For orders of more than $700, the discount is 15%. Create a C program to read the number of small and big metal discs ordered and to display the total cost.

16. Create a C program to read the initial population of a country and its annual population growth (as a percentage). Then, the C program reads the number of years and is to display the new population for each year.

17. Create a C program to read an integer and to display the number of its digits and their sum. For example, if the user enters `2345`, the program should display `5` and `14 (2+3+4+5=14)`.

18. Create a C program to read an integer in [0, 255] continuously and displays it in a binary form. For any value out of [0, 255] the program is to terminate.

19. Create a C program to read an integer that corresponds to a number of lines. Assume the user enters `4` so that the C program adds spaces and `'*'` in accordance to the line number, as shown below (three spaces at the left of one `'*'` in the first line, two spaces at the left of two `'*'` in the second line, and so on). When the last line is reached, the program displays in each line one `'*'` less, and the spaces at the right of `'*'`, until one `'*'` is displayed.

```
   *
  **
 ***
****
***
**
*
```

20. Write a C program to read 100 integers and to display the two different higher values.

21. Write a C program to read the grades of 5 students and to store them in an array. Then, the program displays the average, the greatest, and the worst grade, as well as the positions in the array of their first occurrences. The program is an user defined program and requires the user to enter grades within [50, 100].

22. Write a C program to read an integer and to display the digits that appear more than once and the number of their appearances. If no digit appears more than once, the program is to display a message about it.

23. Create a C program to read integers and to store them in a square matrix. Then, the program checks whether the array is a square; that is, the sum of each row, column, and diagonal is the same.

24. Write a C program to use three pointers to read the grades of a student in three exams. If all grades are greater than or equal to 6, the program displays them in ascending order. Otherwise, the program displays their average.

25. Write a C program to read the grades of 5 students, to store them in an array, and to display the best and the worst grade and the positions of their first occurrences in the array. The program is an user defined program, the user needs to enter grades within [50, 100]. Use pointer arithmetic to process the array.

To be continued