## Lab work #3

**Topics**

* Pointers and References
* Functions
* Overloaded Functions
* Template Functions

**Exercises**

1. Write a function called hms\_to\_secs() that takes three int values - for hours, minutes, and seconds - as arguments, and returns the equivalent time in seconds (type long). Create a program that exercises this function by repeatedly obtaining a time value in hours, minutes, and seconds from the user (format 12:59:59), calling the function, and displaying the value of seconds it returns.

1. The body mass index (BMI) is a measure used to evaluate how far the weight of a normal person deviates from the "ideal" weight. It is given by weight/height², with the weight measured in kilograms and the height measured in meters. Create a function, BMI(weight, height), to calculate the body mass index. Write another function to classify the BMI according to the following table. Use these functions in a program that requests weight and height to the user and prints the BMI and the corresponding category.

| BMI | < 18.5 | [18.5,25[ | [25,30[ | >= 30 |
| --- | --- | --- | --- | --- |
| Category | Underweight | Normal | Overweight | Obese |

1. Start with a program that allows the user to input a number of integers, and then stores them into an int array. You can ask first to the user the number of elements. Use dynamic memory allocation for the array.

Write a function called maxint() that goes through the array, element by element, looking for the largest one. The function should take as arguments the address of the array and the number of elements in it, and return the index number of the largest element. The program should call this function and then display the largest element and its index number.

1. Consider the following tasks regarding string manipulation. Each task should be implemented using a function following the provided declaration.
2. calculate the total number of alphabetic characters in a string

int count\_alpha (const string &s)

1. calculate the number of lowercase and uppercase characters of a string

int count\_lowers (const string &s)

int count\_uppers (const string &s1)

1. indicate (print) whether the two strings are equal (or what is the greater one, i.e., “comes after the other”)

void equals\_or\_greater (const string &s1, const string &s2)

1. convert a string to the corresponding one with lowercase characters

string tolower (const string &s1)

1. replace in a string all occurrences of multiple spaces followed by a single space

string replace (const string &s1)

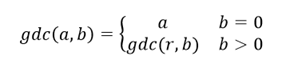
1. capitalize the first letter of each word the string

string capitalize (const string &s1)

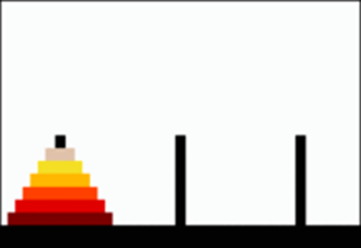
1. indicate whether a string is a palindrome or not. consider only alphabetic characters

bool ispalindrome (const string &s1)

1. Recall exercise 6 from Lab work #2.
   1. Create a function that accepts as parameters the paragraph (string) and the keyword (string), returning the number of occurrences of the keyword in the paragraph. Make keyword an optional parameter, using as default value an empty string, and, in this case, the function should count the total number of words in the paragraph (consider the most common symbols as separators).
   2. Implement a program that starts by getting from the user a keyword and then gets paragraphs in an iterative way (a paragraph finishes with the symbol '$'). The user has to write a paragraph with the word "End" to finish the program. For each paragraph introduced, the program should count the total number of words in the paragraph and the number of occurrences of the keyword.
2. Write a function countdown(N) which prints a countdown starting from a positive number N. Test it in a program which requests the value of N to the user. Do not use loops.
3. Write a function which determines how many digits has an integer positive number. Use it in a program which requests that value to the user. Do not use loops.
4. Write a function that returns a string corresponding to the binary representation of a positive integer number. Use it in a program which requests that value to the user. Do not use loops.
5. Write a function which calculates the greatest common divisor between two integer positive numbers using the Euclidean algorithm, given by the following definition:



where *r* is the remainder of the division of *a* by *b*. Use it in a program which requests that value to the user. Provide two different implementations.

1. The Fibonacci sequence is a sequence of integers in which each element is equal to the sum of the two previous ones: . The first values are defined as e . Write a function, Fibonacci(n), to calculate the nth Fibonacci number. Which is the value of ? Provide two different implementations.
2. Raising a number n to a power p is the same as multiplying n by itself p times. Write a function called power()that takes a double value for n and an int value for p, and returns the result as a double value. Use a default argument of 2 for p, so that if this argument is omitted, the number n will be squared. Write a program that gets values from the user to test this function.
3. Start with the power() function of the previous exercise, which works only with type double. Create a series of overloaded functions with the same name that, in addition to double, also work with types char, short, int, long, and float. Write a program that exercises these overloaded functions with all argument types.
4. Create and test a new function based on templates that can replace all the functions of the previous exercise.
5. Create and test a new function based on templates to compare if the two input arguments are equal. Try this function at least with int, bool and string types.
6. Consider the Towers of Hanoi puzzle. The aim is to move tower from A to C, using B temporarily, obeying to the following rules:

* Move only one disk at a time;
* No disk may be put on top of a smaller disk.