**Programming Languages 1**
**Practicing lesson A**

**Exercises:**

1. Write a C function, which has a character parameter. Assuming that the parameter is always between ’0’ - ’9’, or ’A’-’F’ it gives back the decimal value of the given hexadecimal digit. For example: ’8’ → 8; ’A’ → 10; ’f’ → 15.

2. Write a C function, which has a string parameter. If the parameter formally can be a hexadecimal value, the function gives back the (decimal) integer representation of this not negative hexadecimal number. Use the function written in the first task. For example: ”100” → 256; ”F3” → 243.

3. Write a C function which has an integer parameter. Assuming that the parameter is always between 0 and 15, the function has to return with a character (the hexadecimal form of the parameter). For example: 5 → ’5’; 10 → ’A’; 14 → ’E’.

4. Write a C procedure, which has an (unsigned) integer parameter and it prints out the same value to the screen in hexadecimal form. Use the function written in the previous task. For example: 11 → ”B”; 193 → ”C1”; 512 → ”200”.

5. Write a C procedure, which has no parameter, and it reads two hexadecimal number from keyboard and prints out the sum of the values in decimal form. Use the previous subroutines. For example: ”F1”, ”E” → 255; ”100”, ”23” → 291.

6. Write a C procedure, which has no parameter, and it reads two (not negative) decimal number from the user and prints the sum of them in hexadecimal form. Use the earlier subroutines. For example: 10, 5 → ”F”; 100, 154 → ”FE”.

7. In the main program function create a loop in which the following menu items are printed:
   ”1) Decimal input”
   ”2) Hexadecimal input”
   ”0) Exit”
   Read the choice of the user as a character. If it is the ’1’, call the procedure of task 6. If it is the ’2’, call the procedure of task 5. If it is the ’0’, leave the loop. (For any other character don’t do anything just repeat the loop printing the menu again.)