Programming exercises

1   Exercise

Write a program that calculates your age. Program must print:

```
I was born on 7.5.1992
Today is 13.1.2012
I am 20 years old.
```

Define seven integer variables: three variables for your birthday to hold year, month and day and other three to hold today's date. Assign values to these variables when you define variables. Define one variable to hold your age. First print your birthday and today's date then perform calculations and finally print your age. You don't need to take months into account in age calculation.

2   Exercise

Write a program that uses floating point variables to hold your salary, tax percentage, tax, net income and raise. The program then calculates your net income before and after your salary is raised. Assign initial values to the variables when you define variables.

Program must print:

```
My salary is 1238.75 euros.
My tax percentage is 17.0 percent.
I have to pay 210.59 euros tax.
I have 1028.16 euros to spend or save.
My boss raises my salary by 150 euros.
My new salary is 1388.75 euros.
After taxes I have 1152.66 euros.
```

3   Exercise

Write a program that asks you to enter price of a bus ticket. Program then asks you how much money you have. Program compares ticket price with the amount money you have and prints text which tells you if you have enough money or not and how much you have left after buying a ticket or how much more you need to buy a ticket.

Program must print:

```
Enter price of bus ticket: 4.5
How much money do you have: 9.55
You have enough money for a bus ticket.
You have 5.05 euros left.
```

Another example:

```
Enter price of bus ticket: 4.5
How much money do you have: 3.10
You do not have enough money for a bus ticket.
You need 1.40 euros more to buy a ticket.
```

4   Exercise

Write a program that asks user to enter todays date (date, month, year). Program then prints tomorrow's date and takes into account that tomorrow could be on a different month or year. You can assume that all months have 30 days.

```
Enter today's date (d.m.y): 19.1.2012
Tomorrow is: 20.1.2012

Enter today's date (d.m.y): 30.5.2015
Tomorrow is: 1.6.2012

Enter today's date (d.m.y): 30.12.2011
Tomorrow is: 1.1.2012
```

More challenging version for advanced programmers:
Take into account the real length of months. Lengths of months from 1 to 12 are: 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31

5 Exercise

Write a program that asks user to his/her height in centimeters and weight in kilograms. The program then calculates users BMI (Body Mass Index) and prints it. Then program prints one of the following:

BMI is less than 18.5 program prints "According to BMI you have underweight"

BMI is greater than or equal to 18.5 and less than 25: "According to BMI you have normal weight"

BMI is greater than or equal to 25:"According to BMI you have overweight"

BMI is calculated using the following formula: BMI = mass in kilograms / (height in meters * height in meters). For more information see: en.wikipedia.org/wiki/**Body_mass_index**

```
Enter your weight in kilograms: 65
How tall you are in centimeters: 170
Your BMI is 22.5.
According to BMI you have normal weight.
```

Remember to test your program with different values to ensure that it works properly. Test until you have seen all three messages.

6 Exercise

Write a program that asks user to enter current time in 24 hour format and asks how long you want to sleep. Program then calculates your wake up time and prints it.

```
Enter current time (hh:mm): 21:56
How long do you want to sleep (h:mm): 8:30

If you go to bed now you must wake up at 6:26.
```

7   Exercise

Write a program that defines an array of 12 integers. Then program asks user to enter the first number in the array. When user has entered number program writes a sequence of consecutive numbers into the array starting from the number that user entered. Finally program prints the numbers in the array. Use two loops. One loop writes values to the array and the other prints the values.

```
Enter starting number:   42
The array values are:
42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53,
```

8   Exercise

Write a program that defines an array of 10 integers. Program sets an initial value of 0 to all array elements. Then program enters a loop which asks user to enter new values. New value must be a positive number. If a negative number is entered the program displays an error message and asks user to enter value again.

Define a constant for the array size and use the constant so that you can easily change the number of elements in the array.

```
Enter value for element 0: 3
Enter value for element 1: -9
Value must be positive!
Enter value for element 1: 9
Enter value for element 2: 45
Enter value for element 3: 0
   and so on…
Enter value for element 9: 12
The sum of 10 numbers in your array is: 125
```

9   Exercise

Write a program that defines an array of 5 floats. Program then asks user to enter five numbers. When user has entered numbers program finds the smallest number that user has entered and prints the number with 3 decimals.

Hint: use a separate variable for smallest number. Before starting search initialize smallest number to the first number in the array. Then compare number against other values in the array (use a loop) and replace the previous smallest value with a new one if necessary.

```
Enter five numbers:
Enter number 1: 2.3
Enter number 2: -1.3
Enter number 3: 5
Enter number 4: 59.23
Enter number 5: 43.589
The smallest number that you entered was: -1.300
```

10 Exercise

Write a program that prints a menu and asks user to select an action. Menu contains four choices: enter operands, multiply operands, add operands and quit. If user makes an invalid selection program does nothing and prints the menu again.

Use a loop to repeat menu printing and use switch-case statement inside the loop to determine actions to perform.

```
Select operation:
1) enter operands
2) multiply operands
3) add operands
4) quit
1
Enter operand 1: 4
Enter operand 2: 7

Select operation:
1) enter operands
2) multiply operands
3) add operands
4) quit
2
Result = 28

Select operation:
1) enter operands
2) multiply operands
3) add operands
4) quit
5          (then program quits)
```

11 Exercise

Write a program that asks user to enter the amount of money he wants to borrow. Then program asks user to enter interest rate. Finally program asks user to enter payments. If user enters a negative number program displays an error message and asks user to enter number again. Program stops when the user does not owe any money.

When user has entered a payment the program prints the amount of money he still owes.
The interest that must be paid is the amount of loan **before** the payment multiplied by the interest rate.
Note interest is taken from the paid amount first and what is left after that goes to the balance.

```
How much you want to borrow: 150
Enter interest rate (percent): 7.5
Enter payment: 20
You still owe 141.25 euros.
Enter payment: 50
You still owe 101.84 euros.
<and so on>
```

12 Exercise

Write a function that takes two integer parameters and returns a float. The function calculates currency exchange rate. The first parameter is number of euros and the second parameter is number of dollars that you get for the number of euros (the first parameter).

Write a main function that asks user to enter number of euros and number of dollars for the exchange rate calculation. Then program prints the exchange rate with 4 decimal. Then program asks how many euros user wants to convert and then prints amount of dollars user would get.

```
Enter amount of euros: 100
Enter amount of dollars: 124
Exchange rate: 1.2400
How many euros do want to exchange: 19
You'll get 23.56 dollars.
```

13 Exercise

Write a function which takes one integer as a parameter and does not return a value. The function prints number of hash signs (#) specified by the parameter on a single line.

Write a main function that prints 10 random lines of hash signs using the function above. The number of hash sings on each line must be in the range of 1 to 20. You can get random numbers by calling rand-function. In order to use rand() you must include stdlib.h at the beginning of your source code. Time.h is required to inititalize random number generator seed number. Call srand once at the beginning of your program (see examples below).

```
#include <stdlib.h>
#include <time.h>
```

Rand() takes no parameters and returns an integer in the range of 0 to RAND_MAX. An easy way to limit the range of numbers is to use modulus operator and offset if required.

```
srand(time(NULL));          // this initializes generator seed value
v1 = rand() % 10;           // v1 in the range 0 to 9
v2 = rand() % 10 + 1;       // v2 in the range 1 to 10
```

```
            Random lines:
            ####
            ##########
            ##
            #####################
            ########
            ##################
            etc.
```

14 Exercise

Write a function which takes two integer parameters and returns an integer. The function asks user to enter a number in a range that is specified by the function parameters. The first parameter is the lower limit and the second parameter is the upper limit. If user enters an invalid value function prints an error message and asks user to enter the number again. The function does not return before user has entered a valid value.

Write a main function that uses the function to read one number in range of 10 to 20 and another one in the range of 0 to 5. In the end program prints the numbers and their product.

```
            Enter a number [10 - 20]: 34
            Invalid value! number must be between 10 and 20.
            Enter a number [10 - 20]: 15
            Enter a number [0 - 5]: 3

            3 multiplied by 15 is 45.
```

15 Exercise

Write a function which takes two pointers to float as a parameter and returns no value. The function finds largest of the two numbers and scales the numbers in to range of 0 -. 1.0 so that largest number is set to one and smallest is scaled with equal scale (new value = smallest/largest). This function MUST NOT print anything!

Write a main function that asks user to enter two positive numbers and then uses the function to scale the numbers. Finally the program prints the numbers.

```
Enter 1. number: -1.3
Number must be positive!
Enter 1. number: 1.89
Enter 2. number: 9.46
Scaled values are:
1: 0.19979
2: 1.00000
```
Another example
```
Enter 1. number: 11.29
Enter 2. number: 9.46
Scaled values are:
1: 1.00000
2: 0.83791
```

16 Exercise

Write a function which takes two pointers to float as a parameter and returns a float. The function calculates the average value of the numbers that pointers point to. Then function changes the numbers so that their values will be deviation from the average (value – average). Function returns the average value. This function MUST NOT print anything!

Write a main function that asks user to enter two numbers and then uses the function to calculate the average and the deviations and prints them.

```
Enter 1. number: 14.89
Enter 2. number: 19.36
Average value is:17.125
1. deviation: -2,235
2. deviation: 2,235
```

17 Exercise

Write a function which takes three parameters and returns no value. Parameters are a float, a pointer to a float and an integer. First parameter is currency exchange rate. The second parameter is a pointer to array of currencies (you can treat the pointer just like an array in your function). The third parameter is the array size. Function converts all values in the array using the exchange rate. This function MUST NOT print anything!

Write a main function which defines an array of five floats. Program asks user to enter exchange rate and then five amounts of money to convert. Program uses the function to do the conversion and prints the values. Values must be printed so that **decimal points are aligned** to the same column.

```
Enter exchange rate: 1.135
Enter 1. amount: 23.14
Enter 2. amount: 100.00
Enter 3. amount: 25.00
Enter 4. amount: 33.75
Enter 5. amount: 1000.00
Converted amounts are:
1:    26.26
2:   113.50
3:    28.38
4:    38.31
5: 1135.00
```

18 Exercise

Write a program which defines three strings. One for your first name, one for your last name and one for whole name (first & last combined). Program asks you to enter your first name and then your last name. Program then checks that combined length of both names is not larger than the third string and will copy the strings only if they fit in the third string. Put last name at the beginning of the combined name and remember to add space between names. Note that you have to take the added space into account when calculating the length of combined strings. Use strcpy and strcat. Finally the program prints the combined string and its length.

```
Enter your first name: Keijo
Enter your last name: Länsikunnas

Hello, Länsikunnas Keijo
Length of name = 17
```

19 Exercise

Write a function which takes three parameters: a string that contains question that is displayed to the user, a string to hold the answer and an integer which tells the maximum length of the answer string. Function displays the question to user and reads user answer into the second string. Function must ensure that maximum number of characters is not exceeded (use fgets and remove the trailing line feed if necessary). Function does not return a value.

Write a main function that uses this function ask your name, address and postal code. Finally program prints your full address.

```
Enter your name: Keijo Länsikunnas
Enter your address: PL 4070
Enter postal code: 00079 Metropolia

Your address is:
Keijo Länsikunnas
PL 4070
00079 Metropolia
```

20 Exercise

Define a structure to hold month names and numbers. The structure must have two members: a string and an integer.

Write a function that takes the structure as an argument and returns no value. Function prints month number and name of the month.

Write a program that asks user to enter name of month and month number of first and last month of the year and stores them in separate structure variables. Program must ensure that month number is in range of 1 to 12. Then program calls the printing function to print the months that user has entered.

```
Enter information about first month of year:
Enter month number: 1
Enter name of month number 1: Tammikuu
Enter month number: 13
Month number must be in range 1-12!
Enter information about last month of year:
Enter month number: 12
Enter name of month number 12: Joulukuu

You entered the following months:
1.  Tammikuu
12. Joulukuu
```

21 Exercise

Define a structure to hold student information. Structure must contain: first name, last name and amount of course credits.

Write a program which defines an array of students. Program asks user to enter name and amount of course credits for each student. When all information is entered program prints a list of students. First name, last name and credits **must be printed in aligned columns**.

Your array of students must contain at least 5 students!

```
Enter first name of
student 1: Jack
Enter last name of
student 1: Daniels
Enter number of credits of
student 1: 55
Enter first name of
student 2: John
.
.
.
Enter number of credits of
student 5: 123

Jack      Daniels        55
John      Woo            87
Richard   Westwood      197
Daniel    Knizia         53
Clyde     Dillinger      32
```