

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 calculates circumference of a circle and prints the circumference. Use floating point variables to hold radius, circumference and pi. Assign values to variables before calculations. ($c = 2 * \pi * r$, $\pi = 3.1416$)

Program must print:

```
Radius of circle is 12.00 cm.
Circumference is 75.40 cm.
```

3 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.
```

4 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.
```

5 Exercise

Write a program that asks user to enter today's 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

6 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: $BMI = \text{mass in kilograms} / (\text{height in meters} * \text{height in meters})$. 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.

7 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.
```

8 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,
```

9 Exercise

Write a program that defines an array of 7 floats. Program then asks user to enter seven numbers. When user has entered numbers program find 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 smallest value with new one if necessary.

```
Enter seven numbers:
Enter number 1: 2.3
Enter number 2: -1.3
Enter number 3: 5
    and so on...
Enter number 7: 43.589
The smallest number that you entered was: -1.300
```

10 Exercise

Write a program that defines an array of 20 integers. Program sets an initial value of 0 to all array elements. Then program enters a loop which asks user to enter number of element to change. When user has entered number the program checks that number is within array bounds. If entered number is not valid program displays an error message. If element number is valid program asks user to enter value for the array element.

When user enters 99 as the element number the program prints sum of all number in the array.

Note that program must not display an error message when user enters 99 (stop indicator). Define constant for array size and use constant so that you can easily change the number of elements in the array.

```
Array element number must be between 0 and 19.
Enter 99 to stop.
Enter nr. of array element to change: 2
Enter value for element 2: 3
Enter nr. of array element to change: 21
Error!
Element nr. must be between 0 and 19!
Enter nr. of array element to change: 17
Enter value for element 17: 9
Enter nr. of array element to change: 5
Enter value for element 5: 54
    and so on...
Enter nr. of array element to change: 99
The sum of numbers is: 125
```

11 Exercise

Write a program that prints a menu and asks user to select an action. Menu contains five choices: enter operand 1, enter operand 2, multiply operands, add operands, print result 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 operand 1
2) enter operand 2
3) multiply operands
4) add operands
5) quit

1
Enter operand 1: 4

Select operation:
1) enter operand 1
2) enter operand 2
3) multiply operands
4) add operands
5) quit

2
Enter operand 2: 7

Select operation:
1) enter operand 1
2) enter operand 2
3) multiply operands
4) add operands
5) quit

3
Result = 28

Select operation:
1) enter operand 1
2) enter operand 2
3) multiply operands
4) add operands
5) quit

5
Quit. (then program stops)
```

12 Exercise

Write a program that asks user to enter the amount of money he wants to lend. Then program asks user to enter interest rate. Finally program asks user to enter 3 down payments. If user enters a negative number program displays an error message and asks user to enter number again.

When user has entered all number the program prints a table of total payments with two decimals and amount of money you owe at the end. The interest you have to pay is the amount of loan before the payment multiplied by the interest rate.

```
How much you want to lend: 150
Enter interest rate (percent): 7.5
Enter 1. down payment: 20
Enter 2. down payment: 15
Enter 3. down payment: 28
1. payment: 31.25
2. payment: 24.75
3. payment: 36.63
You still owe 87 euros.
```

13 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.
```

14 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 signs on each line must be in the range of 2 to 25. 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 initialize 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.
```

15 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. First parameter is the lower limit and second parameter is the upper limit. If user enters invalid value function prints an error message and asks user to enter number again. Function returns only when 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 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.
```

16 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 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
```

17 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 average and deviation 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
```


18 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
```

19 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 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
```

20 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
```

21 Exercise

Write a program which defines an array of strings to hold month names. Program then displays numbers from 1 to 12 and asks user to enter name for each month. Finally program prints a list of month numbers and their names. Remember to ensure that user does not enter too long month names!

```
Enter name of month number 1: Tammikuu
Enter name of month number 2: Helmikuu
.
. program asks each month
.
Enter name of month number 12: Joulukuu
Each year has the following months:
1. Tammikuu
2. Helmikuu
.
.
.
12. Joulukuu
```

22 Exercise

Write a function which asks user to enter numbers in to 3 x 3 magic square. 3 x 3 magic square has numbers 1 – 9 distributed in squares so that sum of numbers on each row or column is 15. The function takes 3 x 3 array as a parameter and does not return a value. Function must initialize the array to zero at the beginning of the function. Then function asks user to enter values for each element and checks that entered value is between 1 and 9 and appears only once in the array (for duplicates use the function described below). If value is invalid user is asked again until he/she enters a valid value. The function doesn't need to check if sum of rows or columns is correct.

Write a function which takes 3 x 3 array and integer as parameters and returns an integer. Function returns zero if the integer parameter does not appear in the array and non-zero if the integer is already in the array. Use this function to prevent user from entering duplicates in the array.

Write a main program that uses the functions to ask magic square numbers and then prints the numbers into square.

http://en.wikipedia.org/wiki/Magic_square

```
Enter numbers of 3 x 3 magic square:
Enter row 1, element 1: 3
Enter row 1, element 2: 4
Enter row 1, element 3: 5
Enter row 2, element 1: 3
Duplicate values are not allowed! Enter another number.
Enter row 2, element 1: 1
Enter row 2, element 2: 7
Enter row 2, element 3: 0
Invalid value! Number must be between 1 and 9.
Enter row 2, element 3: 8
Enter row 3, element 1: 9
Enter row 3, element 2: 2
Enter row 3, element 3: 6
Your square is:
3 4 5
1 7 8
9 2 6
```

23 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 and stores them in the structure. Program must ensure that month number is in range of 1 to 12. Then program calls the printing function to print the structure that user has entered.

24 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

25 Exercise

Define a structure to hold share information. Structure must contain: name of the company, price of single share and amount of last dividend.

Write a function which takes pointer to structure as a parameter and returns an integer. Function asks user to enter information of one company.

If user enters empty company name function returns non-zero value and otherwise function returns zero. Function asks user to enter share information and checks that values are valid. Price and dividend must be positive and name of the company must fit in the structure.

Write a function which takes pointer to structure and an integer as parameters and return a float. Integer tells the number of structures. Function calculates and returns average dividend.

Write a program which defines array of structures and uses the functions to read share information. Program must use return value of reading function to determine when to stop. Program then calculates and prints average dividend. Then program prints two lists. First list of companies that paid higher dividend than average and then list companies that paid lower (or equal) dividend than average.

```
Enter company name: Nokia
Enter share price: 9.35
Enter dividend: 0.24
Enter company name: Kone
Enter share price: 3.51
Enter dividend: 0.36
Enter company name: Google
Enter share price: 19.10
Enter dividend: 0.13
Enter company name: Kesko
Enter share price: 4.15
Enter dividend: 0.28
Enter company name:
```

```
Average dividend is: 0,2525
Over average:
Kone          3.51      0.36
Kesko         4.15      0.28
Under average:
Nokia         9.35      0.24
Google       19.10      0.13
```

26 Exercise

Write a program that asks user to enter name of the file. Program opens the file for writing in text mode. If an error occurs program prints an error message and terminates.

Program asks user to enter text. Program converts the first letter to upper case and all other letters to lower case and writes line to file. Program closes the file and exits when user enters a single dot at the beginning of the line.

```
Enter file name: letter.txt
Enter text. Enter single dot to quit.
tHis is a test text.
Marriage is noT a word it is a sentence.
I can enter numb3rs if I want.
two plus 5 makes 10.
.
```

Program writes to file:

```
This is a test text.
Marriage is not a word it is a sentence.
I can enter numb3rs if i want.
Two plus 5 makes 10.
```

27 Exercise

Write a program which asks user to enter the name of a (text) file. Program then reads floating point numbers from file and calculates and prints the average of the numbers and amount of numbers read from the file.

```
Enter file name: numbers.txt
Read 5 numbers.
Average value is 13.42.
```

The file in the example above contained five numbers. Use notepad to create your own test file.

28 Exercise

Write a program which asks user to enter the name of a text file. Program opens the file. If opening fails program prints an error message and terminates.

Program reads numbers from a file. Each line contains two numbers. Program reads lines one by one and checks that input is valid (each line must contain two integers). If line is not properly formatted the program prints an error message. For each properly formatted line the program prints numbers and their multiplication.

Hint: Read line with `fgets` and use `sscanf` to read the integers. Remember to check `sscanf` return value.

```
Enter file name: numbers.txt
Opened numbers.txt
7 * 2 = 14
43 * 9 = 387
7 * 7 = 49
Error! Line number 4 is invalid.
2 * 2 = 4
```

The file of the example above contained following lines:

```
7 2
43 9
7 7
42 aq
2 2
```

29 Exercise

Write a program that defines a structure to hold items in warehouse. Structure must contain name of the item, price (float), number of items in the warehouse and number of outstanding orders.

Write a program that defines array of structures and reads data from a file into array.

Program asks user to enter name of the file and if the file exists reads items from the file into an array of structures. If the file does not exist then a new file is created. File must be opened in binary mode. Program must keep count of items in the array. Program then prints list of items and asks user to enter more items. User may quit by entering a single dot when item name is asked. When user quits program writes all items to disk. Both new and old (if any) are written to disk.

```
Enter file name: warehouse.dat
Read 3 items from file.
Items:           Price      Qty      Orders
Motherboard     123.45      8        5
Hard disk       240.00     20        3
Power supply    55.50      13       10

Enter name of item (single dot quits): Processor
Enter price: 144.00
Enter quantity: 10
Enter outstanding orders: 0

Enter name of item (single dot quits): Cooling fan
Enter price: 22.30
Enter quantity: 5
Enter outstanding orders: 10

Enter name of item (single dot quits): .
Wrote 5 items to disk.
```

30 Exercise

Write a program which reads structures of exercise 29 into a dynamically allocated array and prints the data that was read from the file.

Hint: First open the file then read (and discard) all structures to count how many structures there are in the file. Then rewind back to the beginning of the file, allocate memory and finally read the data.