In each exercise make your source code and output readable.

Exercise 1. Write a program that outputs the sentence "Learn to walk before you run." Repeat the sentence n times, where n is given by the user. Apply the for loop.

Exercise 2. Write a program that outputs the sentence "Learn to walk before you run.".
Repeat the sentence as many times as the user wants, ask the user after displaying the sentence whether to display the sentence once again. Apply the do while loop.

Exercise 3. Repeat computations specified in exercise 1 from lab2exericses.pdf (see below) as many times as the user wants, ask the user after computations whether to do it once again.

Write a program that asks the user to input two integers denoting quantity and price per item, then calculates and prints the total expenses. If the expense is more than 5000, discount of 10% is offered.
The user-computer interaction could look as follows:

```
Enter quantity: 14
Enter price: 300
Total expense: 4200
```

```
Enter quantity: 1001
Enter price: 5
Total expense with discount: 4504.5
```

Hint: Samples of exercise 1 one can find in the Programs menu in the team.

Exercise 4. Write a program that asks the user to enter the amount of their monthly budget. Repeat reading of an amount spent until non-positive value is given. A loop should prompt the user to enter each of the users expenses for the month and keep a running total. When the loop finishes, the program should display the amount that the user is over or under budget. In computations of running total consider only positive values of expenses.
Sample run #1:

```
Enter amount budgeted for the month: 1250.00
Enter an amount spent (non-positive value to end): 145.87
Enter an amount spent (non-positive value to end): 15
Enter an amount spent (non-positive value to end): 198
Enter an amount spent (non-positive value to end): 345.99
Enter an amount spent (non-positive value to end): 0
Budgeted: 1250.00
Spent: 704.86
You are 545.14 under budget. WELL DONE!
```

Sample run #2:

```
Enter amount budgeted for the month: 759
Enter an amount spent (non-positive value to end): 127.43
Enter an amount spent (non-positive value to end): 150
Enter an amount spent (non-positive value to end): 600
Enter an amount spent (non-positive value to end): -100
Budgeted: 759.00
Spent: 877.43
You are 118.43 over budget. PLAN BETTER NEXT TIME!
```

Exercise 5. A large company allocates its employees leave based on the number of hours worked in a week. Each employee gets two hours of leave for each week worked plus 10% of hours worked. Develop a C++ program that uses a loop to input each employee's hours worked for last week and calculates and displays the number of hours of leave accrued by the employee. Process each employee's figures at a time. Repeat calculations until the value -1 for hours worked in a week is given. Sample run:

```
Enter number of hours worked (-1 to end): 10.5
Accrued leave: 3.05 hours

Enter number of hours worked (-1 to end): 45
Accrued leave: 6.50 hours

Enter number of hours worked (-1 to end): 30
Accrued leave: 5.00 hours

Enter number of hours worked (-1 to end): -1
The end of computations.
```

MODIFICATION. Round the number of hours of leave accrued by the employee to the nearest integer. Use round function from cmath library.

Exercise 6. Write a program that that reads real numbers from the input until the user decides to stop the input. If the real number is positive, then program displays square root of the number otherwise the program displays square of the number. In addition, the program counts how many real numbers have been given and how many of them are positive and how many are negative. In a program use pow and sqrt functions from cmath library.

Interaction with the program might look like this:

```
Enter real numbers.
1. Give a number: 3
Square root: 1.73205
Continue (y/n)? y
2. Give a number: -4
Square: 16
Continue (y/n)? y
3. Give a number: -2
Square: 4
Continue (y/n)? y
4. Give a number: 5
Square root: 2.23607
Continue (y/n)? y
5. Give a number: 0
Continue (y/n)? n
5 numbers are given (2 positive and 2 negative)
```

MODIFICATION 1. Additionally, program calculates and displays two sums: the sum of positive numbers and the sum of negative numbers.

MODIFICATION 2. Additionally, program calculates and displays two arithmetic averages: the arithmetic average of positive numbers and the arithmetic average of negative numbers.