

Prerequisites:

- Basic knowledge of using computer and text editor
- Knowledge how to create a program
- Knowledge how to create a function
- Knowledge of different types
- Knowledge about conditional statements
- Knowledge about loops (3 types)
- Knowledge about 1D and 2D arrays

Aims:

In this laboratory student will learn:

- how to use variables
- about range of variables

1 Theoretical background

1.1 Local and global variables

Let's say you have to add two variables $a=3$ and $b=4$. If you already know these values it is even not required to define a and b . You can just use expression $3+4$ and no additional memory space is required. However, in case when you do not know what exact values will be used for these variables (a and b) till program will be started, then you have to reserve a special space in memory for these variables. Such a space in memory could be reserved all the time your program will be running or only in moment when your function will be executed. In the first case you can use global variables. In the second case you can use local variables.

```
int x = 6, y = 7;

int Add()
{
    int a = 3, b = 4;

    return a + b++;
}

int main()
{
    printf("3 + 4 = %i", Add());

    return 0;
}
```

In the above source code there are four variables, two local (a and b) and two global (x and y). Variables a and b have only local range (inside `Add` function). In turn, variables x and y have global range (in all places in your program).

1.2 Static variables

In the above example variables a and b have always values 3 and 4 in the moment of returning result in the `Add()` function, respectively. Variable b is increasing, however, it has no effect in program.

So let's modify body of the `Add()` function:

```
int Add()
{
    static int a = 3, b = 4;

    return a + b++;
}
```

Right now the value of variable b is different each time this function will be called. In fact, the last used value of variable b is remembered and later it is used as a starting value.

2 Exercises

1. Write a source code for example explained above.
2. Use some loop to run function `Add()` 10 times. Take a look carefully what is a result of such a program.
3. Write functions responsible for actions done by a simple calculator.
4. Connect all functionalities together and prepare a separate function which will be responsible for a calculator's menu.
5. Prepare a new program and use there 5 different functions. In each function use a local variable `x`. Try to display from different places these variables and take a look at the result.
6. Create two arrays and fill them using values given by the user. Display these arrays.
7. Merge these arrays together, however, in the first array there will be stored only even values and in the second array there will be stored only odd values.
8. Repeat step no. 6, however, use integer as well as `float`-type values.
9. Merge these arrays together, however, in the first array there will be stored only integer values and in the second array there will be stored other values.