1 student:

Working with files

1. `[st1-task-01.cs] {0.8 points or 1.5}` Download the `in.txt` file and paste it into directory of your project (~/bin/Debug) or some folder you can read from. Create a program to print out **positive numbers** from the file into the console window. You should use the `BinaryReader` class and `PeekChar()` method. Make a program using `try..catch` block. You shouldn't use here *Lists* and lambda expressions.

Expected output:

```
Positives:
7 4 0 2 9 8 4 9 9 1 3
```

The similar tasks you can find here: https://labs-org.ru/c-sharp14-eng/

Classes

2. `[st1-task-02.cs] {1.6 points or 2.5}`

1). Create a *class* with a name `PhysicalCulture` to manage the information about the pupils' achievements in Physical Culture lessons. The fields of the class are:

- pupil's **surname** (of *string* type)
- **sex** (of *char* type: letters `f` or `m`)
- length of the long **jump** (in centimeters, of *double* type)
- number of **push-ups** (of *integer* type)
- 100 meters **running** (in seconds, of *double* type)

2). Create auto read- and write- properties for all the fields (setters and getters).

3). Define a *constructor* with the initializations of the fields and add the `Print` method for the class to output all the information about pupil.

4). Within the `Main` function initialize two instances of the class. Print out the information about those two pupils using `Print` method.

5). Create a `RunningPlanIsDone` method (returns `true` if the plan of 100 meters **running** is done: it must be less than 13.8 seconds – for boys, and for girls is less than 16.3 seconds; returns `false` otherwise). Call the method for the pupils within the `Main` function.

Note: Specify the meaningful names for the class, its fields, and methods.

Expected output:

```
Pupil's info:
surname: Ivanov,  sex: male, jump length: 175 cm, push-ups: 35, running: 12.5
Pupil's info:
surname: Kruglova,  sex: female, jump length: 145 cm, push-ups: 6, running: 18.5
Ivanov: plan of running is done – true
Kruglova: plan of running is done - false
```

The similar tasks you can find here: https://labs-org.ru/c-sharp17-eng/

Lists

3. `[st1-task-03.cs] {1.6 points or 2}` Download `in.txt` file and place it into the folder of your project executable files (~/bin/Debug) or some folder you can read from. Create the program to read numbers from the file and to add them to a variable of *List* type. To add the numbers into the list variable (`L1` is a variable of List type):

```
while (br.PeekChar() != -1)
            {
                L1.Add(br.ReadInt32());
            }
```

1. Create a function to output all the odd numbers from that file. You should use LINQ `Where` method.

2. Create a function to return the **quantity** of all the elements from the file which are **greater than 5**. You should use LINQ `Where` and `Count` methods (`…Where(…).Count();`).

Note: You should use `FileMode.Open` mode and `BinaryReader` class to read from a file.

Expected output:

```
// Odd elements:
-1 7 -5 9 9 -7 9 1 3
// The quantity of all the elements > 5:
5
```