

Assignment 1 – Deadline: Tuesday, June 01, 2021.

You have to do this work individually. You must submit **a single** compressed folder **A1_123456.zip**. Replace 123456 with your student number. The folder must contain 5 files, one for each question.

Each question i must be submitted in the file A1Qi.java ($1 \leq i \leq 5$)

All your files. java must also have these lines at the beginning of the program:

//Student name: your first and last name

//Student number: your student number

The submission must be done on Brightspace, **before the deadline**.

No submission after the deadline will be accepted. Late submissions receive the mark of zero.

- Only submit your A1Qi.**java** files, do **not** submit any .class files.
- You are **NOT** allowed to use any library other than java.util.Scanner in your programs.
- You must test your programs to make sure they give the correct results.
- A program that gives a **compilation error or an execution error will receive a mark of zero**.
- When the question says write a method, you **must respect the name of the method exactly and the names of the parameters** when given in the assignment questions.
- You have to put **comments** in the programs. The comments are **not** needed for each line of code. However, they have to explain in **simple English** what the program does.
- For each question, there are examples given. Make sure your **program output conforms** exactly to those examples.
- Some java files A1Qi.java are provided, **they should be used** and the content already there **cannot** be modified. Which means you need to **keep** the existing code and **add** your code to it.

Marking scheme (total of 100 points)

- Question 1 : 20 points
- Question 2 : 20 points
- Question 3 : 20 points
- Question 4 : 20 points
- Question 5 : 20 points

Academic Integrity

<https://www.uottawa.ca/administration-and-governance/academic-regulation-14-other-important-information>
<https://www.uottawa.ca/vice-president-academic/academic-integrity>

By submitting this assignment, you acknowledge the following:

1. I have read the academic regulations of the University of Ottawa regarding academic fraud.
2. I understand the consequences of plagiarism.
3. With the exception of the source code provided by the professor for this course, all the source code is mine.

Question 1

Implement a Java method called `talk`.

This method takes as first parameter an array of sentences of type `String`, and as second parameter the number `n` of sentences in the array.

The method returns an array of two elements of type `int`.

- The first element is the length of the longest sentence
- The second element is the number of times sentences of this length are repeated in the array.

In the main program, first prompt the user to enter the number `n` of sentences, then prompt the user to enter `n` sentences separated by commas. Call the method `talk` and display the result returned by the method in the way shown in the examples below.

Make sure your program's output conforms to the examples below.

Examples (several executions):

```
Please enter the number of sentences : 3
Please enter an array of 3 sentences separated by commas :
Hello,How are you,I am good
The length of the longest sentence is : 11
Sentences of this length are repeated : 1 time(s)
```

```
Please enter the number of sentences : 1
Please enter an array of 1 sentences separated by commas :
excellent
The length of the longest sentence is : 9
Sentences of this length are repeated : 1 time(s)
```

```
Please enter the number of sentences : 7
Please enter an array of 7 sentences separated by commas :
cat a, rat b, flat, ah, key one, down up, cat b
The length of the longest sentence is : 7
Sentences of this length are repeated : 2 time(s)
```

Question 2

Implement a Java method called `myCalc`, which takes an array of `n` elements of type `int` as first **parameter**, and the number `n` of type `int` as second parameter. This method returns a value `R` of type `int`. The returned value `R` is equal to the number of positive elements (> 0) minus the number of negative elements (< 0) found in the array of `n` elements. Elements of value zero ($= 0$) do not affect the value of `R`.

In the **main program**, ask the user to enter the array of elements separated by **commas**.

For this question **and all** the following questions, **you cannot ask the user the number of elements that are in the array**. You have to find that number `n` within the program you write.

Then, call the method above and pass the array of `int`, and the number `n`, as parameters. Finally, display the result which is the return value `R` of the method.

You can assume that the user will respect the input that you asked for. Which means the values entered will be numbers (`int`) separated by commas. Assume that **no** other character will be entered, and **no** commas at the beginning or at the end, only between each two numbers.

Your program output has to respect the output shown in these examples exactly.

Examples (several executions):

```
Please enter an array of values separated by commas :
```

```
1,-3,-9,0,1,-4
```

```
The number of positive elements minus the number of negative  
elements is: -1
```

```
Please enter an array of values separated by commas :
```

```
5,3,9,0,3,0,2
```

```
The number of positive elements minus the number of negative  
elements is: 5
```

```
Please enter an array of values separated by commas :
```

```
-2,-8,5,0,-1,0,-2
```

```
The number of positive elements minus the number of negative  
elements is: -3
```

Question 3

You need to manage a store. You initially have two arrays:

- The first is array1: contains the names of items in the store, it has elements of type string.
- The second is array2: contains the quantities of these items, it has elements of type int.

You want to build an array of elements of type **Item**, where Item is a class containing both the names of items and quantities of the items in this store. The instance variables of this class Item are:

- 1- name: which are the names of items from array1.
- 2- quantity: which are the respective quantities from array2.

The arrays array1 and array2 are not necessarily of the same length.

If array1 is longer, the item quantity will be zero (= 0) for the rest of item names of array1.

If array2 is longer, the item will contain articles of name “unknown”, and quantities from array2.

In the main program, ask the user to enter the elements of array1 and array2, separated by commas. Construct an array of type **Item**, and display the result as in the examples.

- The class Item is provided, do **not** modify it. Use it as it is to construct your array of Items.
- The method **display** provided in class Item, should be used and is the only one allowed to use to print the array of Items.

Make sure the output of your program is as shown in the examples below.

```
Please enter the items names separated by commas :
shirt, t- shirt, pants, dress, socks, shoes
Please enter the items quantities separated by commas:
20,15,25,30,10,27
the store items are:
shirt : 20 - t- shirt : 15 - pants : 25 - dress : 30 - socks :
10 - shoes : 27 -
```

```
Please enter the items names separated by commas :
shirt, t- shirt, pants, dress, socks, shoes
Please enter the items quantities separated by commas:
20,15,25,30
the store items are:
shirt : 20 - t- shirt : 15 - pants : 25 - dress : 30 - socks :
0 - shoes : 0 -
```

```
Please enter the items names separated by commas :
shirt, t-shirt
Please enter the items quantities separated by commas:
20,15,25,30,10
the store items are:
shirt : 20 - t-shirt : 15 - unknown : 25 - unknown : 30 - unknown
: 10 -
```

Question 4

A player qualifies for a bonus program if he has more than 20 goals or more than 25 assists, or gets less than 25 penalties during the season. Players who qualify for the program according to these rules are separated into three groups:

- Group 1. Players who have been a member of the team for 5 years or more and who have participated in more than 55 matches during the regular season.
- Group 2. Players who have been a member of the team for 5 years or more and who have participated in 55 matches or less during the regular season.
- Group3. Players who have been a member of the team for less than 5 years, regardless of the number of matches played during the regular season.

According to these groups, players get a bonus in this way:

- Group 1 players: get a full bonus.
- Group 2 players: get a partial bonus.
- Group 3 players: get a conditional bonus.
- Players not belonging to any of the 3 groups: do not get any bonus.

In the financial database, the bonuses have the following codes:

Bonus	Bonus Code
Full bonus	3
Partial bonus	2
Conditional bonus	1
No bonus	0

a) Write a Java method that takes a player's data as parameters. So it takes 5 parameters of type int, which are the numbers entered by the user. This method returns the bonus code the player gets. Example: if the player obtains a conditional bonus, the result is 1. This result is of type int.

b) Write the main Java program that asks the user for a player's statistics and reads them at the keyboard. The numbers entered must be positive or zero (≥ 0) so this should be checked, and the user should be re-prompted to enter the number until this condition is verified. If one number is negative (< 0) then the user should be prompted to re-enter **only** this number (not all).

The main program uses the method of question (a) and passes the 5 player's statistics to it.

Then the main program displays the player's bonus code returned by the method.

Make sure you respect the output as shown in the examples below in your program.

These examples do not include all the scenarios, make sure you test the other bonus codes scenarios.

Examples

Enter the number of the player goals during the season: 10

Enter the number of the player assists during the season: 20

Enter the number of penalties during the season: 5

Enter the number of matches played: 20
Enter the number of years of service: 5
The bonus code of this player is: 2

Enter the number of the player goals during the season: 22
Enter the number of the player assists during the season: 30
Enter the number of penalties during the season: -1
Enter the number of penalties during the season: -5
Enter the number of penalties during the season: 0
Enter the number of matches played: 30
Enter the number of years of service: 4
The bonus code of this player is: 1

Question 5

Implement a Java main program that asks the user to enter a few lines (rows) of numbers separated by commas, and which constructs an array of arrays called `arrayOfArrays`, out of these entered numbers. All the elements of `arrayOfArrays` are of type `int`. You **cannot** ask the user how many rows, or how many elements in each row the user will enter.

- The number of elements in a given row of `arrayOfArrays` is initially unknown, and can be different for each row. Elements are separated by commas. Only **one row** per line.
- The number of rows of `arrayOfArrays` is initially unknown. The program knows the user has entered the last row when a line is left empty and “enter” is pressed (see examples below). The user can enter a maximum of 5 rows (ie: `arrayOfArrays` can have between 0 and 5 rows).

Implement a **method display** that takes an array of arrays as parameter and displays all its elements. The main program calls the method `display(arrayOfArrays)` to print the elements as shown below.

You can assume that the user will respect the input that you asked for. Which means the values entered will be numbers (`int`) separated by commas. Assume that **no** other character will be entered, and **no** commas at the beginning or at the end, only between each two numbers.

Make sure the output of your program follows the examples below.

Examples :

Enter the array of arrays one row per line. Enter an empty line when you are done.

Elements of each row should be separated by commas:

1,2,3,4,5

6,7,8,9

0,1,10,14,22,63,70

The array of arrays is:

Row 0 : 1 2 3 4 5

Row 1 : 6 7 8 9

Row 2 : 0 1 10 14 22 63 70

Enter the array of arrays one row per line. Enter an empty line when you are done.

Elements of each row should be separated by commas:

10

33,72,90,2,4,1,8

4,3,56,7

2,5,9

The array of arrays is:

Row 0 : 10

Row 1 : 33 72 90 2 4 1 8

Row 2 : 4 3 56 7

Row 3 : 2 5 9