

A- APROG Ratings

Write a program to read the APROG exam scores for a class with N students (for an array). The number of students (N) is entered by the user. Then calculate and show the class average. Finally, display how many students failed the course (grade <10). The program should display the class average (formatted to 1 decimal place) and how many students failed the course on separate lines with the following format:

average=<average>
failures=<number of failures>

Example:

Input	Output
5 12 8 17 19 5	average=12.2 failures=2

B- Minimum

Write a program to read a set of positive integers (for an array). Number reading ends when a negative number is entered. Then, determine which is the element with minimum value in the set and how many times it has been inserted. The last number (negative) should be avoided. The program should display the number with minimum value and how many times it was inserted, on separate lines and with the following format:

min=<lowest number>

occurrences=<number of occurrences>

Example1:

Input	Output
5 12 2 17 19 -5	min=2 occurrences=1

C- Ascending Sequence

Write a program to verify if a sequence of numbers entered by the user is always ascending. The program should prompt the user for the amount of numbers to enter and then read them.

At the end, the program should display one of the following messages: "always ascending = true" or "always ascending = false" if the sequence of numbers entered is always ascending or not, respectively.

For this purpose, implement:

- a) A method to read and store in an array the numbers entered by the user.
- b) A method to check if the array sequence is ascending. The method must return true or false if the sequence is always ascending or not, respectively.

Example1:

Input	Output
5 11 12 13 14 15	always ascending = true

Example2:

Input	Output
6 11 12 1 10 2 15	always ascending = false

D- Grades frequency

Write a program to read the grades that N students obtained on a test, and calculate the absolute frequencies for the grades obtained. It is intended to know for each grade, from zero (0), one (1), ..., up to twenty (20), how many students obtained this grade. Grades are integer values and can range from zero (0) to twenty (20) inclusive. The program should start by reading the value of N, followed by reading the grade of each student.

The program must display the scale from zero (0) to twenty (20) and, for each value of the scale, the number of students who obtained this grade, separated by a space. Each scale grade (from zero to twenty) and its frequency should appear on separate lines in the following format:

0 <n. students with grade 0>

1 <n. students with grade 1>

...

20 <n. students with grade 20>

Implement the following functionality using modularization:

- Read and store student grades.
- Determine the frequencies of grades from zero to twenty.
- Display frequency values

Example1:

Input	Output
7	0 0
17	1 0
9	2 0
13	3 0
17	4 0
9	5 0
15	6 0
17	7 0
	8 0
	9 2
	10 0
	11 0
	12 0
	13 1
	14 0
	15 1
	16 0
	17 3
	18 0
	19 0
	20 0

Example2:

Input	Output
8	0 0
12	1 0
13	2 0
10	3 0
5	4 0
12	5 1
10	6 0
12	7 0
10	8 0
	9 0
	10 3
	11 0
	12 3
	13 1
	14 0
	15 0
	16 0
	17 0
	18 0
	19 0
	20 0

E- Salary Statistics

Write a program to determine some statistics on employee salaries of a company. The number of employees may vary over time, will be at least one, but will never exceed 20. The program must read to each employee their name and salary. Reading ends when the name "end" is entered. The program must display in separate lines the average salary (with 1 decimal place) as well as the name of each employee earning less than the average salary.

Example1:

Input	Output
ana 1000 berta 500 carla 1200 daniela 5000 end	1925.0 ana berta carla

F- Financial investment

Write a program to simulate a financial investment after 6 months. The investment consists of an initial bank deposit with capitalizable bank rate for each month, ie, at the end of each month the deposit amount will be increased by the respective bank rate of that month. First, the program must read the bank rates for each of the six months. Second, it must read the amount of the initial deposit.

Finally, it must display the amount of the deposit after 6 months, using two decimal places in the following format: *"final value=<value>"*

Example1:

month		1	2	3	4	5	6
rate		0.15	0.20	0.11	0.01	0.02	0.05
deposit	10000.00	11500.00	13800.00	15318.00	15471.18	15780.60	16569.63

Input	Output
0.15 0.20 0.11 0.01 0.02 0.05 10000	final value=16569.63

G- Invert sequence

Write a modular program to read a sequence of N integers and display them in the reverse order of insertion. The value of N is initially entered by the user.

Each number must be displayed in a separate lines.

Example1:

Input	Output
5	50
10	40
20	30
30	20
40	10
50	

H- Rotate sequence

Write a modular program to read a sequence of N integers. The value of N is initially entered by the user. Then the program should read the N integers. Finally, the program must accept 3 possible commands: "right", "left" and "exit" whose meaning is:

"right" - rotate the sequence 1 position to the right (eg 1 2 3 4 → 4 1 2 3)

"left" - rotate the sequence 1 position to the left (eg 1 2 3 4 → 2 3 4 1)

"exit" - end the program

The program must accept "right" or "left" commands until the "exit" command that terminates the program. Each time the "right" or "left" command is executed, the result of the sequence should be displayed.

Each sequence should be displayed on a single line and each element should be displayed in square brackets in the following format: "[a][b]...[n]"

Example1:

Input	Output
5	[50] [10] [20] [30] [40]
10	[40] [50] [10] [20] [30]
20	[50] [10] [20] [30] [40]
30	
40	
50	
right	
right	
left	
exit	

I- Condominium

Consider a completely occupied apartment building with 3 entrances and 4 floors. Write a modular program to:

-**First**, read the name of the owner of each apartment. The reading should be done in order of entrance and floor, ie, for each entrance must be read the owners of each of its floors.

-**Second**, read a name and display which entrance and floor it owns, in the following format:

"name = <name>"

"entrance = <entrance number>"

"floor = <floor number> "

If there are multiple solutions (multiple owners with the same name), all possibilities should be displayed, in ascending order of entrance and floor. If there is no owner with that name in the building, the message "Do not live in the building" should be displayed.

Example:

floor				
3	Luisa Lima	Hercilia Hora	Maria Mota	
2	Carla Costa	Gloria Gomes	Luisa Lima	
1	Berta Barata	Francisca Fe	Joana Jales	
0	Ana Anacleto	Elsa Eira	Ines Iris	
	0	1	2	
		entrance		

Example1: (search for Luisa Lima)

Input	Output
Ana Anacleto	name=Luisa Lima
Berta Barata	entrance=0
Carla Costa	floor=3
Luisa Lima	name=Luisa Lima
Elsa Eira	entrance=2
Francisca Fe	floor=2
Gloria Gomes	
Hercilia Hora	
Ines Iris	
Joana Jales	
Luisa Lima	
Maria Mota	
Luisa Lima	

