

A- Palindrome

Make a program to read a sequence of words until it finds a palindrome (word whose reading from left to right is the same as from right to left). Implement a module to check if a word is a palindrome. The module should receives a word and returns the indication whether or not that word is a palindrome. The output should be the number of words read before the palindrome.

Example:

Entrada	Saída
joao nada banana ana	3

B- Ranking chart

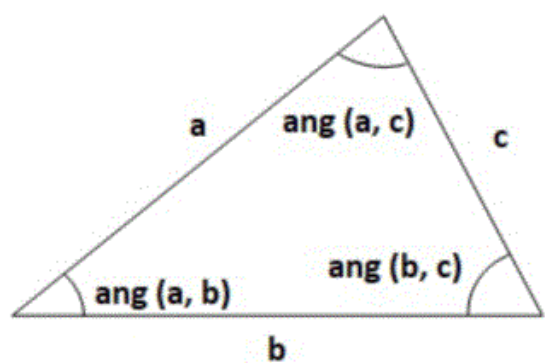
Write a program to display, in the form of bar graph, the number of students in a class that scored "Approved" and "Failed" on a set of subjects. The program should start by asking the number of students in the class and the number of subjects and, for each subject will ask the name of the subject and the number of students with approval. Implement a module to print information about a subject.

Example:

Input	Output
10 2 Portuguese 6 Math 7	Subject: Portuguese - Approved: ***** - Failed: **** Subject: Math - Approved: ***** - Failed: ***

C- Calculate the angles of a triangle

Write a program that asks for the three-sided measurements of a triangle. Make sure they are valid and a triangle can be formed. If so, calculate all 3 internal angles of this triangle. Implement a method that calculates an inner angle of a triangle. The method must take as parameter the measurements on the three sides of the triangle and returns the calculated angle in degrees.



Ângulo	Fórmula
$ang(a,b)$	$arc\cos\left(\frac{a^2+b^2-c^2}{2ab}\right)$
$ang(a,c)$	$arc\cos\left(\frac{a^2+c^2-b^2}{2ac}\right)$
$ang(b,c)$	$arc\cos\left(\frac{b^2+c^2-a^2}{2bc}\right)$

Results should be displayed in separate lines and angle values in degrees (rounded to units). If it is not possible to form a triangle, the message "impossible" should be displayed.

Example:

Input	Output
2 3 4	a=2 b=3 c=4 ang(a,b)=104 ang(a,c)=46 ang(b,c)=28

D- Combinations and permutations

Write a program that calculates the number of possible combinations and permutations of a set of elements. You must enter the total quantity of elements (m) and the subset size to group (n). The formulas are as follows:

- Combinations of m elements, in subsets of size n ($m \geq n$).

$$C(m, n) = \frac{m!}{n! (m-n)!}$$

Permutations of m elements, in subsets of size n ($m \geq n$).

$$P(m, n) = \frac{m!}{(m-n)!}$$

Each result should appear on separate line and in the following format:

C(5.2)=10

P(5.2)=20

NOTE: Implement the following methods: factorial(), permutations(), combinations().

Example:

Input	Output
5 2	C (5.2) =10 P (5.2) =20

E- Digits at common positions

Write a program to read N pairs of positive integers. The value of N is entered by the user. After reading the N value pairs, the program must display the pair that has the most common digits at the same positions. If there is more than one pair with the same number of digits in common, the last pair found must be displayed. Implement a module that receives two positive integers and returns the number of common digits in the same positions.

The result should contain only the pair in question in the following format: "number1/number2". If there are no pair of numbers that have digits in common, the message "no results" should be displayed.

Example:

Input	Output
3 12345 345 13579 12529 123 4567895	13579/12529

F- Volume of solids of revolution

Write a program to determine volumes of solids of revolution. Consider only these types of solids: cylinder, cone and sphere. For each solid, the type of solid and its dimensions (radius and height if required) will be entered. The program must be able to function repeatedly until the word "end" is entered as a solid type. Each result should be presented on separate lines and with two decimal places. Implement the program in a modular way.

NOTE:

Volume sphere = $\frac{4}{3} \pi R^3$

Volume cylinder = $\pi R^2 \text{ Height}$

Volume cone = $\frac{1}{3} \pi R^2 \text{ Height}$

Example:

Input	Output
cone 2.5 3 sphere 3.1 end	19.63 124.79

G- Armstrong Numbers

Write a program to display all Armstrong numbers up to a user-entered value N. An Armstrong number has the following characteristic: the sum of the digits raised to the number of digits equals the number itself.

For example:

$$2 = 2^1 \quad (1 \text{ digit} \Rightarrow \text{sum all digits raised to } 1)$$

$$407 = 4^3 + 0^3 + 7^3 \quad (3 \text{ digits} \Rightarrow \text{add all digits raised to } 3)$$

$$1634 = 1^4 + 6^4 + 3^4 + 4^4 \quad (4 \text{ digits} \Rightarrow \text{add up to } 4 \text{ digits})$$

Implement a module to verify that an integer is an Armstrong number.

Example:

Input	Output
200	0 1 2 3 4 5 6 7 8 9 153

H- Palindrome

Write a program to find a palindrome in a sequence of user-entered integers. A palindrome is a sequence that reads the same backward as forward. The program should read a number, verify if it is a palindrome and, if so, the message "palindrome" should be displayed, otherwise another number should be read. The program ends when it finds a palindrome or the size of the sequence exceeds a threshold value (5). The threshold value is the maximum number of attempts (5) without finding a palindrome. The message "attempts exceeded" should be displayed if the threshold value is exceeded. Implement a method that checks whether or not an integer is a palindrome.

Example:

Input	Output
213 33331 42124	palindrome

I- Fibonacci Number

Write a program to determine if a user-entered integer is a number of the Fibonacci sequence.

In Fibonacci's succession, the first term is zero (0), the second is one (1), and any of the other terms is the sum of the previous two (0,1,1,2,3,5,8,13, 21,34,55,89,144, ...).

The message "is a Fibonacci number" or "is not a Fibonacci number" should be displayed if the number entered belongs to the sequence of Fibonacci or not, respectively.

Example1:

Input	Output
13	is a Fibonacci number

Example2:

Input	Output
100	is not a Fibonacci number

J- Guess the number

Write a program to implement a game so you can have fun with your tablemate. The game is intended for two players.

Player1 writes a secret number on keyboard (integer in range [0.100]), without Player2 seeing. Then Player2 tries to guess the secret number. The program should read the secret number (Player1) and print 30 blank lines so that the entered number is no longer visible to Player2.

Next, Player2 must enter numbers until he guesses the secret number entered by Player1.

For each attempt by Player2, the program should display (per line) one of the following messages:

- "Try smaller" if Player2's guess is greater than the secret number;
- "Try bigger" if Player2's guess is less than the secret number;
- "Guessed" if Player2 guessed the secret number.

At the end, the number of attempts used should also be displayed. Write a method that gets a secret integer per parameter and ask player2 to guess what that number is. The method should indicate whether the user's guess was bigger, smaller or guessed the secret number.

The method should return the number of attempts made to guess the secret number. In the next examples, the phrase "... 30 blank lines ..." should not be displayed. It is merely illustrative. Represents 30 blank lines.

Example1:

Input	Output
10	... 30 blank lines ...
5	Try bigger
7	Try bigger
20	Try smaller
10	Guessed
	4

Example2:

Input	Output
10	... 30 blank lines ...
10	Guessed
	1

