modularização

A - Média de números com menos de X algarismos

a) Construa um módulo para determinar a quantidade de dígitos de um número dado como parâmetro.

b) Utilizando o módulo anterior faça um programa que leia uma sequência de números inteiros com menos de N algarismos e determine a sua média. O valor de N deve ser o primeiro número a ser lido. A sequência termina quando for introduzido um número com N ou mais algarismos.
A média deve ser visualizada com 2 casas decimais.

c) Adapte o programa para terminar no máximo ao fim de K números introduzidos.

OBS: Utilize uma constante para definir o número máximo de elementos da sequência (K=5).

Exemplo1:

Entrada	Saída
4 123 5 41 1000	56.33

Exemplo2:

Entrada	Saída
5 1234 23 12345678	628.50

modularização

B - Gráfico de classificações

Faça um programa que represente sob a forma de gráficos de barras, o número de positivas e negativas dos alunos de uma turma a um conjunto de disciplinas.
O programa deverá começar por pedir o nº de alunos da turma e o nº de disciplinas e para cada disciplina pedirá o nome da disciplina e o nº de alunos aprovados.
Deve implementar um módulo para imprimir a informação de uma disciplina.
O output produzido deverá ter o seguinte aspeto:

Disciplina: Portugues
- Positivas: **********
- Negativas: ****
Disciplina: Matematica
- Positivas: **********
- Negativas: *********

Exemplo:

Entrada	Saída
10 2 Portugues 6 Matematica 7	Disciplina: Portugues - Positivas: ****** - Negativas: **** Disciplina: Matematica - Positivas: ******* - Negativas: ***

modularização

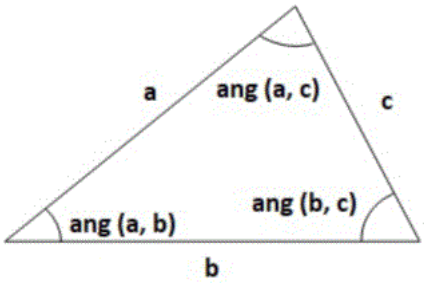
C - Calcular ângulos de um triângulo

a) Implemente um método (calcAng()) que calcule um ângulo interno de um triângulo, sendo dadas as medidas dos três lados desse triângulo. O método deve receber por parâmetro as medidas dos três lados do triângulo e retorna r o ângulo calculado em graus.

b) Faça um programa que peça as medidas de três lados, verifique se elas são válidas e se é possível formar um triângulo. Em caso afirmativo calcule todos os ângulos internos desse triângulo. Para isso chame três vezes o método desenvolvido na alínea anterior.

OOs resultados deverão ser apresentados em linhas separadas e os valores dos ângulos em graus, com duas casas decimais. No caso de o triângulo não ser possível, a mensagem deverá ser: "impossivel". O resultado deverá apresentar o seguinte formato:
a=2.00
b=3.00
c=4.00

ang(a,b)=104.48
ang(a,c)=46.57
ang(b,c)=28.96



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)	$arc\cos\left(\frac{b^2+c^2-a^2}{2bc}\right)$

Exemplo:

Entrada	Saída
2 3 4	a=2.00 b=3.00 c=4.00 ang(a,b)=104.48 ang(a,c)=46.57 ang(b,c)=28.96

modularização

D - Combinações e permutações

Faça um programa que calcule a quantidade de combinações e permutações possíveis de um conjunto de elementos. Deve introduzir a quantidade total de elementos (m) e a quantidade a agrupar (n).
As fórmulas são as seguintes:

- Combinações de m elementos, n a n (m>=n).

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

-Permutações de m elementos, n a n (m>=n).

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

Cada resultado deverá aparecer em linhas separadas e no seguinte formato:
C(5,2)=10
P(5,2)=20

OBS: Implemente os seguintes métodos:
fatorial()
combinacoes()
permutacoes()

Exemplo:

Entrada	Saída
5 2	C(5,2)=10 P(5,2)=20

modularização

E - Algarismos em posições comuns

a) Implemente um módulo que dados dois números inteiros positivos retorne a quantidade de dígitos comuns nas mesmas posições.

b) Elabore um programa que leia N pares de valores inteiros positivos, sendo N introduzido pelo utilizador e validado. Após a leitura dos N pares de valores o programa deve apresentar o par que tiver mais dígitos comuns nas mesmas posições. No caso de haver mais do que um par com a mesma quantidade de dígitos em comum, deve ser apresentado o último par encontrado.
O resultado deve conter apenas o par em causa no formato "numero1/numero2".
No caso de não haver nenhum par de números que tenha algarismos em comum deve ser apresentada a mensagem: "sem resultados".

Exemplo:

Entrada	Saída
3 12345 345	13579/12529

13579 12529 123 4567895	

modularização

F - Volume de sólidos de revolução

Faça um programa que permita determinar volumes de sólidos de revolução (cilindro, cone e esfera).
Para cada sólido será introduzido o tipo de sólido e as respetivas dimensões (raio e altura, se necessário).
Cada resultado deverá ser apresentado em linhas separadas e com duas casas decimais. O programa termina quando o tipo de sólido for a palavra “fim”.
Implemente o programa de forma modular.

OBS:
V esfera = 4/3 π R3
V cilindro = π R2 Altura
V cone = 1/3 π R2 Altura

Exemplo:

Entrada	Saída
cone 2.5 3 esfera 3.1 fim	19.63 124.79

modularização

G - Números de Armstrong

Faça um programa que imprima todos os números de Armstrong até um valor N inserido pelo utilizador.
Construa um módulo para verificar se um número inteiro é um número de Armstrong.
Um número de Armstrong possui a seguinte característica: a soma dos algarismos elevados à quantidade de algarismos é igual ao próprio número.
Por exemplo:
2 = 2^1 (1 algarismo => somar todos os algarismos elevados a 1)
407 =4^3 + 0^3 + 7^3 (3 algarismos => somar todos os algarismos elevados a 3)
1634 = 1^4 + 6^4 + 3^4 + 4^4 (4 algarismos => somar todos os algarismos elevados a 4)

Exemplo:

Entrada	Saída
200	0 1 2 3 4 5 6 7 8 9 153

modularização

H - Capicua

a) Faça um método que verifique se um número inteiro é ou não capicua.

b) Faca um programa que leia uma sequência de números inteiros e termine quando for introduzido um número capicua ou quando tiver analisado um máximo de tentativas (5) sem o encontrar.

Deve ser visualizado a mensagem "capicua" ou "tentativas excedidas"

Exemplo:

Entrada	Saída
213 33331	capicua

42124	

modularização

I - Elemento de Fibonacci

Elabore um programa para determinar se um número inteiro introduzido pelo utilizador é um elemento da sucessão de Fibonacci.
Na sucessão de Fibonacci, o primeiro termo é zero (0), o segundo é um (1) e qualquer um dos outros termos é a soma dos dois anteriores (0,1,1,2,3,5,8,13,21,34,55,89,144,...).

Deve ser visualizada a mensagem "e de Fibonacci" ou "nao e de Fibonacci" caso o número inserido pertença à sucessão de Fibonacci ou não, respetivamente.

Exemplo1:

Entrada	Saída
13	e de Fibonacci

Exemplo2:

Entrada	Saída
100	nao e de Fibonacci

modularização

J - Adivinha o número

Escreva um programa para implementar um jogo de forma a que possa divertir-se com o seu colega.

O jogo destina-se a dois jogadores. O jogador1 escreve um número secreto (inteiro no intervalo [0,100]), sem o jogador2 ver qual é. De seguida, o jogador2 tenta acertar no número secreto.

O programa deve ler o número secreto (jogador1) e imprimir 30 linhas em branco para que o número inserido deixe de ser visível para o jogador2.
De seguida, o jogador2 deve inserir números inteiros até adivinhar o número secreto inserido pelo utilizador1. Para cada tentativa do jogador2, o programa deve indicar uma das seguintes mensagens:
• “Tente menor”, se o palpite do jogador2 é maior que o número secreto;
• “Tente maior”, se o palpite do jogador2 é menor que o número secreto;
• “Acertou”, se o jogador2 acertou no número secreto.
No final, deve também ser visualizado o número de tentativas utilizadas.

Escreva um método que recebe um número inteiro secreto por parâmetro e solicite ao jogador2 que tente adivinhar qual é esse número. O método deve indicar se o palpite do utilizador foi maior, menor ou se acertou no número secreto. O método deve retornar a quantidade de tentativas realizadas para acertar no número secreto.

Exemplo1:

Entrada	Saída
10 5 7 20 10	...30 linhas em branco... Tente maior Tente maior Tente menor Acertou 4

Exemplo2:

Entrada	Saída
10 10	...30 linhas em branco... Acertou 1

modularização

K - Substituir caracter numa frase

Elabore um programa que substitua um caracter por outro numa frase. O programa deve ler uma frase e dois caracteres e substitui todas as ocorrências do primeiro carácter pelo segundo carácter.
O programa deve visualizar a frase resultante.

Implemente um módulo que recebe por parâmetro uma frase e dois caracteres, procede à substituição dos caracteres e retorna a frase resultante.

Exemplo1:

Entrada	Saída
Hoje vou correr e beber agua. e *	Hoj* vou corr*r * b*b*r agua.

Exemplo2:

Entrada	Saída
2 + 2 sao 4 e *	2 + 2 sao 4

modularização

L - Quantidade de palavras de uma frase

Elabore um programa que leia uma frase e escreva quantas palavras possui essa frase.
Considera-se uma palavra a um qualquer conjunto de símbolos que se encontram entre espaços.

Implemente um módulo que recebe a frase e retorna a quantidade de palavras existentes nessa frase.

Exemplo1:

Entrada	Saída
Olá bom dia.	3

Exemplo2:

Entrada	Saída
quem sou eu ?	4

modularização

M - Converter uma frase para maiúsculas e minúsculas

Elabore um programa que leia uma frase e a converta para maiúsculas e minúsculas.
Cada frase deve ser visualizada em linhas separadas.

Implemente um módulo que recebe por parâmetro uma frase e um valor inteiro indicando o tipo de conversão a realizar (>0 converte para maiúsculas, <0 converte para minúsculas) e retorna uma nova frase convertida.

Exemplo1:

Entrada	Saída
Hoje Esta 1 SOL maRAvilha.	HOJE ESTA 1 SOL MARAVILHA. hoje esta 1 sol maravilha.

Exemplo2:

Entrada	Saída
2 + 2 sao 4	2 + 2 SAO 4 2 + 2 sao 4

modularização

X - Palíndromo

Faça um programa que leia uma sequência de palavras até encontrar um palíndromo (palavra cuja leitura da esquerda para a direita é igual à da direita para a esquerda).
Deve implementar um módulo para verificar se uma palavra é um palíndromo, isto é, o módulo recebe uma palavra e retorna a indicação se essa palavra é ou não um palíndromo.
O programa deve mostrar o número de palavras lidas que antecedem o palíndromo.

Exemplo:

Entrada	Saída
joao nada	3

banana ana	

modularização

Y - Tabuadas de um intervalo

Faça um programa que permita escrever tabuadas de multiplicação dos números inteiros positivos pertencentes a um intervalo fechado definido pelo utilizador.

Implemente o programa usando:
a) Um módulo (lerValorInteiroPositivo) que leia e retorne um número inteiro positivo maior que zero. Devem ser lidos números continuamente até ser introduzido um número válido. O módulo retorna o número validado.

b) Um módulo (mostrarTabuadasDoIntervalo) que recebe por parâmetro os limites do intervalo e tem a responsabilidade de processar a tabuada de todos os números desse intervalo. As tabuadas devem ser processadas por ordem crescente.

b) Um módulo (mostrarTabuadaDoNumero) que escreve a tabuada de um número específico recebido como parâmetro. O resultado deve seguir o seguinte formato (exemplo para a tabuada de 7):

Tabuada de 7
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70

Exemplo1:

Entrada	Saída
8 7	Tabuada de 7 7 x 1 = 7 7 x 2 = 14 7 x 3 = 21 7 x 4 = 28 7 x 5 = 35 7 x 6 = 42 7 x 7 = 49 7 x 8 = 56 7 x 9 = 63 7 x 10 = 70 Tabuada de 8 8 x 1 = 8 8 x 2 = 16 8 x 3 = 24 8 x 4 = 32 8 x 5 = 40 8 x 6 = 48 8 x 7 = 56 8 x 8 = 64 8 x 9 = 72 8 x 10 = 80

Exemplo2:

Entrada	Saída
-2 3 -5 -6 3	Tabuada de 3 3 x 1 = 3 3 x 2 = 6 3 x 3 = 9 3 x 4 = 12 3 x 5 = 15 3 x 6 = 18 3 x 7 = 21 3 x 8 = 24 3 x 9 = 27 3 x 10 = 30