

arrays

## A - Classificações de APROG

Faça um programa que leia as classificações de exame de APROG de uma turma com n alunos (para um array). O número de alunos (n) é inserido pelo utilizador.

De seguida, calcule e mostre a média da turma. Finalmente, mostre quantos alunos reprovaram à disciplina (nota <10).

O programa deve mostrar a média da turma (formatada com 1 casa decimal) e quantos alunos reprovaram à disciplina, em linhas separadas e com o seguinte formato:

media=<media>

reprovacoes=<nº de reprovações>

Exemplo:

Entrada	Saída
5 12 8 17 19 5	media=12.2 reprovacoes=2

arrays

## B - Menores elementos

Faça um programa que leia um conjunto de números inteiros positivos (para um array). A leitura de números termina quando for inserido um número negativo.

De seguida, determine qual o menor elemento do conjunto e quantas vezes foi inserido.

O programa deve mostrar o menor número inserido e quantas vezes foi inserido, em linhas separadas e com o seguinte formato:

menor=<numero>

ocorrencias=<nº de ocorrencias>

Exemplo1:

Entrada	Saída
5 12 2 17 19 -5	menor=2 ocorrencias=1

Exemplo2:

Entrada	Saída
12 5 12	menor=5 ocorrencias=3

```
8
5
17
5
-51
```

arrays

## C - Sequência crescente

Faça um programa que verifique se a sequência de números inseridos pelo utilizador é sempre crescente. O programa deve pedir ao utilizador a quantidade de números a inserir e, de seguida, proceder à sua leitura.

No final, o programa deve mostrar uma das seguintes mensagens: "sempre crescente = true" ou "sempre crescente = false" caso a sequência dos números inseridos seja sempre crescente ou não, respetivamente.

Para este efeito construa:

- Um método para ler e guardar num array os números inseridos pelo utilizador.
- Um método para verificar se a sequência do array é crescente. O método deve retornar true ou false, caso a sequência seja sempre crescente ou não, respetivamente.

Exemplo1:

Entrada	Saída
5 11 12 13 14 15	sempre crescente = true

Exemplo2:

Entrada	Saída
6 11 12 1 10 2 15	sempre crescente = false

arrays

## D - Frequências das notas

Faça um programa que leia as classificações de um teste de avaliação resolvido por n alunos e calcule as frequências absolutas para as notas obtidas. Pretende-se saber para cada nota, de zero (0), um (1), ..., até vinte (20), quantos alunos obtiveram essa nota.

As classificações são valores inteiros e podem variar entre zero (0) e vinte (20) inclusive. O programa deve ler o valor de n, seguindo-se a leitura das n notas dos alunos.

O programa deve mostrar a escala zero (0) a vinte (20) e, para cada valor da escala, a quantidade de alunos que obtiveram essa nota, separado por um espaço. Cada nota da escala (de zero a vinte) e respetiva frequência deve aparecer em linhas separadas, no seguinte formato:

```
0 <n. alunos com nota 0>
1 <n. alunos com nota 1>
...
20 <n. alunos com nota 20>
```

Implemente as seguintes funcionalidades usando modularização:

- Ler e armazenar as notas dos alunos.
- Determinar as frequências das notas, de zero a vinte.
- Mostrar os valores das frequências

Exemplo1:

Entrada	Saída
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

Exemplo2:

Entrada	Saída
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

arrays

## E - Estatísticas de vencimentos

Faça um programa para determinar algumas estatísticas sobre vencimentos de funcionários de uma empresa. O número de funcionários pode variar ao longo do tempo, será pelos menos um, mas nunca será superior a 20.

O programa deve mostrar o valor da média dos vencimentos assim como o nome dos funcionários com vencimentos inferior à média.

Usando modularização, implemente as seguintes funcionalidades:

a) Ler os nomes e vencimentos dos funcionários da empresa. Para cada funcionário deve ser lido o seu nome e respetivo vencimento. A leitura termina quando for inserido o nome "fim".

b) Calcular a média dos vencimentos.

c) Mostrar resultados. Mostrar em linhas separadas, o valor da média (1 casa decimal) e os nomes dos funcionários com vencimento inferior a essa média.

Exemplo1:

Entrada	Saída
ana 1000 berta 500 carla 1200 daniela 5000 fim	1925.0 ana berta carla

Exemplo2:

Entrada	Saída
ana 150.5 berta 500.7 fim	325.6 ana

arrays

## F - Aplicação financeira

Faça um programa que realize a simulação de uma aplicação financeira ao fim de 6 meses. A aplicação consiste num depósito bancário inicial com juros capitalizáveis, isto é, no final de cada mês o montante em depósito será acrescido do juro respetivo desse mês.

O funcionamento do programa deve obedecer à seguinte sequência de instruções:

1. Ler as taxas de juro para cada um dos meses (6).
2. Ler o valor do depósito inicial.
3. Mostrar o montante final após os 6 meses de depósito, usando duas casas decimais, no seguinte formato: "valor final=<valor>"

Exemplo1:

Entrada	Saída
0.15 0.20 0.11 0.01 0.02 0.05 10000	valor final=16569.63

arrays

## G - Inverter sequência

Faça um programa modular que leia uma sequência de N números inteiros e os mostre pela ordem inversa da inserção. O valor de N é inserido, inicialmente, pelo utilizador.

Exemplo1:

Entrada	Saída
5 10 20 30 40 50	50 40 30 20 10

arrays

## H - Rodar sequência

Faça um programa modular que leia uma sequência de N números inteiros. O valor de N é inserido, inicialmente, pelo utilizador.

De seguida, o programa deve ler os N números inteiros.

Finalmente, o programa deve aceitar 3 comandos possíveis- "direita", "esquerda" e "sair" cujas funcionalidades são:

"direita" - rodar 1 posição a sequência de números para a direita (ex: 1 2 3 4 → 4 1 2 3)

"esquerda" - rodar 1 posição a sequência de números para a esquerda (ex: 1 2 3 4 → 2 3 4 1)

"sair" - termina o programa

O programa deve aceitar comandos "direita" ou "esquerda" até ser inserido o comando "sair" que termina o programa. Sempre que for executado o comando "direita" ou "esquerda" deve ser mostrado o resultado da sequência.

Cada sequência deve ser visualizada numa linha única e cada um dos seus elementos deve ser mostrado entre parêntesis retos, no seguinte formato:

"[a][b]...[n]"

Exemplo1:

Entrada	Saída
5	[50] [10] [20] [30] [40]
10	[40] [50] [10] [20] [30]
20	[50] [10] [20] [30] [40]
30	
40	
50	
direita	
direita	
esquerda	
sair	

arrays

## I - Condomínio

Considere um prédio de apartamentos com 3 entradas e 4 pisos, completamente ocupados.

Faça um programa modular que leia o nome do proprietário de cada um dos apartamentos.

De seguida, o programa deve ler um dado nome inserido pelo utilizador, procurar e mostrar para esse nome qual é a respetiva entrada e piso, no seguinte formato:

"nome=<nome>"

"entrada=<nº da entrada>"

"piso=<nº do piso>"

Se existirem múltiplas soluções (vários proprietários com o mesmo nome), deve indicar todas as possibilidades.

Caso não exista um proprietário com esse nome no prédio, deve imprimir a mensagem "Nao mora no predio".

Exemplo:

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

## entrada

## Exemplo1: (procurar Luisa Lima)

Entrada	Saída
Ana Anacleto Berta Barata Carla Costa Luisa Lima Elsa Eira Francisca Fé Gloria Gomes Hercilia Hora Ines Iris Joana Jales Luisa Lima Maria Mota <b>Luisa Lima</b>	nome=Luisa Lima entrada=0 pisso=3 nome=Luisa Lima entrada=2 pisso=2

## Exemplo2: (procurar Zita Zulmira)

Entrada	Saída
Ana Anacleto Berta Barata Carla Costa Luisa Lima Elsa Eira Francisca Fé Gloria Gomes Hercilia Hora Ines Iris Joana Jales Luisa Lima Maria Mota <b>Zita Zulmira</b>	Nao mora no predio

arrays

## J - Top 3

Escreva um programa que leia o nome e salário dos funcionários de uma empresa.  
A leitura termina quando for inserido "FIM" como o nome de um funcionário.

De seguida, calcule e mostre o top três (3) dos funcionários mais bem pagos. Caso haja igualdade de vencimentos, a escolha deve seguir a ordenação alfabética crescente do nome.

Mostre em linhas separadas o nome e salário de cada funcionário, ordenados de forma decrescente do salário, no seguinte formato:

"#<ordem>:<nome>:<salario>"

<ordem> indica o lugar no top.

<nome> nome do funcionário

<salario> valor do salário

**Exemplol:**

Entrada	Saída
Ana Anacleto 1000 Berta Barata 1000 Carla Costa 100 Luisa Lima 125 Elsa Eira 5000 Francisca Fé 123 Gloria Gomes 55 Hercilia Hora 250 FIM	#1:Elsa Eira:5000 #2:Ana Anacleto:1000 #3:Berta Barata:1000

arrays

## K - Sem repetições

Escreva um programa que leia um conjunto N de números inteiros e os visualize pela mesma ordem mas sem repetições.

O programa deve ler, primeiro, o valor de N e, de seguida, os N números do conjunto.

Cada um dos elementos do conjunto resultado deve ser visualizado em linhas separadas.

**Exemplol:**

Entrada	Saída
7 10 20 50 10 10 60 20	10 20 50 60

arrays

## L - Vizinhos

Escreva um programa que leia um conjunto números inteiros e visualize todos os números maiores que os seus vizinhos.



Um número é maior que os seus vizinhos quando é, simultaneamente, maior que o número anterior e que o número seguinte.

A leitura termina quando for inserido um número negativo. O tamanho do conjunto de números nunca será superior a 30.

Os elementos do resultado devem ser visualizados em linhas separadas.

8, 2, 4, 1, 6, 12, 5, 9, -1

Exemplo1:

Entrada	Saída
8	4
2	12
4	
1	
6	
12	
5	
9	
-1	

arrays

## M - Algarismos diferentes

Escreva um programa que leia um conjunto números inteiros positivos e visualize, para cada um, quantos algarismos diferentes o constituem.

A leitura termina quando for inserido um número negativo.

Os elementos do resultado devem ser visualizados em linhas separadas, no seguinte formato:

<nºinserido>:<algarismos diferentes>

Exemplo1:

Entrada	Saída
12123518	12123518:5
2222	2222:1
400	400:2
12345	12345:5
-1	

arrays

## N - Diagonais

Escreva um programa que leia uma matriz quadrada de números inteiros e escreva os valores correspondentes às várias diagonais com o mesmo sentido da diagonal principal e da direita para a esquerda. O programa deve ler um valor N ( $1 < N \leq 20$ ) correspondendo à dimensão da matriz. De seguida, deve ler N linhas contendo, cada uma, N números separados por espaços.

Os elementos de cada diagonal devem ser visualizados em linhas separadas e no seguinte formato: [a]...[z].

Exemplo1:

Entrada	Saída
3 10 20 30 50 60 70 90 100 110	[30] [20] [70] [10] [60] [110] [50] [100] [90]

arrays

## O - Quadrantes

Escreva um programa que leia uma matriz quadrada de números inteiros e visualize a soma dos números de cada um dos seus quadrantes.

Se dividir-mos ao meio uma matriz na vertical e na horizontal obtemos 4 quadrantes (q1, q2, q3, q4) na seguinte ordem:

q2	q1
q3	q4

Cada quadrante possui elementos próprios, não compartilhados com outros quadrantes.

O programa deve ler um valor N correspondendo à dimensão da matriz ( $1 < N \leq 20$ ). De seguida, deve ler N linhas contendo, cada uma, N números separados por espaços.

No final deve ser visualizado o valor dos 4 quadrantes da seguinte forma:

[q2][q1]  
[q3][q4]

Exemplo1:

Entrada	Saída
4 10 20 30 40 50 60 70 10 90 100 110 1 1 2 3 4	[140] [150] [193] [118]

arrays

## P - Fatorial dos algarismos

Escreva um programa que leia um número inteiro N ( $0 \leq N \leq 10000000$ ) e um inteiro X ( $0 \leq X \leq 10000000$ ) e determine quantos dos números positivos de zero até N (inclusive) possuem a soma dos fatoriais dos seus algarismos superior a X.

exemplo: para N=5 e X=10

0 : 0! = 1

1 : 1! =1  
2 : 2! =2  
3 : 3! =6  
4 : 4! =24  
5 : 5! =120  
saída: 2

Implemente uma solução eficiente.

Exemplo1:

Entrada	Saída
5 10	2