Aula prática #6 – Vetores e Funções

Problema 1

Escreva um programa que desenhe um histograma de barras horizontais correspondentes ao número de vezes que sai cada uma das faces de um dado. Considere que são feitos 30 lançamentos aleatórios, guardando os resultados num vetor.

Exemplo

```
Histograma de 30 lancamentos:
1 - ****
2 - *******
3 - ****
4 - *****
5 - ******
6 - ****
```

Problema 2

Escreva um programa que crie, preencha e faça o somatório de um vetor de números reais positivo. Neste exercício, assuma que o número de elementos do vetor está limitado a 15. Para tal use as seguintes funções:

```
void preencher_vetor(float v[], int n);
float somatorio_vetor(float v[], int n);
void imprimir_vetor(float v[], int n);
```

Exemplo

```
Quantos elementos pretende armazenar?
18
O numero de elementos esta limitado a 15!
Quantos elementos pretende armazenar?
5
Introduza o elemento 1: 5.2
Introduza o elemento 2: 4.1
Introduza o elemento 3: 3.3
Introduza o elemento 4: 2.7
Introduza o elemento 5: 1.4
O vetor lido foi: {5.2 4.1 3.3 2.7 1.4}
e a soma dos seus elementos e' 16.7.
```

Problema 3

Escreva um programa que preencha um vetor com 10 números introduzidos pelo utilizador. Deverá depois imprimir:
a) A média de todos os seus elementos, calculada por uma função:

```
float avg(float x[]);
```

b) O maior elemento, calculado por uma função:

```
float max(float x[]);
```

c) O menor elemento, calculado por uma função:

```
float min(float x[]);
```

d) O conteúdo do vetor.

Utilize as funções `preencher_vetor` e `imprimir_vetor` que implementou no problema anterior.

Exemplo

```
Introduza o elemento 1: 5.1
Introduza o elemento 2: 3.2
Introduza o elemento 3: 6.3
Introduza o elemento 4: 9.4
Introduza o elemento 5: 10.9
Introduza o elemento 6: 3.8
Introduza o elemento 7: 1.7
Introduza o elemento 8: -1.5
Introduza o elemento 9: 1.0
Introduza o elemento 10: 9.2
Media: 4.91
Maximo: 10.9
Minimo: -1.5
Vetor: {5.1 3.2 6.3 9.4 10.9 3.8 1.7 -1.5 1.0 9.2}
```

Problema 4

Escreva um programa para ler uma sequência de números introduzidos pelo utilizador para um vetor, um de cada vez, terminando com um número negativo. De seguida, deve pedir um número para pesquisar no vetor criado. O programa deve imprimir o tamanho da maior sequência de ocorrências desse número no vetor. Use para esse efeito a seguinte função:

```
1  int contaRepeticao(int v[], int tamanhoVetor, int numero);
```

Exemplo

```
1  Introduza um numero: 1
2  Introduza um numero: 2
3  Introduza um numero: 3
4  Introduza um numero: 4
5  Introduza um numero: 4
6  Introduza um numero: 1
7  Introduza um numero: 3
8  Introduza um numero: 4
9  Introduza um numero: -1
10 Numero a pesquisar: 4
11 Maior sequencia com numeros 4 tem tamanho 2
```

Problema 5

Crie um programa que implemente e teste as seguintes funções:

a) A soma de pares de elementos consecutivos (sem repetição) devolvida no vetor `vRet`. O tamanho do vetor original é indicado no parâmetro `n` e o tamanho do vetor `vRet` é devolvido pela função.

```
1     int sum_v(float x[], int n, float vRet[]);
```

Por exemplo, o vetor {1, 2, 3, 4}, teria como resultado {3, 7}.

b) A diferença entre o maior e o menor elemento.

```
1     float range_v(float x[]);
```

c) Por cada par de elementos, cálculo do maior desvio absoluto em relação à sua média geométrica, devolvido no vetor `vRet`. O tamanho do vetor original é indicado no parâmetro `n` e o tamanho do vetor `vRet` é devolvido pela função.

```
int diff_v(float x[], int n, float vRet[]);
```

A média geométrica de um par de números é dada por: $\sqrt{a_1 a_2}$.
Por exemplo, o vetor {1, 2, 3, 4} resultaria nas médias geométricas {1.414, 3.464} e o resultado da função seria {0.586 0.536}.

Utilize as funções `preencher_vetor` e `imprimir_vetor` que implementou no problema 3 e assuma que o número de elementos dos vetores está limitado a 15.

Exemplo

```
Introduza o elemento 1: 5.1
Introduza o elemento 2: 3.2
Introduza o elemento 3: 6.3
Introduza o elemento 4: 9.4
Introduza o elemento 5: 10.9
Introduza o elemento 6: 3.8
Introduza o elemento 7: 1.7
Introduza o elemento 8: 1.5
Introduza o elemento 9: 1.0
Introduza o elemento 10: 9.2
Vetor Soma: {8.3  15.7  14.7  3.2  10.2}
Gama: 9.9
Vetor Desvio Max Par: {1.1 1.7 4.5 0.1 6.2}
```

Problema 6

Crie um programa que mantenha um vetor ordenado. O utilizador deverá escrever valores, um de cada vez, que serão inseridos na posição respetiva, enquanto os outros elementos serão "arrastados" pelo vetor.
Assuma que o vetor não terá mais que 15 elementos inteiros. Imprima o vetor de cada vez que um valor novo seja inserido.

Exemplo

```
Introduza o elemento 1: 5.1
Vetor: {5.1}
Introduza o elemento 2: 3.2
Vetor: {3.2 5.1}
Introduza o elemento 3: 6.3
Vetor: {3.2 5.1 6.3}
Introduza o elemento 4: 9.4
Vetor: {3.2 5.1 6.3 9.4}
Introduza o elemento 5: 10.9
Vetor: {3.2 5.1 6.3 9.4 10.9}
Introduza o elemento 6: 3.8
Vetor: {3.2 3.8 5.1 6.3 9.4 10.9}
Introduza o elemento 7: 1.7
Vetor: {1.7 3.2 3.8 5.1 6.3 9.4 10.9}
Introduza o elemento  ...
```

Problema 7

Crie uma função `multiEspecial` que receba dois vetores e faça a multiplicação entre os elementos do primeiro vetor e os elementos por ordem invertida do segundo. A função deverá devolver o vetor de resultados através de `vRet`. O tamanho de ambos os vetores é indicado no parâmetro `n`.

```
1     void multiEspecial(int n, int v1[], int v2[], int vRet[]);
```

Crie um pequeno programa de teste de forma a verificar o funcionamento da sua função.

Exemplo

```
1 Quantos elementos? 4
2 Vetor 1: 5 4 1 2
3 Vetor 2: 10 5 1 0
4 Multiplicacao especial: {0 4 5 20}
```