

Exercício 6.1

Escreva um programa que leia uma sequência de N números inteiros, sendo o valor N pedido ao utilizador antes do início da introdução dos números. O programa deve depois imprimir esses números pela ordem inversa com que foram inseridos.

```
import java.util.*;
public class e6_1 {
    static Scanner read = new Scanner(System.in);
    public static void main (String args[])    {
        System.out.print("Quantos elementos ? ");
        int N = read.nextInt();
        int a[] = new int[N];
        for(int i=0; i<a.length;i++)
        {
            System.out.print("elemento "+i+" ");
            a[i] = read.nextInt();    }
        System.out.println("Elementos por ordem inverso:");
        for(int i=a.length-1; i>=0;i--)
            System.out.printf("a[%d] = %d\n",i,a[i]);
    }
}
```

```
Quantos elementos ?    5
elemento 0    334
elemento 1    876
elemento 2    11
elemento 3    43
elemento 4    668
Elementos por ordem inverso:
a[4] = 668
a[3] = 43
a[2] = 11
a[1] = 876
a[0] = 334
Press any key to continue . . .
```

Exercício 6.2

Escreva um programa que leia uma sequência de números inteiros positivos e conte o número de vezes que um determinado número, pedido ao utilizador, aparece na sequência. A leitura deve terminar após a introdução de 100 números ou com o aparecimento de um número negativo.

```
import java.util.*;
public class e6_2 {
    static final int N = 100;
    static final Scanner read = new Scanner(System.in);
    public static void main (String args[]) {
        System.out.print("número ? ");
        int Num = read.nextInt();
        int array[] = new int[N];
        int conta = 0;
        for (int i=0; i<array.length; i++)
        {   array[i] = read.nextInt();
            if (array[i] < 0) break;
            if (array[i] == 0) continue;
            if(Num == array[i]) conta++; }
        System.out.printf("Foi introduzido %d números %d\\n",conta,Num);
    }
}
```

```
número ? 4
23
56
23
4
6
7
2
4
-1
Foi introduzido 2 números 4`
Press any key to continue . . . _
```

Exercício 6.3

Pretende-se escrever um programa que leia do teclado uma sequência de números inteiros positivos e que permita detetar um conjunto de características acerca da sequência. A cada operação deverá corresponder uma função. A leitura da sequência termina quando aparecer o número zero como indicador de paragem ou quando tiverem sido lidos 50 números. A interação com o programa deverá ser feita através de um menu, tal como apresentado de seguida:

Análise de uma sequência de números inteiros

- 1 - Ler a sequência
- 2 - Escrever a sequência
- 3 - Calcular o máximo da sequência
- 4 - Calcular o mínimo da sequência
- 5 - Calcular a média da sequência
- 6 - Detetar se é uma sequência só constituída por números pares
- 10 - Terminar o programa

Opção ->

```
public class e6_3 {  
    static Scanner read = new Scanner(System.in);  
  
    public static void main (String args[]) {  
        int num_array[] = new int[50];  
        int op = 0;  
        do {  
            op = printMenu();  
            switch(op) {  
                case 1:  
                    readSeq(num_array);  
                    break;  
                case 2:  
                    printSeq(num_array);  
                    break;  
                case 3:  
                    printMax(num_array);  
                    break;  
                case 4:  
                    printMin(num_array);  
                    break;  
                case 5:  
                    printMed(num_array);  
                    break;  
                case 6:  
                    printIfEven(num_array);  
                    break;  
            }  
        } while(op != 10);  
        System.out.println("O programa vai terminar...");  
    }  
}
```

```
public static int printMenu() {  
    System.out.println("\nAnalise de uma sequencia de numeros inteiros");  
    System.out.println(" 1 - Ler a sequencia");  
    System.out.println(" 2 - Escrever a sequencia");  
    System.out.println(" 3 - Calcular o maximo da sequencia");  
    System.out.println(" 4 - Calcular o minimo da sequencia");  
    System.out.println(" 5 - Calcular a media da sequencia");  
    System.out.println(" 6 - Detetar se e uma sequencia so constituida por numeros pares");  
    System.out.println("10 - Terminar o programa");  
    System.out.print("Opcao -> ");  
    int op = 0;  
    do {  
        op = read.nextInt();  
        if(!((op > 0 && op < 7) || op == 10)) System.out.print("Opção inválida!\nOpção -> ");  
    } while(!((op > 0 && op < 7) || op == 10));  
    return op;  
}
```

```
public static void readSeq(int[] num_array) {  
    int n = 0, tmp;  
    do {  
        System.out.println("Insira valores:");  
        System.out.printf("Valor[%d]: ", n);  
        tmp = read.nextInt();  
        if(tmp > 0) {  
            num_array[n] = tmp;  
            n++;  
        } else if(tmp < 0) System.out.println("Valor inválido!");  
    } while(tmp != 0);  
}
```

```
public static void printSeq(int[] num_array) {  
    int n = 0;  
    do {  
        if(num_array[n] > 0)  
            System.out.print(" " + num_array[n++] + " ");  
    } while(num_array[n] != 0);  
}
```

```
public static void printMax(int[] num_array) {  
    int n=0, tmp=0;  
    do {  
        if(num_array[n] > tmp) tmp = num_array[n];  
        n++;  
    } while(num_array[n] != 0);  
    System.out.printf("Máximo: %d", tmp);  
}
```

```
public static void printMin(int[] num_array) {  
    int n=0, tmp=Integer.MAX_VALUE;  
    do {  
        if(num_array[n] < tmp) tmp = num_array[n];  
        n++;  
    } while(num_array[n] != 0);  
    System.out.printf("Mínimo: %d", tmp);  
}
```

```
public static void printMed(int[] num_array) {  
    int n=0, tmp=0;  
    do {  
        if(num_array[n] > 0) {  
            tmp += num_array[n];  
            n++;  
        }  
    } while(num_array[n] != 0);  
    tmp /= n;  
    System.out.printf("Média: %d", tmp);  
}
```

```
public static void printIfEven(int[] num_array) {  
    int n=0;  
    boolean onlyEven = true;  
    do {  
        if(num_array[n] % 2 != 0) onlyEven = false;  
        n++;  
    } while(num_array[n] != 0);  
    if(onlyEven) System.out.println("São todos pares");  
    else System.out.println("Existem impares");  
}  
}
```

```

Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituída por numeros pares
10 - Terminar o programa
Opcao -> 1
Insira valores:
Valor[0]: 23
Insira valores:
Valor[1]: 432
Insira valores:
Valor[2]: 16
Insira valores:
Valor[3]: 29
Insira valores:
Valor[4]: 43
Insira valores:
Valor[5]: 0

Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituída por numeros pares
10 - Terminar o programa
Opcao -> 2
23 432 16 29 43
Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituída por numeros pares
10 - Terminar o programa
Opcao -> 3
Máximo: 432
Analise de uma sequencia de numeros inteiros
1 - Ler a sequencia
2 - Escrever a sequencia
3 - Calcular o maximo da sequencia
4 - Calcular o minimo da sequencia
5 - Calcular a media da sequencia
6 - Detetar se e uma sequencia so constituída por numeros pares
10 - Terminar o programa
Opcao -> 10
O programa vai terminar...
Press any key to continue . . .

```


Exercício 6.4

Tendo como base o problema 6.2, considere agora que, dada uma determinada sequência de notas (valores inteiros de 0 a 20), pretende-se construir e desenhar o respetivo histograma (contagem do número de ocorrências de cada nota). O número de notas a processar deverá ser pedido ao utilizador no início do programa. Implemente esse programa, desenhando o histograma com o seguinte aspeto:

Histograma de notas

```
-----  
20 | *****  
19 | *  
.  
.  
.  
01 | *  
00 | **
```

Comece por associar um ”*” a cada nota encontrada, fazendo depois a normalização (linear) do gráfico para que o valor máximo do histograma corresponda a 50 asteriscos.


```

import java.util.*;
public class e6_4 {
    static Scanner read = new Scanner(System.in);
    public static void main (String args[]) {
        int hist_notas[] = new int[21];
        int tmp;
        do {
            System.out.print("Quantas notas pretende introduzir: ");
            tmp = read.nextInt();
        } while (tmp <= 1 && tmp >= 51);
        int notas_in[] = new int[tmp];
        int n = 0;
        do {
            System.out.printf("Nota[%d]: ",n);
            tmp = read.nextInt();
            if(tmp > 0 && tmp <= 20) {
                notas_in[n] = tmp;    n++;
            }
            else System.out.println("Valor invalido!");
        } while(n<notas_in.length);
        n = 0;
        do {
            hist_notas[notas_in[n]]++;
            n++;
        } while(n<notas_in.length);
        System.out.println();
        System.out.println(" Histograma de notas");
        System.out.println("-----");
        n = 20;
        do{
            System.out.printf("%2d | ", n);
            tmp = 0;

```

```

do {
    if(tmp < hist_notas[n]) {
        System.out.print("*");
        tmp++;
    } while(tmp < hist_notas[n]);
    System.out.println();    n--;
} while(n > 0);
}
}

```

```

Quantas notas pretende introduzir: 10
Nota[0]: 11
Nota[1]: 0
Valor invalido!
Nota[1]: 12
Nota[2]: 14
Nota[3]: 18
Nota[4]: 22
Valor invalido!
Nota[4]: 11
Nota[5]: 12
Nota[6]: 18
Nota[7]: 17
Nota[8]: 20
Nota[9]: 17

```

Histograma de notas

```

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

```

Press any key to continue . . . _

Exercício 6.5

Escreva um programa que leia uma sequência de N números reais, sendo o valor N pedido ao utilizador antes do início da introdução dos números. O programa deverá calcular a média e o desvio padrão da sequência e imprimir no ecrã os valores superiores à média.

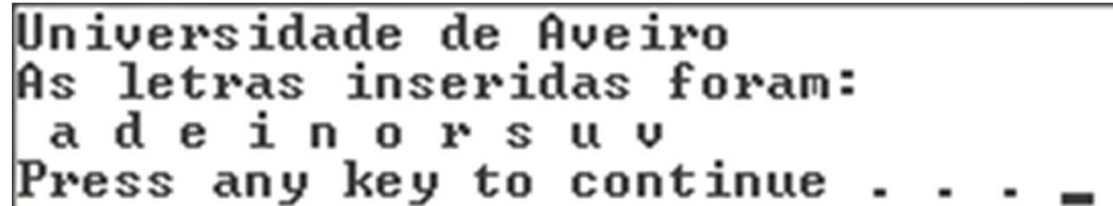
```
import java.util.*;
public class e6_5 {
    static Scanner read = new Scanner(System.in);
    public static void main (String args[]) {
        int n = 0;
        do {
            System.out.print("Quantos valores: ");
            n = read.nextInt();
        } while(n <= 0);
        int num_array[] = new int[n];
        int n_pos = 0;
        do {
            System.out.printf("Valor[%d]: ", n_pos);
            num_array[n_pos++] = read.nextInt();
        } while(n_pos < n);
        double media = 0;
        for(int i = 0; i < n; i++)    media += num_array[i];
        media /= n;
        double desvio = 0;
        for(int i = 0; i < n; i++)    desvio += Math.pow(num_array[i] - media, 2);
        desvio = Math.sqrt(desvio/n-1);
        System.out.printf("Media: %.2f\n", media);
        System.out.printf("Desvio-padrao: %.2f\n", desvio);
        System.out.println("Valores acima da media:");
        for(int i = 0; i < n; i++)
            if(num_array[i] > media) System.out.println(num_array[i]);
    }
}
```

```
Quantos valores: 7
Valor[0]: 1
Valor[1]: 2
Valor[2]: 3
Valor[3]: 4
Valor[4]: 5
Valor[5]: 6
Valor[6]: 7
Media: 4.00
Desvio-padrao: 1.73
Valores acima da media:
5
6
7
Press any key to continue . . .
```

Exercício 6.6

Escreva um programa que leia uma frase e imprima no monitor quais as letras do alfabeto que apareceram nessa frase. Para a resolução deste problema, sugere-se a utilização de uma sequência de valores booleanos de modo a sinalizar quais os caracteres do alfabeto que apareceram pelo menos uma vez.

```
import java.util.*;
public class e6_6 {
    static Scanner read = new Scanner(System.in);
    public static void main (String args[]) {
        char alfa[] = {'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','x','y','w','z'};
        boolean isOne[] = new boolean[26];
        String s;
        s = readString();
        for(int i = 0; i < s.length(); i++)
            for(int j = 0; j < alfa.length; j++)
                if(Character.toLowerCase(s.charAt(i)) == alfa[j]) isOne[j] = true;
        System.out.println("As letras inseridas foram:");
        for(int i = 0; i < isOne.length; i++)
            if(isOne[i]) System.out.print(" " + alfa[i]);
        System.out.println();
    }
    public static String readString() {
        String s = new String();
        s = read.nextLine();
        return s;
    }
}
```



```
Universidade de Aveiro
As letras inseridas foram:
a d e i n o r s u v
Press any key to continue . . . _
```

Exercício 6.7

Considere agora um caso genérico do problema 6.2, e escreva um programa que conte o número de vezes que cada elemento ocorre numa sequência de números inteiros. A leitura deve terminar após a introdução de 100 números ou com o aparecimento de um número negativo.

Considere o seguinte exemplo. Supondo a sequência

`a = {4, 2, 5, 4, 3, 5, 2, 2, 4}`

o resultado do programa devia ser o seguinte:

```
4 ocorre 3 vezes
2 ocorre 3 vezes
5 ocorre 2 vezes
3 ocorre 1 vez
```

```
4 2 5 4 3 5 2 2 4
-1
4 ocorre 3 vezes
2 ocorre 3 vezes
5 ocorre 2 vezes
3 ocorre 1 vezes
Press any key to continue . . . _
```

```
import java.util.*;
public class e6_7 {
    static final int N = 100;
    static final Scanner read = new Scanner(System.in);
    public static void main (String args[]) {
        int array[] = new int[N];
        int r, count, length=0;
        for (r=0; r<array.length; r++)
        {   array[r] = read.nextInt();
            if (array[r] < 0) break;   }
        length = r;
        for(int i=0; i<length-1; i++)
        { count = 0;
            for(int j=0; j<array.length; j++)
            {
                if ( (array[i] == array[j]) && (j<i) ) break;
                if (array[i] == array[j]) count++;
            }
            if (count == 0) continue;
            System.out.println(array[i]+" ocorre "+count+" vezes");
        }
    }
}
```

```
import java.util.*;
public class e6_7_no_break {
    static final int N = 100;
    static final Scanner read = new Scanner(System.in);
    public static void main (String args[]) {
        int array[] = new int[N];
        int r, count, length=0;
        for (r=0; (r<array.length) && ((array[r] = read.nextInt()) >= 0); r++);
        length = r;
        for(int i=0; i<length-1; i++)
        { count = 0;
            for(int j=0; (j<array.length) && (( array[i] == array[j]) && (j<i) ) == false); j++)
                if (array[i] == array[j]) count++;
            if (count != 0) System.out.println(array[i]+" ocorre "+count+" vezes");
        }
    }
}
```

```

import java.util.*;
public class labels {
    static final int N = 100;
    static final Scanner read = new Scanner(System.in);
    public static void main (String args[]) {
        int array[] = new int[N];
        int r, count, length=0;
        for (r=0; (r<array.length) && ((array[r] = read.nextInt()) >= 0); r++);
        length = r;
        label1: for(int i=0; i<length-1; i++)
        { count = 0;
            for(int j=0; (j<array.length) && (( array[i] == array[j]) && (j<i) ) == false); j++)
            { if (array[i] == array[j]) count++; if (count == 2) break label1; }
            if (count != 0) System.out.println(array[i]+" ocorre "+count+" vezes");
        }
    }
}

```

É permitido mas é melhor evitar

```

1 2 3 4 5 6 7 4 8 -1
1 ocorre 1 vezes
2 ocorre 1 vezes
3 ocorre 1 vezes
Press any key to continue . . . _

```