Básicos e Estruturas de Controlo

1. Programa que receba 4 valores inteiros positivos (o programa não deverá permitir valores negativos) de uma caixa de texto (correspondendo aos valores de venda de uma empresa nos anos se 2000, 2001, 2002, 2003) e, ao premir um botão apresente:
2. Se algum dos valores de vendas em algum ano foi mais do dobro do que no ano anterior.

label2.Text = "";

int n1 = Convert.ToInt32(textBox1.Text);

int n2 = Convert.ToInt32(textBox2.Text);

int n3 = Convert.ToInt32(textBox3.Text);

int n4 = Convert.ToInt32(textBox4.Text);

if (n2>(n1\*2))

{

label2.Text += "Em 2001 aumentou mais do dobro de 2000 \n";

}

if (n3>(n2\*2))

{

label2.Text += "Em 2002 aumentou mais do dobro de 2001 \n";

}

if (n4>(n3\*2))

{

label2.Text += "Em 2003 aumentou mais do dobro de 2002 \n";

}

if(label2.Text=="")

{

label2.Text += "Não houve aumentos de dobro \n";

}

1. Se o valor das vendas teve um crescimento constante, se decresceu todos os anos, ou se foi variável.

if (n1 < n2 && n2 < n3 && n3 < n4)

{

label4.Text = "crescente";

}

else if (n1 > n2 && n2 > n3 && n3 > n4)

{

label4.Text = "decrescente";

}

else

{

label4.Text = "variavel";

}

1. A média (double) das vendas nestes quatro anos.

double media=0.0;

media = (n1 + n2 + n3 + n4) / 4;

media = Math.Round(media, 2);

label6.Text = media.ToString();

1. O crescimento percentual 2000-2001 2001-2002 2002-2003

label8.Text = "";

double num1 = ((n2-n1) / n1)\*100;

double num2 = ((n3 - n2) / n2)\*100;

double num3 = ((n4-n3) / n3)\*100;

label8.Text += "2000-2001 -> " + num1 + "\n";

label8.Text += "2001-2002 -> " + num2 + "\n";

label8.Text += "2002-2003 -> " + num3 + "\n";

1. O ano onde houve um maior crescimento em valor absoluto

double nume1 = (n2 - n1);

double nume2 = (n3 - n2);

double nume3 = (n4 - n3);

if (nume1 > nume2 && nume1 > nume3)

label10.Text = "O ano em que mais cresceu foi 2000-2001";

else if (nume2 > nume1 && nume2 > nume3)

label10.Text = "O ano em que mais cresceu foi 2001-2002";

else if (nume1 == nume2 && nume2 == nume3)

label10.Text = "O crescimento foi igual";

else

label10.Text = "O ano em que mais cresceu foi 2002-2003";

1. Programa que leia as notas de um aluno (inteiros) a 3 momentos de avaliação e calcule a nota final em que a melhor nota terá um peso de 50%, tendo as outras duas notas um peso de 25% cada. O programa deverá indicar se o aluno passou (média superior a 9,5 valores)

double media;

int av1 = Convert.ToInt32(textBox1.Text);

int av2 = Convert.ToInt32(textBox2.Text);

int av3 = Convert.ToInt32(textBox3.Text);

if (av1 > av2 && av1 > av3)

{

media = (av1 \* 0.50) + (av2 \* 0.25) + (av3 \* 0.25);

}

else if (av2 > av1 && av2 > av3)

{

media = (av1 \* 0.25) + (av2 \* 0.50) + (av3 \* 0.25);

}

else

{

media = (av1 \* 0.25) + (av2 \* 0.25) + (av3 \* 0.50);

}

if (media > 9.5)

{

label1.Text = "passou";

}

else

{

label1.Text = "Nao passou";

}

1. Programa que leia 2 números inteiros do utilizador e indique se algum dos números é igual ao outro número elevado ao quadrado ou ao cubo. Deverá dar esta indicação na saída. Também deverá indicar se os números forem iguais.
2. Programa que permita ao utilizador introduzir 3 números ***double*** em 3 caixas de texto. Ao ser premido um botão o programa deverá indicar se os 3 números têm a mesma parte inteira. Exemplo: Os números

* 2.39 | 2.12 | 2.54 têm a mesma parte inteira.
* Os números 3.5 | 3.8 | 4.1 não têm a mesma parte inteira.

label1.Text = "";

double n = Convert.ToDouble(textBox1.Text);

double n1 = Convert.ToDouble(textBox2.Text);

double n2 = Convert.ToDouble(textBox3.Text);

n = Math.Truncate(n);

n1 = Math.Truncate(n1);

n2 = Math.Truncate(n2);

if(n==n1 && n==2)

{

label1.Text = "IGUAIS";

}

else

{

label1.Text = "Diferentes";

}

1. Crie um programa que leia 3 notas de um aluno (0-20): P, M, T e indique se o aluno passou à disciplina. O programa não deverá permitir valores fora do intervalo. Para passar à disciplina o aluno deverá ter uma nota igual ou superior a 10 nas 3 disciplinas. Deverá ser realizado um exame oral se alguma das notas for 4 ou mais valores superior a qualquer uma das outras. O programa deverá indicar se o aluno passou e, neste caso, se tem de realizar exame oral.
2. 1. Coloque 3 caixas de texto e um botão numa aplicação. O utilizador introduzirá 3 valores inteiros nas caixas de texto. Ao ser premido o botão, o programa indicará se algum dos valores é igual à soma ou o produto dos outros dois.
3. Crie uma aplicação com duas caixas de texto onde o utilizador vai introduzir dois números inteiros. Os números introduzidos deverão ser positivos. Ao ser premido um botão, o programa deverá indicar todos os números pares não divisíveis por 10 existentes entre esses dois números (inclusive).

**Leitura de valores num programa controlado por eventos.**

1. Crie um programa que leia valores de uma caixa de texto e indique se os valores introduzidos são iguais, maiores ou menores que os valores anteriores.
2. Crie um programa que indique qual o maior valor introduzido até à altura.
3. Crie um programa que conte quantos valores negativos, pares e ímpares são introduzidos numa caixa de texto. Deverá parar quando atingir os 5 valores negativos introduzidos.
4. [flag] Programa que permita ler 5 números inteiros a partir de uma caixa de texto. No fim deverá indicar se todos os números são maiores ou iguais ao anterior, ou seja se foram introduzidos de forma crescente. Também deverá dar uma indicação no caso de os números serem todos iguais.
5. Escreva um programa que leia 10 inteiros do utilizador e indique qual o maior número par introduzido.
6. [flag] Escreva um programa que leia 10 inteiros do utilizador e indique no fim se os números são todos pares.
7. Crie um programa que lerá números inteiros sucessivamente de uma caixa de texto terminando quando um número for igual à soma dos dois números anteriores introduzidos. Não há qualquer restrição aos números introduzidos. Naturalmente, o programa só poderá terminar a partir do 3º número introduzido.
8. Crie um programa que leia sucessivamente números inteiros do utilizador e apresente a média dos 3 últimos números introduzidos, incluindo o próprio número. O programa termina quando for introduzido o valor -1.
9. Crie uma aplicação onde o utilizador introduzirá sequencialmente valores inteiros (obrigatoriamente positivos) numa caixa de texto. O programa terminará quando for introduzido um quadrado perfeito ou quando o valor introduzido for igual ao dobro do anterior. O programa deverá dar a indicação do motivo porque terminou.
10. Crie uma aplicação em que o utilizador introduza sequencialmente valores (sem restrições) numa caixa de texto. O programa terminará quando for introduzido um valor múltiplo do primeiro valor introduzido pelo utilizador.

**Ciclos**

1. Programa que contenha 3 caixas de texto e um botão. Na primeira caixa de texto será introduzido um número inteiro ***inicio***, na segunda um número inteiro ***fim*** e na 3ª um número inteiro ***inc***. Ao ser premido o botão o programa deverá mostrar num *label* todos os números inteiros entre ***inicio*** e ***fim*** inclusive com um incremento igual ao número inteiro ***inc***. O programa não deverá permitir que o valor de fim seja superior ao valor inicio ou que o valor ***inc*** seja negativo ou zero.
2. Crie um programa para apresentar os primeiros 10 termos de uma progressão geométrica. O utilizador deverá introduzir um número *double* entre 1 e 4, inclusive. A progressão deverá começar em 1, sendo que cada termo sucessivo deverá ser igual ao anterior multiplicado pelo termo da progressão. Exemplo para um valor de 1,5:

|  |
| --- |
| 1 |
| 1,5 |
| 2,25 |
| 3,375 |
| 5,0625 |
| 7,59375 |
| 11,390625 |
| 17,0859375 |
| 25,62890625 |
| 38,44335938 |

double n = Convert.ToDouble(textBox1.Text);

double r = 1;

for(int i=0;i<10;i++)

{

label1.Text += r.ToString()+"\n";

r = r \* n;

}

1. Crie um programa que leia um número inteiro positivo de uma caixa de texto e apresente todos os números pelos quais o número é divisível. Se o número for primo o programa deverá dar esta indicação.

label1.Text = "";

int n = Convert.ToInt32(textBox1.Text);

int cont = 0;

for(int i=1;i<=n;i++)

{

if(n%i==0)

{

label1.Text += i + "\n";

cont++;

}

}

if(cont==2)

{

label1.Text = "É um numero primo";

}

1. Escreva um programa em que o utilizador introduza números até introduzir dois números primos seguidos.
2. Crie um programa que leia 10 números inteiros do utilizador e no fim indique quantos são quadrados perfeitos. Os números deverão estar entre 0 e 1000.
3. Crie um programa que leia um valor inteiro positivo do utilizador e indique qual é menor número múltiplo de 5 superior ao número introduzido.
4. Crie um programa que leia do utilizador um número inteiro positivo n, e apresente os primeiros n números primos.
5. Crie um programa que leia sucessivamente números inteiros do utilizador de uma caixa de texto e indique se este número faz parte da série de Fibonnaci. O programa terminará quando o número introduzido for negativo.
6. Crie um programa que escreva todos os números inteiros entre 0 e 100 que não sejam múltiplos de 7 nem incluam o algarismo 7. Assim, os números 7, 14, 17, 21, 27 etc. não deverão ser apresentados.

Métodos

1. Crie um método que receba um número inteiro n como argumento e retorne a soma dos primeiros n inteiros. Por exemplo se receber o valor 4 deverá retornar o valor 10 (1+2+3+4). Teste o método com um programa onde o valor é introduzido pelo utilizador.

int soma(int a)

{

int soma = 0;

for (int i = 1; i <= a; i++)

soma += i;

return soma;

}

private void button1\_Click(object sender, EventArgs e)

{

int n = Convert.ToInt32(textBox1.Text);

int res=soma(n);

label1.Text = res + " ";

}

}

1. Crie um método que receba 3 valores inteiros a,b,c como argumentos e retorne um valor aleatório *double* que se deverá encontrar entre os dois maiores valores. Exemplo: se o programa receber os valores 5, 8 e 3 deverá retornar um valor double aleatório entre 5 e 8. Teste o método com um programa onde os valores são introduzidos pelo utilizador.

double aleatorio(double a,double b,double c)

{

Random random = new Random();

List<double> arr = new List<double>();

arr.Add(a);arr.Add(b); arr.Add(c);

double maximum = arr.Max();

arr.Remove(maximum);

double minimum = arr.Max();

double al = Math.Round(random.NextDouble() \* (maximum - minimum) + minimum, 2);

al = Math.Round(al, 2);

return al;

}

private void button1\_Click(object sender, EventArgs e)

{

double n1 = Convert.ToDouble(textBox1.Text);

double n2 = Convert.ToDouble(textBox2.Text);

double n3 = Convert.ToDouble(textBox3.Text);

double res = aleatorio(n1,n2,n3);

label1.Text = res.ToString();

}

1. Crie um método que receba um número inteiro como argumento e retorne a soma dos primeiros n números inteiros positivos não primos.

int primo(int n)

{

int i;

if (n == 1) return 0;

for (i = 2; i \* i <= n; i++)

{

if (n % i == 0) return 0;

}

return 1;

}

int soma(int a)

{

int soma = 0;

for (int i = 2; i <= a; i++)

{

if(primo(i)==0)

{

soma += i;

}

}

return soma;

}

private void button1\_Click(object sender, EventArgs e)

{

int n = Convert.ToInt32(textBox1.Text);

int res = soma(n);

label1.Text = res + " ";

}

1. Crie um método que receba quatro valores inteiros como argumentos e retorne o valor que tiver o maior algarismo das unidades. No caso de haver mais de um número com o maior valor das unidades, deverá retornar o maior valor absoluto entre os referidos números. Teste com um programa onde os valores são introduzidos em caixas de texto.
2. Crie um método que retorne um número primo aleatório entre dois valores apresentados como argumentos.

int primo(int n)

{

int i;

if (n == 1) return 0;

for (i = 2; i \* i <= n; i++)

{

if (n % i == 0) return 0;

}

return 1;

}

double aleatorio(int a,int b)

{

Random random = new Random();

int maximum = Math.Max(a, b);

int minimum = Math.Min(a, b);

bool flag = false;

double al=0;

while (flag==false)

{

al = Math.Round(random.NextDouble() \* (maximum - minimum) + minimum, 0);

int t = Convert.ToInt32(al);

if (primo(t)==1)

{

flag = true;

}

}

return al;

}

private void button1\_Click(object sender, EventArgs e)

{

int n = Convert.ToInt32(textBox1.Text);

int n1 = Convert.ToInt32(textBox2.Text);

double res =aleatorio(n,n1);

label1.Text = res + " ";

}

1. Crie um método que retorne uma letra minúscula aleatória (char).

char aleatorio()

{

string abc = "qwertyuioplkjhgfdsazxcvbnm";

Random random = new Random();

int maximum = abc.Length;

int minimum = 0;

double al=0;

al = Math.Round(random.NextDouble() \* (maximum - minimum) + minimum, 0);

int t = Convert.ToInt32(al);

char letra = abc[t];

return letra;

}

private void button1\_Click(object sender, EventArgs e)

{

char res =aleatorio();

label1.Text = res + " ";

}

1. Crie um método que gere um número inteiro entre 10 e 40, *excepto* números entre 20 e 30, inclusive.

int aleatorio()

{

Random random = new Random();

int maximum = 40;

int minimum = 10;

double al=0;

int t=0;

bool flag = false;

while(flag==false)

{

al = Math.Round(random.NextDouble() \* (maximum - minimum) + minimum, 0);

t = Convert.ToInt32(al);

if(t>20 && t<30)

{

flag = false;

}

else

{

flag = true;

}

}

return t;

}

private void button1\_Click(object sender, EventArgs e)

{

int res =aleatorio();

label1.Text = res + " ";

}

1. Crie um método que receba dois valores inteiros como argumento e determine se são números primos gémeos. Dois números primos dizem-se gémeos se a sua diferença for 2. Os primeiros pares de números primos gémeos são (3,5), (5,7), (11,13), (17, 19), (29, 31). O método deverá retornar um booleano e ser testado através da introdução de valores em caixas de texto.

int primo(int n)

{

int i;

if (n == 1) return 0;

for (i = 2; i \* i <= n; i++)

{

if (n % i == 0) return 0;

}

return 1;

}

bool gemeos(int a,int b)

{

if (primo(a) == 1 && primo(b) == 1)

{

if (Math.Max(a, b) - Math.Min(a, b) == 2)

{

return true;

}

else

{

return false;

}

}

else

return false;

}

private void button1\_Click(object sender, EventArgs e)

{

int n = Int16.Parse(textBox1.Text);

int n2 = Int16.Parse(textBox2.Text);

if(gemeos(n,n2)==true)

label1.Text = "GEMEOS PRIMOS";

else

label1.Text = "NAO GEMEOS";

}

Vectores

1. Crie um programa que permita ao utilizador preencher um vector de inteiros com 5 posições. Todos os valores deverão ser positivos. O programa não deverá permitir a introdução de mais do que 2 números primos. Se o utilizador tentar introduzir um terceiro número primo, o programa deverá apresentar uma mensagem e não permitir a introdução do número. No fim deverá apresentar o vector e indicar quantos números primos existem no vector.
2. Crie um programa onde o utilizador introduza valores inteiros numa caixa de texto. O programa deverá terminar quando um (qualquer) dos valores for introduzido pela 3ª vez (consecutiva ou não).
3. Crie um programa que preencha um vector com os 100 primeiros números inteiros não múltiplos de 5. Ao ser premido um botão deverá indicar a soma desses valores. 1 2 3 4 6 7 8 9 11 12 13 14 16..
4. Crie um programa que preencha um vector com os primeiros 100 números inteiros positivos não primos e calcule a sua soma.
5. Escreva um método SomaNegativos() que receba um vector de inteiros como argumento e retorne a soma dos valores negativos do vector.
6. Crie um método Vezes() que receba um valor inteiro e um vector de inteiros como argumento e retorne o número de vezes que o valor surge no vector.
7. Escreva um método que determine se, numa matriz de 4X4 obtida aleatoriamente com valores inteiros entre 1 e 5, todas as linhas e colunas somam o mesmo valor.
8. Escreva um programa que apresente na saída todos os números múltiplos de 5 existentes num vector de inteiros. Teste com um vector preenchido aleatoriamente com valores entre 0 e 100.
9. Crie um programa que permita ler um vector de inteiros de 10 posições do utilizador. O programa deverá apenas permitir a introdução de valores positivos e negativos alternados, ou seja, o programa não deverá permitir que o utilizador introduza dois valores positivos seguidos ou dois valores negativos seguidos. Se o utilizador não introduzir valores alternados o programa dará uma mensagem. No fim da introdução o programa apresentará o vector e deverá indicar se a média dos valores introduzidos é positiva ou negativa. Não é necessário indicar o valor da média.
10. Crie um programa que leia um algarismo **val** do utilizador (entre 1 e 9 inclusive) e preencha um vector de 10 posições com os primeiros 10 números inteiros que obedeçam a uma das seguintes condições:
    * 1. Sejam múltiplos de **val**
      2. Contenham o algarismo **val**.
11. Crie um método que receba um vector de inteiros como argumento e retorne um valor booleano true se todos os elementos do vector forem pares ou se forem todos negativos (inferiores a zero). Teste o método com um vector declarado e inicializado no próprio programa.
12. Crie um método que receba um vector de inteiros do utilizador e retorne o valor inteiro 1 se o vector estiver ordenado em ordem crescente, -1 se estiver ordenado em ordem decrescente e 0 se não estiver ordenado.
13. Crie um método que receba um vector de inteiros (**vec**) e um inteiro (**val**) como argumentos e retorne o valor booleano true se o vector contiver apenas múltiplos do valor inteiro val.
14. Crie um programa que permita ler um vector de inteiros de 10 posições do utilizador. O programa deverá apenas permitir a introdução de valores crescentes, ou seja, o programa não deverá permitir que o utilizador introduza um valor que seja igual ou inferior ao anterior. Se o utilizador não introduzir valores crescentes o programa dará uma mensagem. No fim da introdução o programa apresentará o vector.
15. Crie um método que receba um vector de inteiros como argumento e converta todos os valores negativos para zero e os valores positivos para o múltiplo de 5 mais próximo.
16. Crie um programa que permita procurar um valor num vector. O vector deverá estar declarado e inicializado no programa. O programa lerá um valor inteiro do utilizador através de uma caixa de texto. O programa deverá indicar se encontrou o valor, quantas vezes o valor ocorre no vector. Se o valor não for encontrado deverá apresentar o valor mais próximo encontrado.
17. Crie e teste um método que receba um vector de inteiros como argumento e retorne o número de elementos do vector cujo valor seja superior à moda do vector.
18. Crie um programa que preencha um vector com 10 números inteiros aleatórios entre 0 e 1000, inclusive. A única restrição é a de que os números não podem ter algarismos repetidos. Por exemplo, os valores 33, 131, 447, 777 não deverão ser aceites. Atenção: os valores 5 e 50 (por exemplo) deverão ser aceites, pois não são considerados zeros à esquerda. Deverá apresentar o vector no final.
19. Crie um programa em que o utilizador introduza valores inteiros num vector de 10 posições, terminando a introdução quando o utilizador introduzir os dez valores, quando a soma dos valores introduzidos for igual ou superior a 100 ou quando o utilizador introduzir o valor zero (o que acontecer primeiro). O programa deverá indicar a razão pela qual terminou a introdução. O último valor introduzido deverá fazer parte do vector. Os valores restantes deverão ter o valor zero. No final o programa deverá apresentar o vector.
20. Crie um programa que ordene um vector de inteiros pela soma dos algarismos de cada valor inteiro. Não é necessário introduzir o vector pelo interface gráfico.

Strings

1. Crie um programa que leia 2 strings do utilizador e, ignorando maiúsculas e minúsculas, indique:
   * 1. Se têm caracteres em comum.
     2. Quantos caracteres têm em comum.
     3. Quais os caracteres que têm em comum
     4. Se têm todos os caracteres em comum.
2. Crie um programa onde esteja declarado um vector de strings (com 5 a 10 elementos) e o apresente ordenado na saída de acordo com o tamanho das strings. Valorização: Apresente também o vector ordenado de acordo com o número de letras maiúsculas em cada string.
3. Crie um programa que leia 4 nomes (strings) do utilizador e indique quais das strings começa por um algarismo.
4. Crie um método Elimina(…) que receba uma string como argumento e retorne a mesma string eliminando a primeira letra, a última letra e os espaços.
5. Crie um programa que leia o nome completo do utilizador (usando uma caixa de texto) e indique qual dos nomes tem um comprimento superior.
6. Crie um método MaisMaiusculas que receba uma string como argumento e retorne o valor booleano true se a string contiver mais letras maiúsculas do que minúsculas. Teste com uma string introduzida pelo utilizador.
7. Crie um programa que leia uma string do utilizador e a escreva duplicando todas as palavras. Exemplo:

*Entrada: “Olá senhor João!”*

*Saída: “Olá Olá senhor senhor João João!”*

O programa também deverá escrever a string na ordem correcta seguida pela ordem inversa de palavras. Exemplo:

*Entrada: “Olá senhor João”*

*Saída: “Olá senhor João João senhor Olá”*

1. Crie um método que receba uma string como argumento e retorne um valor inteiro correspondendo ao número de dígitos (algarismos) existentes na string. Use este método e o “Bubble Sort” para ordenar um vector de strings de acordo com o número de algarismos presentes nas strings. Não é necessário introduzir o vector de strings pelo interface gráfico.
2. Crie um método que receba uma string (correspondendo a um nome pessoal) como argumento e retorne uma string correspondendo ao último nome da pessoa, totalmente em maiúsculas. Por exemplo, se a entrada for: “João Passos Dias Aguiar Mota”, o valor retornado será “MOTA”. Use este método para ordenar um vector de strings alfabeticamente pelo último nome.
3. Crie uma aplicação em que o utilizador introduza o nome completo através de uma caixa de texto. A aplicação deverá indicar:
   1. Por quantas palavras é constituído o nome.
   2. O maior nome (mais comprido) do utilizador. Se houver “empates” deverá indicar os nomes com maior comprimento.
   3. O nome que se encontra primeiro na ordem alfabética.
   4. Se algum dos nomes rima com “Zebedeu”. Se houver mais do que um nome deverá dar a indicação. Se não houver deverá indicar “nenhum nome rima com “Zebedeu”.
   5. O nome completo do utilizador na ordem correcta, mas em que cada nome venha invertido. “João Passos Dias Aguiar Mota” será “Oãoj SossaP Said Raiuga Atom”. Repare que a primeira letra de cada nova palavra deverá ser maiúscula e as restantes minúsculas.

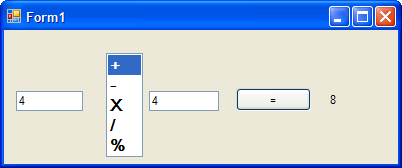
Interface gráfico

1. Crie um programa com um pergunta de resposta múltipla com um limite de tempo de 10 segundos. Deverá ter uma barra de progresso. Ao fim dos 10 segundos o programa deverá impedir a resposta.
2. Crie um programa com dois botões de START e STOP. O utilizador deverá premir START e depois o STOP ao fim de exactamente 10 segundos. O programa não dá qualquer indicação sobre o tempo decorrido. No fim deverá indicar quanto o utilizador se desviou dos 10 segundos. Poderá criar um jogo para 2 ou mais utilizadores.
3. Crie uma aplicação que permita movimentar uma fotografia nas 4 direcções até aos limites da form.

Exercícios de exame

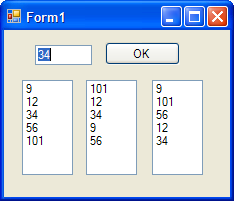
**Exercício 1**

(1) Crie uma aplicação que contenha duas caixas de texto para introdução de números, uma “List Box”, um *label* e um botão (com o texto “=”). A List Box deverá conter os operadores aritméticos + - X / %.

(2) O utilizador deverá introduzir números inteiros maiores ou iguais a zero nas duas caixas de texto. Deverá ainda seleccionar um dos operadores na List Box. Ao ser premido o botão “=” deverá surgir no label o resultado da operação.

(1) A aplicação deverá apenas permitir a introdução de números inteiros maiores ou iguais a zero e menores ou iguais a 100 (cem). Em caso contrário deverá apresentar uma mensagem.

(1) No caso do segundo operando ser nulo (zero) na divisão, também deverá ser dada uma mensagem.

**Exercício 2**

Crie uma aplicação que leia, do utilizador, números inteiros sucessivamente até ser introduzido o valor -1. A introdução poderá ser feita através de um botão ou premindo “ENTER”.

Em cada introdução, o programa deverá apresentar o conjunto dos números ordenados de três maneiras diferentes em 3 labels (ou 3 listas):

1. (1) De forma crescente.
2. (2) Pelo valor da soma dos algarismos.
3. (2) Pelo resto da divisão inteira do número por 9.

**Exercício 3**

Crie um método que receba uma *string* (que poderá ter várias palavras) e retorne o valor booleano true no caso de a *string* ter mais do que uma palavra iniciada pela mesma letra (ignorando maiúsculas e minúsculas). Teste o método com uma string introduzida pelo utilizador numa caixa de texto. Se o utilizador introduzir uma *string* vazia, o programa deverá apresentar uma mensagem e não prosseguir. Por exemplo, a string “A menina comeu a maçã”, tem dois pares de palavras começadas pela mesma letra.

**Exercício 4**

Crie um método que leia 5 *strings* (nomes completos de alunos) do utilizador (através de caixas de texto) e as apresente ordenadas das seguintes maneiras:

a) (1) Pelo número de nomes que contêm.

b) (2) Alfabeticamente pelo último nome.

c) (2) Pelo tamanho do maior nome na *string*.

Nota: é indiferente a apresentação de forma ascendente ou descendente.