



QB Neste guião irás aprender **algumas funções pré-definidas** em **QBasic** que te vão **ajudar** muito ao **longo** deste **curso** de **programação**.

QB **Abre o Microsoft QuickBasic 4.50**

QB **Digita as seguintes linhas de código:**

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB
File Edit View Search Run Debug Calls Options Help
Untitled
CLS
DO
PRINT "Programação"
LOOP UNTIL INKEY$ = "s"
```

Corre o programa e para a sua **execução** pressionado a tecla **"s"**.

QB **Guarda este programa** como **inkey.bas**

QB **Cria um novo programa.**

QB Neste **programa** cria o **seguinte menu** (com **screen 13**)

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB
*****
*          1 - Jogar          *
*          2 - Sair          *
*****
```

QB Pretende-se que **utilizes** uma **estrutura** de **repetição** que **enquanto não pressionarem a tecla 2** fica à **espera** de ler outra **tecla**. Se **pressiona a tecla 2** sai da **estrutura** de **repetição** e **termina o programa**. Se **pressiona a tecla 1** limpa o **ecrã** e diz **"Bem-vindo ao jogo"**.

QB **Guarda o teu programa** como **inloop.bas**

QB **Cria um novo programa.**

QB **Copia o seguinte código** para o teu **IDE**.

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB
File Edit View Search Run Debug Calls Options Help
Untitled
CLS
DIM nome AS STRING
nome = "Programação"
PRINT LEN(nome)
```

A função **LEN** conta o **número** de **carateres** da **STRING** que está **dentro** da **variável nome**.

QB **Altera a palavra programação** para o **teu nome** e verifica o **número** de **letras** que **contém**.

QB **Altera para o teu nome completo**. Pretende-se que **cortes** a **string** com recurso à função **LEFT\$** para que fique **apenas** o **teu primeiro nome**.

Fica a **Sintaxe**:

LEFT\$ (STRING, nº de carateres desde a esquerda até cortar)

QB Utiliza agora a função **RIGHT\$** para conseguires mostrar no **ecrã** **apenas** o **teu último nome**.

Fica a **sintaxe**:

RIGHT\$ (STRING, nº de carateres desde a direita até cortar)



QB Finalmente, utiliza a função **MID\$** para mostrares **apenas** o teu **nome** do **meio**.

Fica a **sintaxe**:

MID\$ (STRING, nº de caracteres desde a esquerda até cortar +1, nº de caracteres a mostrar a partir do corte)

QB Guarda o teu programa como **strcut.bas**

QB Cria um novo programa. Copia o seguinte código.

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB
File Edit View Search Run Debug Calls Options Help
CONVINT.BAS
CLS
DIM x AS SINGLE ' definir variável para números decimais
x = 4.643
PRINT x
PRINT INT(x) ' imprime apenas a parte inteira do nº sem arredondar
```

A função **INT** devolve a **parte inteira** do número.

QB Acrescenta funcionalidades. Cria **duas variáveis** do **tipo single**.

Faz com que essas **variáveis** sejam **preenchidas** com **valores solicitados** ao **utilizador** da **seguinte forma**:

Qual o valor ao qual quer aplicar raiz quadrada:

Aqui deves **utilizar** a função **SQR** para **obter** o **valor** da **raiz quadrada** e **imprimir** o **resultado** no **ecrã**

Qual o valor que quer converter em positivo:

Aqui deves **aplicar** a função **ABS** para **converter** o **nº** em **positivo** e **imprimir** o **resultado** no **ecrã**

QB Guarda o teu programa como **opnum.bas**

QB Cria um novo programa.

QB Neste programa pretende-se **saber** se um **número** é **par** ou **ímpar**. Existe uma **função** que **devolve 0** se um **determinado número** for **divisível** por **outro**. Essa função é a **MOD**. Por exemplo:

16 MOD 4 dá 0 porque o 16 é divisível por 4

17 MOD 4 não dá 0 logo não é divisível por 4

QB Neste programa deves criar uma **estrutura** de **decisão** que **verifique** se o **valor introduzido** pelo **utilizador** é **divisível** por 2 ou **não**. Caso **seja** dizes que o **número introduzido** é **par** caso contrário **“Número ímpar”**.

QB Guarda o teu programa como **odd.bas**

QB Cria um novo programa.

QB Neste programa vais **aprender** a **sortear números (aleatórios)**.

QB Começa por **copiar** as **seguintes linhas** de **código** para o teu **IDE**.

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB
File Edit View Search Run Debug Calls Options Help
RND.BAS
CLS
PRINT RND * 10
—
SLEEP
```



QB Este programa utiliza o comando **RND** que tem a seguinte **sintaxe**:

RND * N^o max a sortear (exclusive)

Quer dizer que **sorteia** um **número** entre **0** (a começar no 1 se for inteiro) e o **número** imediatamente **anterior** ao **max**

QB Altera o programa para que **sorteie um número** até **19 exclusive**.

QB Faz com que seja **mostrada apenas a parte inteira** do **número sorteado**.

QB Se reparares o valor **sorteado** é **sempre o mesmo**. Para que dê **valores diferentes** utiliza a seguinte **expressão** antes de **utilizares** o **RND**.

Randomize timer

Assim os números serão baseados no **relógio interno** do computador **gerando sempre RND diferentes**.

QB Guarda o programa como **RND.bas**

QB Abre o programa **odd.bas**

QB Guarda como **oddrnd.bas**

QB Em vez de serem **solicitados números** ao **utilizador** é **questionado** o **seguinte** ao **utilizador**:

Quer sortear um valor entre 1 e que número?

Conforme o valor **introduzido** deve ser **sorteado** um **valor** e deve dar a **seguinte informação** ao **utilizador**:

O número que saiu no sorteio foi o X e trata-se de um número ímpar/par.

QB Guarda o programa.

QB Cria um **novo programa**.

QB O **RND** permite sortear números **sem começar** sempre **no 0**. Imagina que querias **começar** a **sortear** a **partir** do número **5** até ao **10(exclusive)**, como farias? Bem poderá ser assim?

Int(rnd*10) + 5

Se analisarmos o sorteio é de **0** a **10**, mas acrescentamos **5** ao **resultado**. Vamos ver se **funciona**. Quando sai:

RND*10	+5
0	5
1	6
2	7
3	8
4	9
5	10
6	11
7	12
8	13
9	14



Se **reparares** até **começa bem**. Quando sai **0** na **verdade** mostraria **5** no **ecrã**. O problema é quando chega a **5** mostra **10** que já é **exclusive**. O que fazer?

$\text{Int}(\text{RND} * \Delta_{\text{max_min}}) + \text{min}$

A solução seria: $\text{int}(\text{RND} * 5) + 5$. Experimenta.

- QB** Altera o **programa** para que **consigas sortear** números de **55 a 75 (exclusive)**.
- QB** Guarda o **programa** como **rndmeio.bas**
- QB** Chama o teu **professor** para **avaliar**.