

# Programação RETR0

QBasic

PSINF

07  
MISC



# Funções/comandos predefinidas

- ❑ O Qbasic já disponibiliza várias funções (e comandos), prontas a utilizar algumas das quais iremos aprofundar neste capítulo
- ❑ Podemos, contudo criar as nossas próprias funções, mas isso será noutra capítulo!

# SLEEP

- ❑ Este comando permite parar a execução da próxima linha durante X segundos
- ❑ Sintaxe:

`SLEEP` n° de segundos

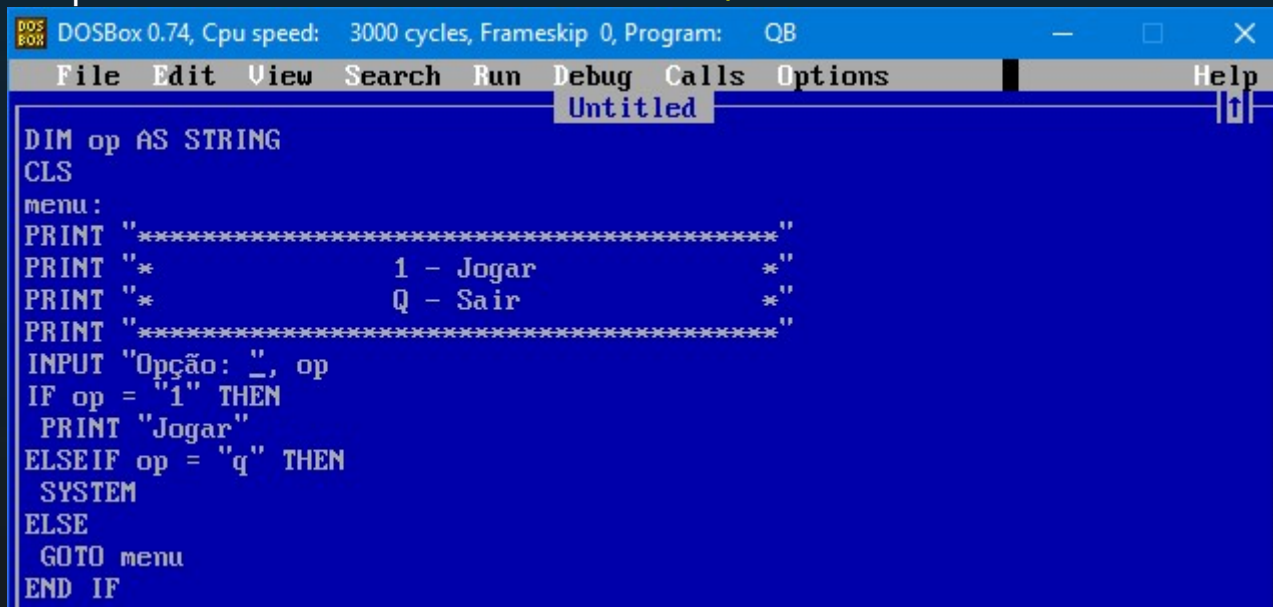
Exemplos:

`SLEEP 2` ` espera 2 segundos antes de avançar

`SLEEP 10` ` espera 10 segundos antes de avançar

# GOTO

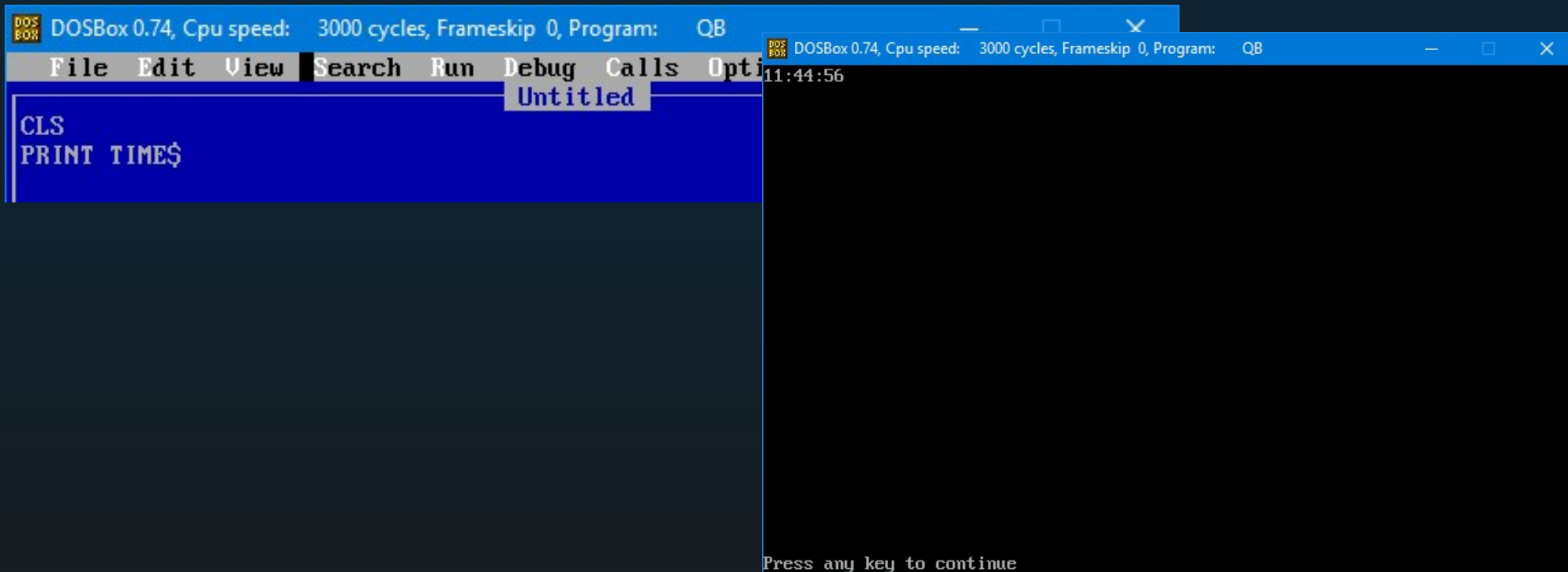
- ❑ Por vezes queres **voltar** a **determinada parte** do **código** e o Qbasic permite ir para um determinado local através do **GOTO**.
- ❑ É necessário **identificar** o **local** com um **número/texto** seguido de **dois pontos**
- ❑ Depois basta fazer **GOTO numero/texto**



```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB
File Edit View Search Run Debug Calls Options Help
Untitled
DIM op AS STRING
CLS
menu:
PRINT "*****"
PRINT "*          1 - Jogar          *"
PRINT "*          Q - Sair           *"
PRINT "*****"
INPUT "Opção: ", op
IF op = "1" THEN
PRINT "Jogar"
ELSEIF op = "q" THEN
SYSTEM
ELSE
GOTO menu
END IF
```

# TIME\$

- TIME\$ - devolve a hora no formato HH:MM:SS
- Exemplo:



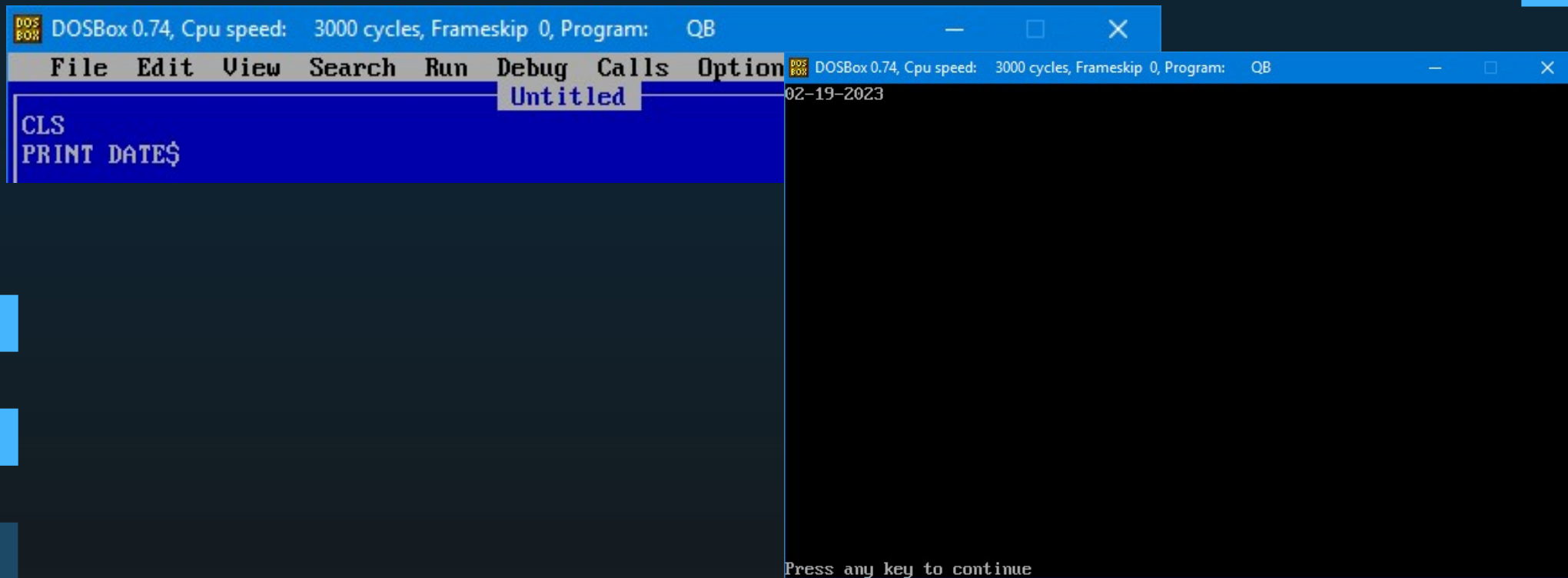
The screenshot shows two DOSBox 0.74 windows. The left window is titled 'Untitled' and contains the following code:

```
CLS  
PRINT TIME$
```

The right window is titled 'QB' and shows the output of the program: '11:44:56'. At the bottom of the right window, it says 'Press any key to continue'.

# DATE\$

- DATE\$ - devolve a data no formato DD:MM:AAAA
- Exemplo:



The screenshot shows two overlapping DOSBox windows. The foreground window is titled 'DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB' and displays a menu with options: File, Edit, View, Search, Run, Debug, Calls, and Options. Below the menu, the text 'Untitled' is visible. The command prompt shows 'CLS' followed by 'PRINT DATE\$'. The background window, also titled 'DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB', displays the output '02-19-2023' and 'Press any key to continue' at the bottom.

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB
File Edit View Search Run Debug Calls Options
Untitled
CLS
PRINT DATE$

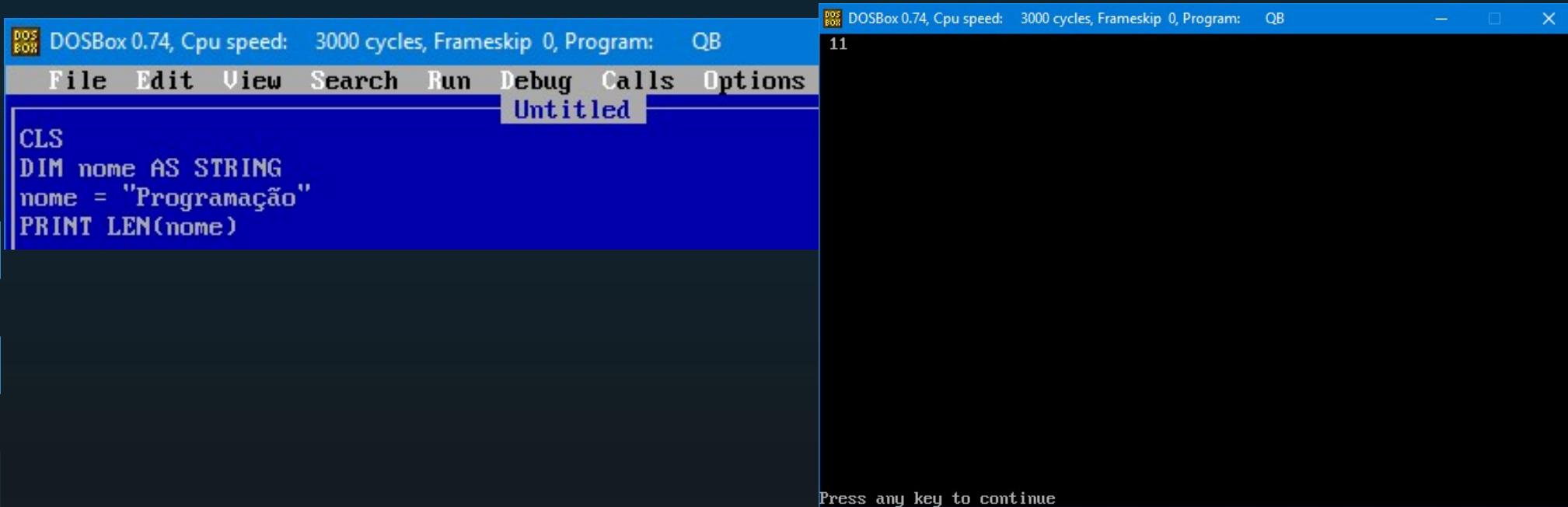
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB
02-19-2023
Press any key to continue
```





# LEN

- **Length** - tamanho
- Muito **útil** para saber o **tamanho** de uma **string** (nº de caracteres)
- Para todos os efeitos **retorna** o **nº** de bytes, mas neste caso byte=**caracteres**



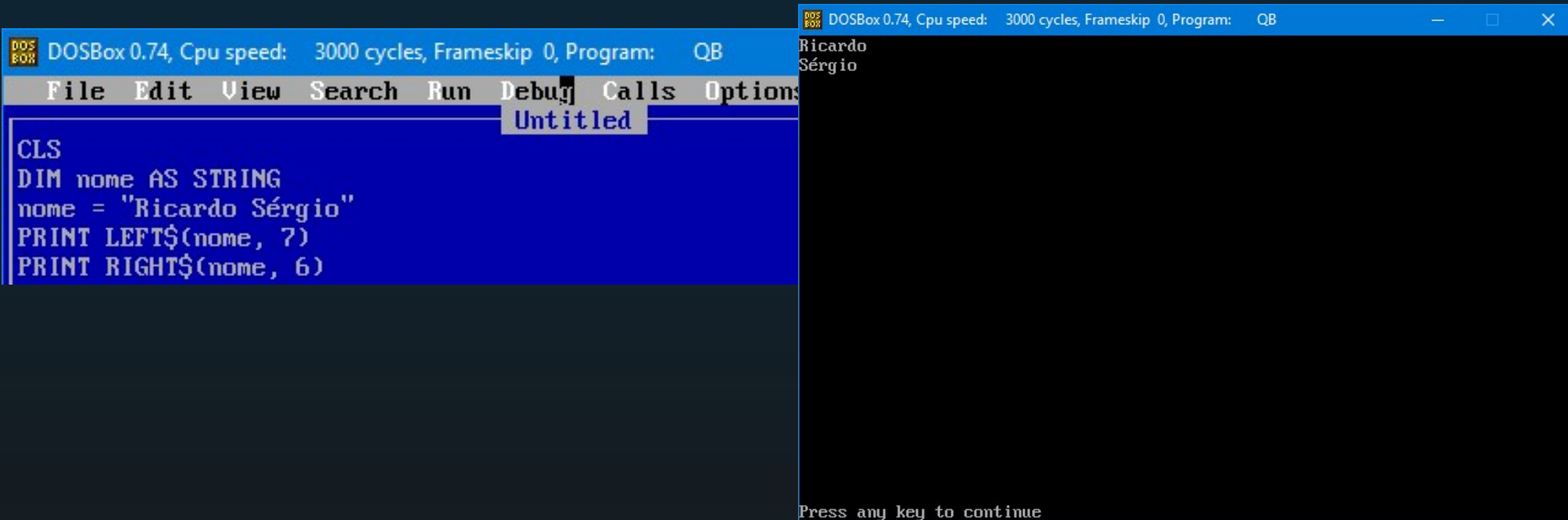
```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB
File Edit View Search Run Debug Calls Options
Untitled
CLS
DIM nome AS STRING
nome = "Programação"
PRINT LEN(nome)
11
Press any key to continue
```

# LEFT\$ e RIGHT\$

- LEFT\$ e RIGHT\$ servem para partir uma string pelo n° de caracteres

LEFT\$ (STRING, n° de caracteres desde a esquerda até cortar)

RIGHT\$ (STRING, n° de caracteres desde a direita até cortar)



The image shows two overlapping screenshots of a DOSBox window. The left screenshot shows the source code of a BASIC program in a text editor. The right screenshot shows the output of the program, which prints the first 7 characters and the last 6 characters of the string "Ricardo Sérgio".

```
CLS
DIM nome AS STRING
nome = "Ricardo Sérgio"
PRINT LEFT$(nome, 7)
PRINT RIGHT$(nome, 6)
```

Output:

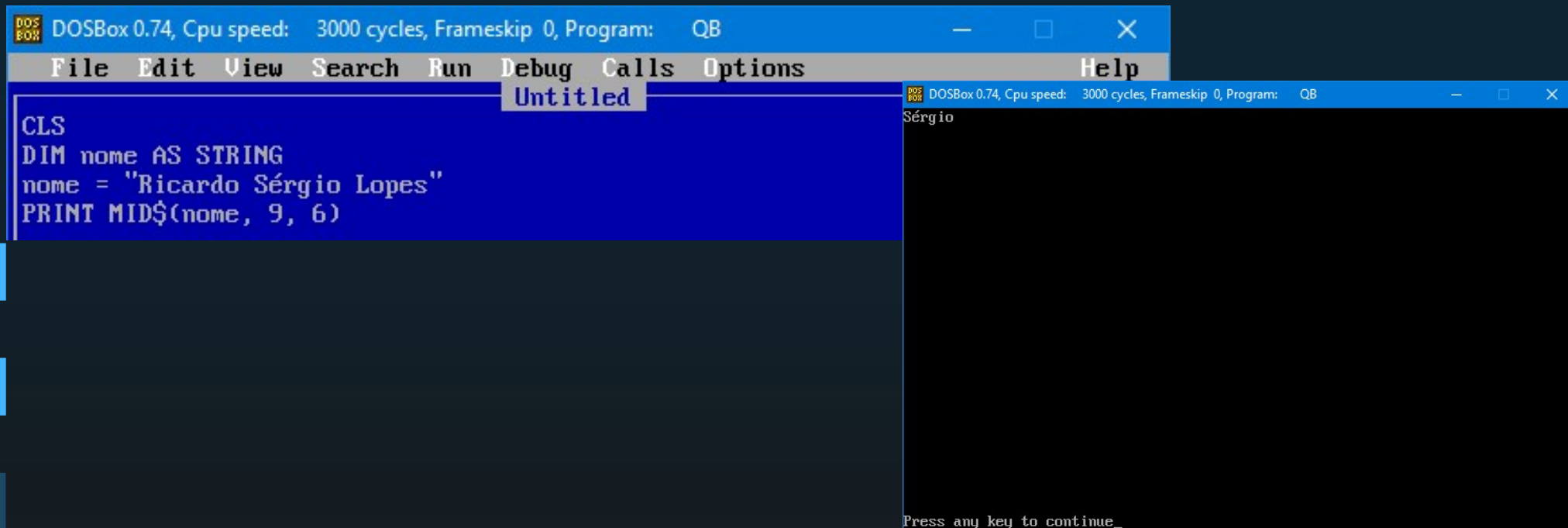
```
Ricardo
Sérgio
```

Press any key to continue

# MID\$

- MID\$ obtém parte de uma string

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



```
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 = "Ricardo Sérgio Lopes"
PRINT MID$(nome, 9, 6)
Sérgio
Press any key to continue_
```

# INT

- Esta função **obtem** a **parte inteira** de um **número**

```
DOS  
BOX  
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 s
```

```
DOS  
BOX  
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB  
4.643  
4  
Press any key to continue
```

# SQR e ABS

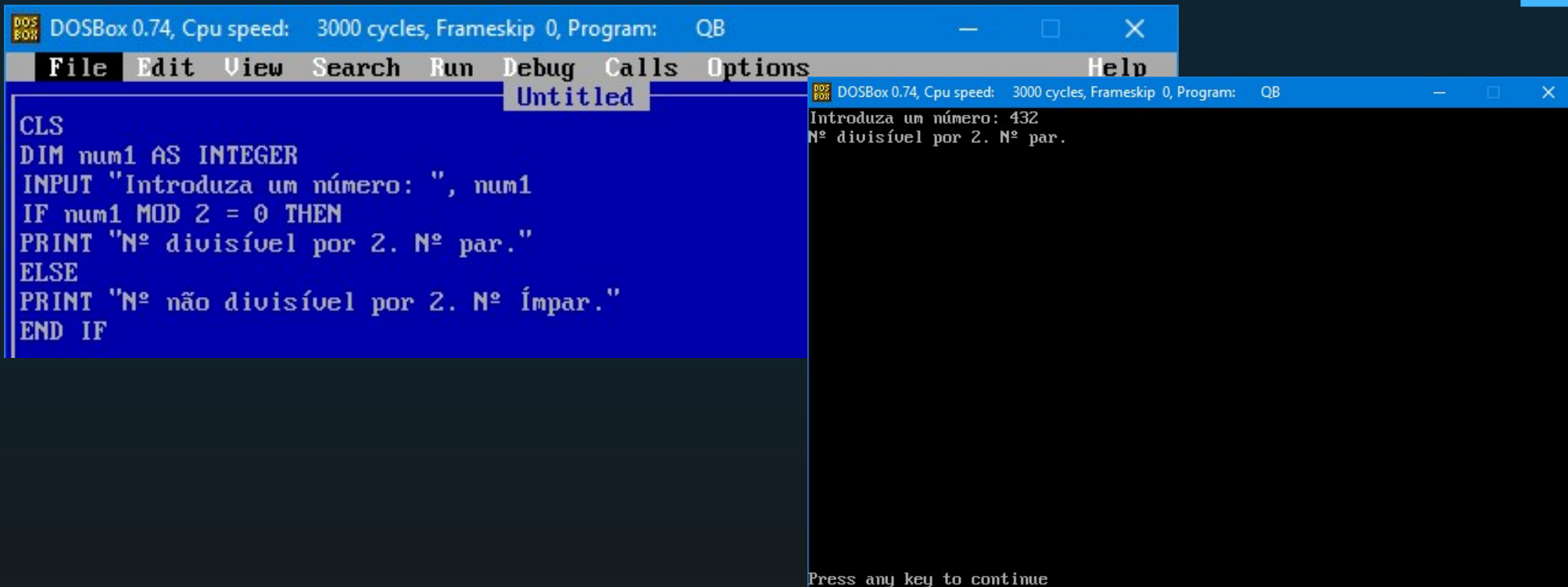
- **SQR** obtém a **raíz quadrada** de um  $n^\circ$
- **ABS** obtém a parte **absoluta** (módulo) do número (converte para positivo)

```
DOS  
FOR  
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB  
File Edit View Search Run Debug Calls Options Help  
Untitled  
CLS  
DIM num1 AS SINGLE  
DIM num2 AS SINGLE  
num1 = 16  
num2 = -3  
PRINT SQR(num1)  
PRINT ABS(num2)
```

```
DOS  
FOR  
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB  
4  
3  
Press any key to continue
```

# MOD

- ❑ MOD - Resto da divisão inteira
- ❑ Permite saber se um número é divisível por outro



```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB
File Edit View Search Run Debug Calls Options Help
Untitled
CLS
DIM num1 AS INTEGER
INPUT "Introduza um número: ", num1
IF num1 MOD 2 = 0 THEN
PRINT "Nº divisível por 2. Nº par."
ELSE
PRINT "Nº não divisível por 2. Nº Ímpar."
END IF
Introduza um número: 432
Nº divisível por 2. Nº par.
Press any key to continue
```

# RND

- ❑ RND faz um sorteio de números
- ❑ Funciona ligeiramente diferente das outras linguagens.
- ❑ Sintaxe:

**RND** \* N<sup>o</sup> max a sortear (exclusive)

Sorteia sempre um número de 0 até ao n<sup>o</sup> max exclusive

Exemplo:

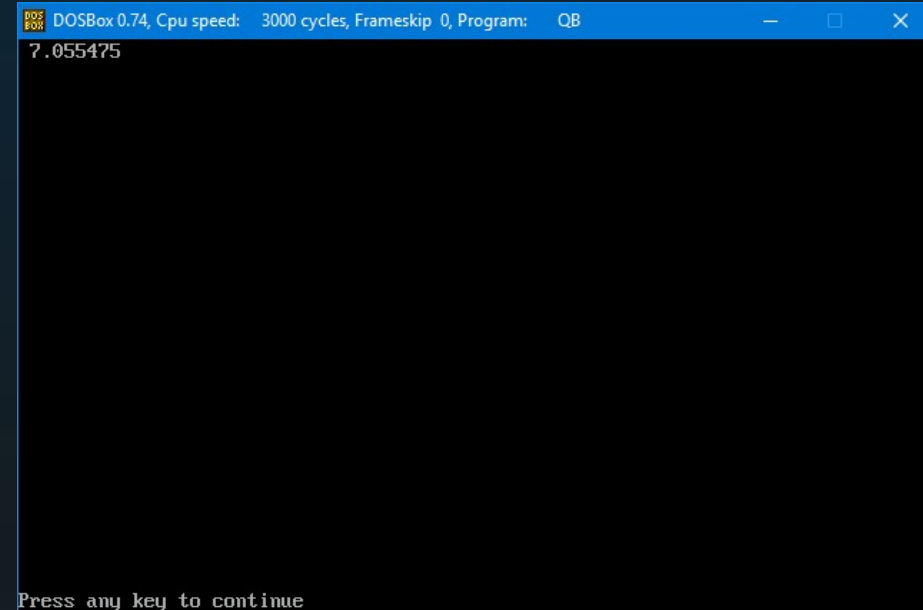
```
DOS BOX DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB
File Edit View Search Run Debug Calls Options
RND.BAS
CLS
PRINT RND * 10
_SLEEP
```

```
DOS BOX DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB
7.055475
Press any key to continue
```

# RND

- ❑ O que acontece se correremos outra vez o programa?
- ❑ Dá exatamente o mesmo resultado!!!
- ❑ Pois aqui está a grande diferença relativamente à maioria das outras linguagens.
- ❑ Deveria sair um novo  $n^o$ , senão em que parte isto é aleatório???

E se aplicarmos um ciclo o que acontecerá?

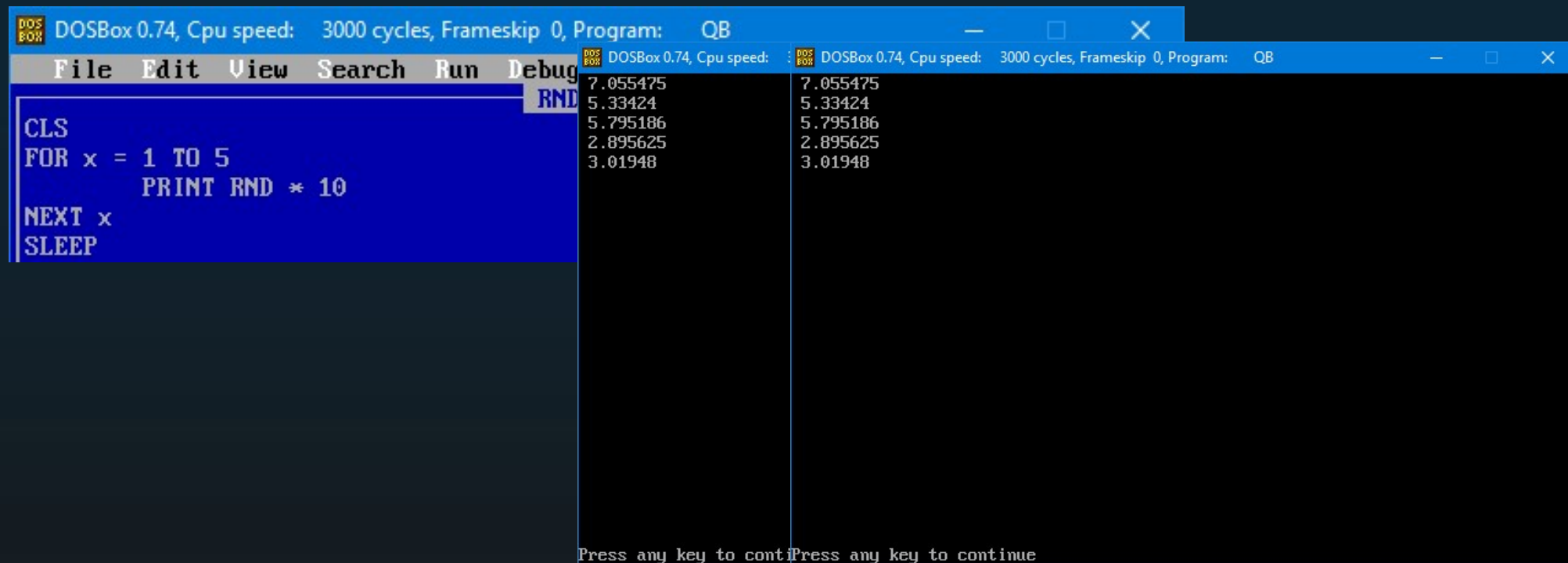


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB  
7.055475  
Press any key to continue



# RND

- Dá valores diferentes por cada ciclo FOR mas...
- Dá exatamente o mesmo resultado se voltarmos a correr o programa!!!



The screenshot shows a DOSBox window with a BASIC program and its output. The program is:

```
CLS
FOR x = 1 TO 5
  PRINT RND * 10
NEXT x
SLEEP
```

The output shows five random numbers, which are identical in two consecutive runs of the program:

```
7.055475
5.33424
5.795186
2.895625
3.01948
```

Press any key to continue

# RND TIMER

- ❑ Como resolver?
- ❑ Se basearmos o nosso RND em algo variável como o nosso relógio interno do computador já conseguimos ultrapassar o problema. Vejamos:
- ❑ Chama-se Randomize timer e assim já funciona.

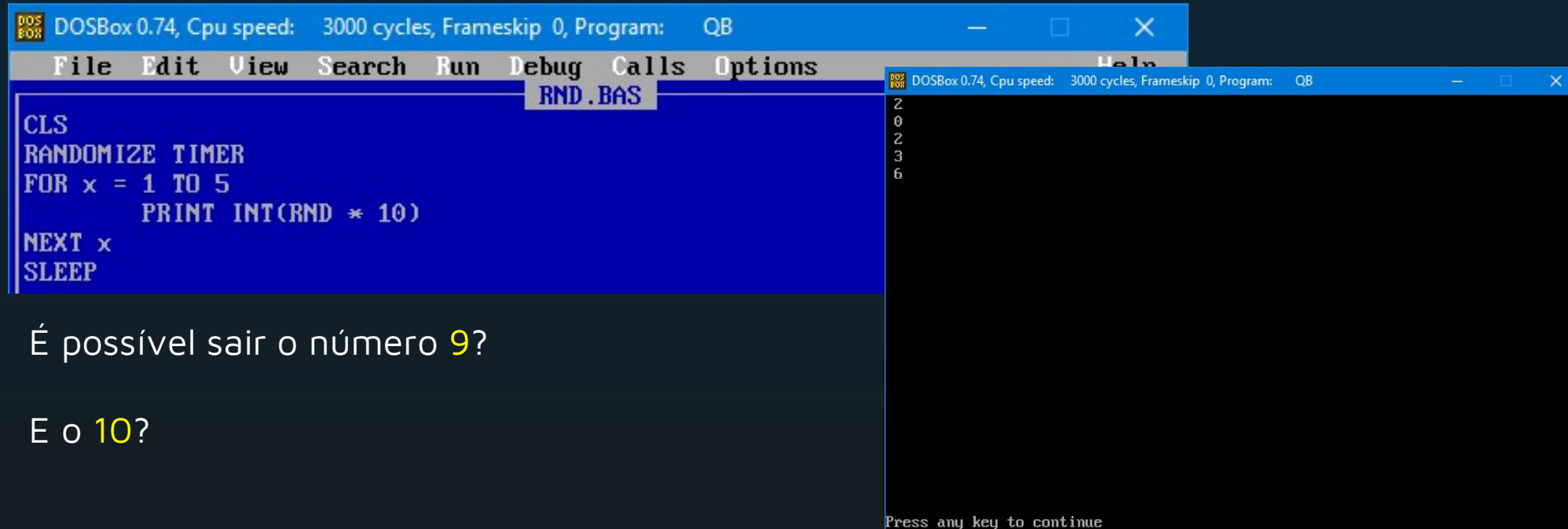
```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB
File Edit View Search Run Debug
CLS
RANDOMIZE TIMER
FOR x = 1 TO 5
  PRINT RND * 10
NEXT x
SLEEP
```

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB
9.877588
4.211651
7.131703
4.979823
8.485636
Press any key to continue
```

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB
6.987421
.8509028
3.307698
8.170588
3.75709
Press any key to continue
```

# RND

- ❑ Se necessitarmos de obter apenas número inteiros podemos recorrer a outra função que já aprendemos a INT.
- ❑ Exemplo:



The screenshot shows two windows from DOSBox 0.74. The left window displays the source code for a BASIC program named RND.BAS. The code is as follows:

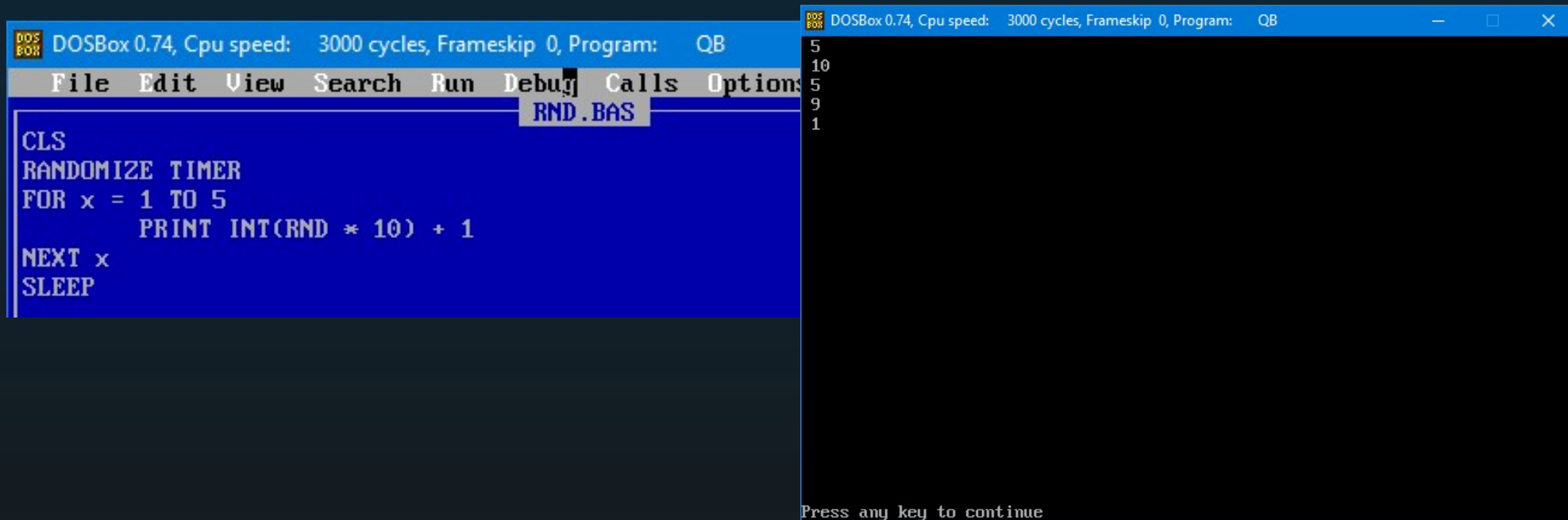
```
CLS
RANDOMIZE TIMER
FOR x = 1 TO 5
  PRINT INT(RND * 10)
NEXT x
SLEEP
```

The right window shows the output of the program, which consists of five lines of random integers: 2, 0, 2, 3, and 6. At the bottom of the right window, the text "Press any key to continue" is visible.

- ❑ É possível sair o número 9?
- ❑ E o 10?

# RND

- ❑ Mas tem de **começar sempre** em 0????
- ❑ Claro que **não**. Se quisermos a começar em 1 dizemos que ao **resultado adicionamos 1**. Se sair 0 ele na verdade mostra 1.



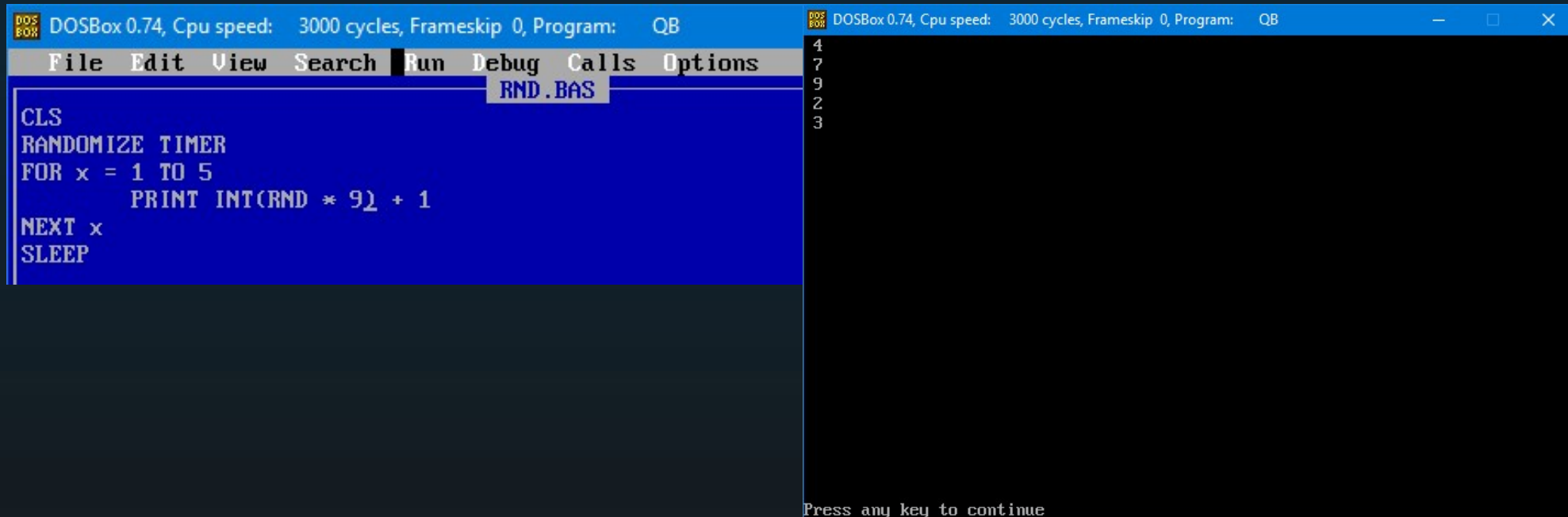
The image shows two screenshots of a DOSBox window. The left screenshot displays the source code of a BASIC program named RND.BAS. The code is as follows:

```
CLS
RANDOMIZE TIMER
FOR x = 1 TO 5
    PRINT INT(RND * 10) + 1
NEXT x
SLEEP
```

The right screenshot shows the output of the program, which consists of five random numbers: 5, 10, 5, 9, and 1. The DOSBox window title bar indicates the program is QB and the CPU speed is 3000 cycles. The prompt "Press any key to continue" is visible at the bottom of the window.

# RND

- ❑ Mas e o limite??? Se **adiciona 1** ficamos com **+1** que o **limite**. Se o limite era **10 exclusive** podia sair **9** e com **+1** veríamos o **10** no ecrã!!!???
- ❑ Resolução: ao adicionar **X** ao **valor inicial** temos de **subtrair** ao **final** para **resultar**.



The image shows two DOSBox windows. The left window displays a BASIC program named RND.BAS. The right window shows the output of the program, which consists of five random numbers: 4, 7, 9, 2, and 3. The program code is as follows:

```
CLS
RANDOMIZE TIMER
FOR x = 1 TO 5
    PRINT INT(RND * 9) + 1
NEXT x
SLEEP
```

The output of the program is:

```
4
7
9
2
3
```

At the bottom of the right window, it says "Press any key to continue".

# RND

- ❑ Desafio:
- ❑ Números entre 40 e 50(exclusive)

```
DOS
BOX DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB
File Edit View Search Run Debug Calls Options Help
RND.BAS
CLS
RANDOMIZE TIMER
FOR x = 1 TO 5
    PRINT INT(RND * 10) + 40
NEXT x
SLEEP
```

```
DOS
BOX DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QB
43
40
48
46
49
Press any key to continue
```

- ❑ What????
- ❑ Será sempre:
- ❑  $\text{INT}(\text{RND} * \text{diferença entre max e min}) + \text{valor min}$

# Para hoje

Realizar os guiões no Website.

**CREDITS:** This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik** and illustrations by **Stories**

