

# 3ª Aula - Ciclos. Leitura de Dados. Funções de Intervalo (I).

## Programação Mestrado em Engenharia Física Tecnológica

Samuel M. Eleutério  
sme@tecnico.ulisboa.pt

Departamento de Física  
Instituto Superior Técnico  
Universidade de Lisboa

## Converter polegadas em centímetros (II)

- Se quiséssemos agora **mostrar no ecran** uma tabela com as conversões de **1 a 10**, fazia pouco sentido repetir o programa anterior 10 vezes mas... ainda se fazia.

## Converter polegadas em centímetros (II)

- Se quiséssemos agora **mostrar no ecrã** uma tabela com as conversões de **1 a 10**, fazia pouco sentido repetir o programa anterior 10 vezes mas... ainda se fazia.

**E como seria se o quiséssemos fazer 100000 vezes? ...**

## Converter polegadas em centímetros (II)

- Se quiséssemos agora **mostrar no ecran** uma tabela com as conversões de **1 a 10**, fazia pouco sentido repetir o programa anterior 10 vezes mas... ainda se fazia.

**E como seria se o quiséssemos fazer 100000 vezes? ...**

- Esta tarefa corresponde a um **ciclo** constituído por

## Converter polegadas em centímetros (II)

- Se quiséssemos agora **mostrar no ecrã** uma tabela com as conversões de **1 a 10**, fazia pouco sentido repetir o programa anterior 10 vezes mas... ainda se fazia.

**E como seria se o quiséssemos fazer 100000 vezes? ...**

- Esta tarefa corresponde a um **ciclo** constituído por

**Início** ...

## Converter polegadas em centímetros (II)

- Se quiséssemos agora **mostrar no ecran** uma tabela com as conversões de **1 a 10**, fazia pouco sentido repetir o programa anterior 10 vezes mas... ainda se fazia.

**E como seria se o quiséssemos fazer 100000 vezes? ...**

- Esta tarefa corresponde a um **ciclo** constituído por

**Início**     ...     **Corpo**     ...

## Converter polegadas em centímetros (II)

- Se quiséssemos agora **mostrar no ecran** uma tabela com as conversões de **1 a 10**, fazia pouco sentido repetir o programa anterior 10 vezes mas... ainda se fazia.

**E como seria se o quiséssemos fazer 100000 vezes? ...**

- Esta tarefa corresponde a um **ciclo** constituído por

**Início**

...

**Corpo**

...

**Fim**

## Converter polegadas em centímetros (II)

- Se quiséssemos agora **mostrar no ecrã** uma tabela com as conversões de **1 a 10**, fazia pouco sentido repetir o programa anterior 10 vezes mas... ainda se fazia.

**E como seria se o quiséssemos fazer 100000 vezes? ...**

- Esta tarefa corresponde a um **ciclo** constituído por

**Início**    ...    **Corpo**    ...    **Fim**

- Primeiro temos de saber como fazer (**definir**) um **ciclo**.



## Converter polegadas em centímetros (II)

- Se quiséssemos agora **mostrar no ecrã** uma tabela com as conversões de **1 a 10**, fazia pouco sentido repetir o programa anterior 10 vezes mas... ainda se fazia.

**E como seria se o quiséssemos fazer 100000 vezes? ...**

- Esta tarefa corresponde a um **ciclo** constituído por

**Início**    ...    **Corpo**    ...    **Fim**

- Primeiro temos de saber como fazer (**definir**) um **ciclo**.
- Depois temos de **caracterizar** cada uma das suas partes.

## Converter polegadas em centímetros (II)

- Se quiséssemos agora **mostrar no ecrã** uma tabela com as conversões de **1 a 10**, fazia pouco sentido repetir o programa anterior 10 vezes mas... ainda se fazia.

**E como seria se o quiséssemos fazer 100000 vezes? ...**

- Esta tarefa corresponde a um **ciclo** constituído por

**Início**    ...    **Corpo**    ...    **Fim**

- Primeiro temos de saber como fazer (**definir**) um **ciclo**.
- Depois temos de **caracterizar** cada uma das suas partes.
- Iremos aqui usar o ciclo **while** (há outros):

## Converter polegadas em centímetros (II)

- Se quiséssemos agora **mostrar no ecrã** uma tabela com as conversões de **1 a 10**, fazia pouco sentido repetir o programa anterior 10 vezes mas... ainda se fazia.

**E como seria se o quiséssemos fazer 100000 vezes? ...**

- Esta tarefa corresponde a um **ciclo** constituído por

**Início**    ...    **Corpo**    ...    **Fim**

- Primeiro temos de saber como fazer (**definir**) um **ciclo**.
- Depois temos de **caracterizar** cada uma das suas partes.
- Iremos aqui usar o ciclo **while** (há outros):

```
while (condição) { ... corpo ... }
```

## Converter polegadas em centímetros (II)

- Se quiséssemos agora **mostrar no ecran** uma tabela com as conversões de **1 a 10**, fazia pouco sentido repetir o programa anterior 10 vezes mas... ainda se fazia.

**E como seria se o quiséssemos fazer 100000 vezes? ...**

- Esta tarefa corresponde a um **ciclo** constituído por

**Início**    ...    **Corpo**    ...    **Fim**

- Primeiro temos de saber como fazer (**definir**) um **ciclo**.
- Depois temos de **caracterizar** cada uma das suas partes.
- Iremos aqui usar o ciclo **while** (há outros):  
**while** (**condição**) { ... **corpo** ... }
- Enquanto a '**condição**' for satisfeita, o ciclo ir-se-á repetir. Assim, alguma '**coisa**' se deverá alterar quando da execução do **corpo**.

## Converter polegadas em centímetros (II)

- Se quiséssemos agora **mostrar no ecrã** uma tabela com as conversões de **1 a 10**, fazia pouco sentido repetir o programa anterior 10 vezes mas... ainda se fazia.

**E como seria se o quiséssemos fazer 100000 vezes? ...**

- Esta tarefa corresponde a um **ciclo** constituído por

**Início**    ...    **Corpo**    ...    **Fim**

- Primeiro temos de saber como fazer (**definir**) um **ciclo**.
- Depois temos de **caracterizar** cada uma das suas partes.
- Iremos aqui usar o ciclo **while** (há outros):

**while** (**condição**) { ... **corpo** ... }

- Enquanto a '**condição**' for satisfeita, o ciclo ir-se-á repetir. Assim, alguma '**coisa**' se deverá alterar quando da execução do **corpo**. No nosso caso, as **polegadas irão aumentando**.

## Converter polegadas em centímetros (II)

- Se quisessemos agora **mostrar no ecran** uma tabela com as conversões de **1 a 10**, fazia pouco sentido repetir o programa anterior 10 vezes mas... ainda se fazia.

**E como seria se o quisessemos fazer 100000 vezes? ...**

- Esta tarefa corresponde a um **ciclo** constituído por

**Início** ... **Corpo** ... **Fim**

- Primeiro temos de saber como fazer (**definir**) um **ciclo**.
- Depois temos de **caracterizar** cada uma das suas partes.
- Iremos aqui usar o ciclo **while** (há outros):

**while** (**condição**) { ... **corpo** ... }

- Enquanto a '**condição**' for satisfeita, o ciclo ir-se-á repetir. Assim, alguma '**coisa**' se deverá alterar quando da execução do **corpo**. No nosso caso, as **polegadas irão aumentando**.
- Quanto à condição de execução, ela **deixará de ser verdadeira** quando tivermos **mais de 10 polegadas**.

## Converter polegadas em centímetros (III)

- Vamos então alterar o nosso programa para mostrar no ecran as conversões de polegadas em centímetros de **1 a 10**, **avançando** de **1 em 1** (**Prog02\_02.c**). Partindo do programa já escrito:

## Converter polegadas em centímetros (III)

- Vamos então alterar o nosso programa para mostrar no ecran as conversões de polegadas em centímetros de **1 a 10**, **avançando** de **1 em 1** (**Prog02\_02.c**). Partindo do programa já escrito:

```
#include <stdio.h>
main ()
{
    float p, c ;
    p = 1.;

    c = 2.54 * p;
    printf ("%f\n", c);

}
```



## Converter polegadas em centímetros (III)

- Vamos então alterar o nosso programa para mostrar no ecran as conversões de polegadas em centímetros de **1 a 10**, **avançando** de **1 em 1** (**Prog02\_02.c**). Partindo do programa já escrito:

```
#include <stdio.h>  
main ()
```

```
{
```

```
float p, c ;
```

```
p = 1.;
```

```
c = 2.54 * p;
```

```
printf ("%f\n", c);
```

```
}
```

- Vamos aplicar o ciclo **while** para executar a repetição

## Converter polegadas em centímetros (III)

- Vamos então alterar o nosso programa para mostrar no ecran as conversões de polegadas em centímetros de **1 a 10**, **avançando** de **1 em 1** (**Prog02\_02.c**). Partindo do programa já escrito:

```
#include <stdio.h>
main ()
{
    float p, c ;
    p = 1.;
    while (Condição)

        c = 2.54 * p;
        printf ("%f\n", c);

}
```

- Vamos aplicar o ciclo **while** para executar a repetição

## Converter polegadas em centímetros (III)

- Vamos então alterar o nosso programa para mostrar no ecran as conversões de polegadas em centímetros de **1 a 10**, **avançando** de **1 em 1** (**Prog02\_02.c**). Partindo do programa já escrito:

```
#include <stdio.h>
main ()
{
    float p, c ;
    p = 1.;
    while (Condição)

        c = 2.54 * p;
        printf ("%f\n", c);

}
```

- Vamos aplicar o ciclo **while** para executar a repetição de um **bloco** de instruções;

## Converter polegadas em centímetros (III)

- Vamos então alterar o nosso programa para mostrar no ecrã as conversões de polegadas em centímetros de **1 a 10**, **avançando** de **1 em 1** (**Prog02\_02.c**). Partindo do programa já escrito:

```
#include <stdio.h>
main ()
{
    float p, c ;
    p = 1.;
    while (Condição)
    {
        c = 2.54 * p;
        printf ("%f\n", c);
    }
}
```

- Vamos aplicar o ciclo **while** para executar a repetição de um **bloco** de instruções;

## Converter polegadas em centímetros (III)

- Vamos então alterar o nosso programa para mostrar no ecran as conversões de polegadas em centímetros de **1 a 10**, **avançando** de **1 em 1** (**Prog02\_02.c**). Partindo do programa já escrito:

```
#include <stdio.h>
main ()
{
    float p, c ;
    p = 1.;
    while (Condição)
    {
        c = 2.54 * p;
        printf ("%f\n", c);
    }
}
```

- Vamos aplicar o ciclo **while** para executar a repetição de um **bloco** de instruções;
- Uma vez feito o cálculo e escrito, queremos calcular o **valor seguinte**, isto é, somar **"1."** ao valor actual;

## Converter polegadas em centímetros (III)

- Vamos então alterar o nosso programa para mostrar no ecran as conversões de polegadas em centímetros de **1 a 10**, **avançando** de **1 em 1** (**Prog02\_02.c**). Partindo do programa já escrito:

```
#include <stdio.h>
main ()
{
    float p, c ;
    p = 1.;
    while (Condição)
    {
        c = 2.54 * p;
        printf ("%f\n", c);
        p = p + 1.;
    }
}
```

- Vamos aplicar o ciclo **while** para executar a repetição de um **bloco** de instruções;
- Uma vez feito o cálculo e escrito, queremos calcular o **valor seguinte**, isto é, somar "**1.**" ao valor actual;

## Converter polegadas em centímetros (III)

- Vamos então alterar o nosso programa para mostrar no ecran as conversões de polegadas em centímetros de **1 a 10**, **avançando** de **1 em 1** (**Prog02\_02.c**). Partindo do programa já escrito:

```
#include <stdio.h>
main ()
{
    float p, c ;
    p = 1.;
    while (Condição)
    {
        c = 2.54 * p;
        printf ("%f\n", c);
        p = p + 1.;
    }
}
```

- Vamos aplicar o ciclo **while** para executar a repetição de um **bloco** de instruções;
- Uma vez feito o cálculo e escrito, queremos calcular o **valor seguinte**, isto é, somar "**1.**" ao valor actual;
- A condição que impomos para a **repetição** é "**p <= 10.0**".

## Converter polegadas em centímetros (III)

- Vamos então alterar o nosso programa para mostrar no ecran as conversões de polegadas em centímetros de **1 a 10**, **avançando** de **1 em 1** (**Prog02\_02.c**). Partindo do programa já escrito:

```
#include <stdio.h>
main ()
{
    float p, c ;
    p = 1.;
    while (Condição)
    {
        c = 2.54 * p;
        printf ("%f\n", c);
        p = p + 1.;
    }
}
```

- Vamos aplicar o ciclo **while** para executar a repetição de um **bloco** de instruções;
- Uma vez feito o cálculo e escrito, queremos calcular o **valor seguinte**, isto é, somar "**1.**" ao valor actual;
- A condição que impomos para a **repetição** é "**p <= 10.0**".  
**Nota:** quando se lida com reais é conveniente não usar a igualdade como teste de saída.



## Converter polegadas em centímetros (III)

- Vamos então alterar o nosso programa para mostrar no ecrã as conversões de polegadas em centímetros de **1 a 10**, **avançando de 1 em 1 (Prog02\_02.c)**. Partindo do programa já escrito:

```
#include <stdio.h>
main ()
{
    float p, c ;
    p = 1.;
    while (p < 10.1)
    {
        c = 2.54 * p;
        printf ("%f\n", c);
        p = p + 1.;
    }
}
```

- Vamos aplicar o ciclo **while** para executar a repetição de um **bloco** de instruções;
- Uma vez feito o cálculo e escrito, queremos calcular o **valor seguinte**, isto é, somar **"1."** ao valor actual;
- A condição que impomos para a **repetição** é **"p <= 10.0"**.  
**Nota:** quando se lida com reais é conveniente não usar a igualdade como teste de saída.

## Converter polegadas em centímetros (IV)

Com os programas seguintes, pode ver-se como se pode melhorar o aspecto de saída dos resultados:

- Com o programa **Prog02\_03.c** mostram-se os valor de **p** e de **c**;

## Converter polegadas em centímetros (IV)

Com os programas seguintes, pode ver-se como se pode melhorar o aspecto de saída dos resultados:

- Com o programa **Prog02\_03.c** mostram-se os valor de **p** e de **c**;
- Com o programa **Prog02\_04.c** mostram-se os valor de **p** e de **c** e **junta-se-lhe algum texto**;

## Converter polegadas em centímetros (IV)

Com os programas seguintes, pode ver-se como se pode melhorar o aspecto de saída dos resultados:

- Com o programa **Prog02\_03.c** mostram-se os valor de **p** e de **c**;
- Com o programa **Prog02\_04.c** mostram-se os valor de **p** e de **c** e **junta-se-lhe algum texto**;
- Com os programas **Prog02\_05.c** e **Prog02\_06.c** **alteram-se os textos de saída**;

## Converter polegadas em centímetros (IV)

Com os programas seguintes, pode ver-se como se pode melhorar o aspecto de saída dos resultados:

- Com o programa **Prog02\_03.c** mostram-se os valor de **p** e de **c**;
- Com o programa **Prog02\_04.c** mostram-se os valor de **p** e de **c** e **junta-se-lhe algum texto**;
- Com os programas **Prog02\_05.c** e **Prog02\_06.c** **alteram-se os textos de saída**;
- Com o programa **Prog02\_07.c** **coloca-se um título** e deixa-se **uma linha de intervalo** no início e no fim.

## Leitura de Dados a Partir da *Shell*

- Para escrever no ecran usámos a função **printf** e incluímos os caracteres de formatação ("**%d**" para **int**, "**%f**" para **float**, ...):

## Leitura de Dados a Partir da *Shell*

- Para escrever no ecran usámos a função **printf** e incluímos os caracteres de formatação ("**%d**" para **int**, "**%f**" para **float**, ...):  
**printf** ("Int: **%d**\n", **i1**);    **printf** ("Float: **%f**\n", **x1**);

## Leitura de Dados a Partir da *Shell*

- Para escrever no ecran usámos a função **printf** e incluímos os caracteres de formatação ("**%d**" para **int**, "**%f**" para **float**, ...):  
**printf** ("Int: **%d**\n", **i1**);    **printf** ("Float: **%f**\n", **x1**);
- A função equivalente ao **printf**, para **leitura**, é **scanf**.



## Leitura de Dados a Partir da *Shell*

- Para escrever no ecran usámos a função **printf** e incluímos os caracteres de formatação (" **%d**" para **int**, " **%f**" para **float**, ...):  
**printf** ("Int: **%d**\n", **i1**);    **printf** ("Float: **%f**\n", **x1**);
- A função equivalente ao **printf**, para **leitura**, é **scanf**.
- As instruções de **leitura** equivalentes são:

## Leitura de Dados a Partir da *Shell*

- Para escrever no ecrã usámos a função **printf** e incluímos os caracteres de formatação ("**%d**" para **int**, "**%f**" para **float**, ...):  
`printf ("Int: %d\n", i1);    printf ("Float: %f\n", x1);`
- A função equivalente ao **printf**, para **leitura**, é **scanf**.
- As instruções de **leitura** equivalentes são:  
`scanf ("%d", &i1);    scanf ("%f", &x1);`

## Leitura de Dados a Partir da *Shell*

- Para escrever no ecrã usámos a função **printf** e incluímos os caracteres de formatação ("**%d**" para **int**, "**%f**" para **float**, ...):  
**printf** ("Int: **%d**\n", **i1**);    **printf** ("Float: **%f**\n", **x1**);
- A função equivalente ao **printf**, para **leitura**, é **scanf**.
- As instruções de **leitura** equivalentes são:  
**scanf** ("**%d**", **&i1**);    **scanf** ("**%f**", **&x1**);  
Note-se que as variáveis estão **precedidas** do operador **&**.

## Leitura de Dados a Partir da *Shell*

- Para escrever no ecrã usamos a função **printf** e incluímos os caracteres de formatação ("**%d**" para **int**, "**%f**" para **float**, ...):  
`printf ("Int: %d\n", i1);`    `printf ("Float: %f\n", x1);`
- A função equivalente ao **printf**, para **leitura**, é **scanf**.
- As instruções de **leitura** equivalentes são:  
`scanf ("%d", &i1);`    `scanf ("%f", &x1);`  
Note-se que as variáveis estão **precedidas** do operador **&**.
- **&i1** (ou **&x1** ou, em geral, **&variável**) dá o **endereço** da posição de memória do **espaço reservado** para essa variável.

## Leitura de Dados a Partir da *Shell*

- Para escrever no ecrã usámos a função **printf** e incluímos os caracteres de formatação ("**%d**" para **int**, "**%f**" para **float**, ...):  
`printf ("Int: %d\n", i1);    printf ("Float: %f\n", x1);`
- A função equivalente ao **printf**, para **leitura**, é **scanf**.
- As instruções de **leitura** equivalentes são:  
`scanf ("%d", &i1);    scanf ("%f", &x1);`  
Note-se que as variáveis estão **precedidas** do operador **&**.
- **&i1** (ou **&x1** ou, em geral, **&variável**) dá o **endereço** da posição de memória do **espaço reservado** para essa variável. A função **scanf** irá ler do teclado o **valor** que digitarmos e **escrevê-lo**, com o formato indicado, nessa **posição de memória**.

## Leitura de Dados a Partir da *Shell*

- Para escrever no ecrã usámos a função **printf** e incluímos os caracteres de formatação ("**%d**" para **int**, "**%f**" para **float**, ...):  
`printf ("Int: %d\n", i1);`    `printf ("Float: %f\n", x1);`
- A função equivalente ao **printf**, para **leitura**, é **scanf**.
- As instruções de **leitura** equivalentes são:  
`scanf ("%d", &i1);`    `scanf ("%f", &x1);`  
Note-se que as variáveis estão **precedidas** do operador **&**.
- **&i1** (ou **&x1** ou, em geral, **&variável**) dá o **endereço** da posição de memória do **espaço reservado** para essa variável. A função **scanf** irá ler do teclado o **valor** que digitarmos e **escrevê-lo**, com o formato indicado, nessa **posição de memória**.

...	37376	37377	37378	37379	37380	37381	...
-----	-------	-------	-------	-------	-------	-------	-----

## Leitura de Dados a Partir da *Shell*

- Para escrever no ecrã usámos a função **printf** e incluímos os caracteres de formatação ("**%d**" para **int**, "**%f**" para **float**, ...):  
`printf ("Int: %d\n", i1);`    `printf ("Float: %f\n", x1);`
- A função equivalente ao **printf**, para **leitura**, é **scanf**.
- As instruções de **leitura** equivalentes são:  
`scanf ("%d", &i1);`    `scanf ("%f", &x1);`  
Note-se que as variáveis estão **precedidas** do operador **&**.
- **&i1** (ou **&x1** ou, em geral, **&variável**) dá o **endereço** da posição de memória do **espaço reservado** para essa variável. A função **scanf** irá ler do teclado o **valor** que digitarmos e **escrevê-lo**, com o formato indicado, nessa **posição de memória**.

...	37376	37377	37378	37379	37380	37381	...
-----	-------	-------	-------	-------	-------	-------	-----

↑ **&x1**

## Leitura de Dados a Partir da *Shell*

- Para escrever no ecrã usámos a função **printf** e incluímos os caracteres de formatação ("**%d**" para **int**, "**%f**" para **float**, ...):  
`printf ("Int: %d\n", i1);`    `printf ("Float: %f\n", x1);`
- A função equivalente ao **printf**, para **leitura**, é **scanf**.
- As instruções de **leitura** equivalentes são:  
`scanf ("%d", &i1);`    `scanf ("%f", &x1);`  
Note-se que as variáveis estão **precedidas** do operador **&**.
- **&i1** (ou **&x1** ou, em geral, **&variável**) dá o **endereço** da posição de memória do **espaço reservado** para essa variável. A função **scanf** irá ler do teclado o **valor** que digitarmos e **escrevê-lo**, com o formato indicado, nessa **posição de memória**.

...	37376	37377	37378	37379	37380	37381	...
	↑ &x1				↑ &i1		



## Leitura de Dados a Partir da *Shell*

- Para escrever no ecrã usámos a função **printf** e incluímos os caracteres de formatação ("**%d**" para **int**, "**%f**" para **float**, ...):  
`printf ("Int: %d\n", i1);`    `printf ("Float: %f\n", x1);`
- A função equivalente ao **printf**, para **leitura**, é **scanf**.
- As instruções de **leitura** equivalentes são:  
`scanf ("%d", &i1);`    `scanf ("%f", &x1);`  
Note-se que as variáveis estão **precedidas** do operador **&**.
- **&i1** (ou **&x1** ou, em geral, **&variável**) dá o **endereço** da posição de memória do **espaço reservado** para essa variável. A função **scanf** irá ler do teclado o **valor** que digitarmos e **escrevê-lo**, com o formato indicado, nessa **posição de memória**.

...	37376	37377	37378	37379	37380	37381	...
	↑ &x1				↑ &i1		

- **scanf** retorna o **número** de leituras efectuadas com sucesso.

## Leitura de Dados a Partir da *Shell*

Vamos agora **alterar** o nosso último programa (**Prog02\_07.c**) para permitir a **leitura dos valores**, a partir da **shell**, para a sua execução:

## Leitura de Dados a Partir da *Shell*

Vamos agora **alterar** o nosso último programa (**Prog02\_07.c**) para permitir a **leitura dos valores**, a partir da **shell**, para a sua execução:

- Começemos por **ler** o **valor inicial** das polegadas (**08**):

## Leitura de Dados a Partir da *Shell*

Vamos agora **alterar** o nosso último programa (**Prog02\_07.c**) para permitir a **leitura dos valores**, a partir da **shell**, para a sua execução:

- Começemos por **ler** o **valor inicial** das polegadas (**08**):

```
printf(" Qual o primeiro valor para as polegadas que deseja?");  
scanf ("%f", &p);
```

## Leitura de Dados a Partir da *Shell*

Vamos agora **alterar** o nosso último programa (**Prog02\_07.c**) para permitir a **leitura dos valores**, a partir da **shell**, para a sua execução:

- Começemos por **ler** o **valor inicial** das polegadas (**08**):  

```
printf(" Qual o primeiro valor para as polegadas que deseja?");  
scanf ("%f", &p);
```
- Podemos também querer **variar** o **incremento da variável** (**09**):

## Leitura de Dados a Partir da *Shell*

Vamos agora **alterar** o nosso último programa (**Prog02\_07.c**) para permitir a **leitura dos valores**, a partir da **shell**, para a sua execução:

- Começemos por **ler** o **valor inicial** das polegadas (**08**):  

```
printf(" Qual o primeiro valor para as polegadas que deseja?");  
scanf ("%f", &p);
```
- Podemos também querer **variar** o **incremento da variável** (**09**):  

```
printf (" Qual o incremento que deseja? ");  
scanf ("%f", &dp);
```

## Leitura de Dados a Partir da *Shell*

Vamos agora **alterar** o nosso último programa (**Prog02\_07.c**) para permitir a **leitura dos valores**, a partir da **shell**, para a sua execução:

- Começemos por **ler** o **valor inicial** das polegadas (**08**):  

```
printf(" Qual o primeiro valor para as polegadas que deseja?");  
scanf ("%f", &p);
```
- Podemos também querer **variar** o **incremento da variável** (**09**):  

```
printf (" Qual o incremento que deseja? ");  
scanf ("%f", &dp);
```
- Podemos ainda querer **alterar** o **valor final** da tabela (**10**):

## Leitura de Dados a Partir da *Shell*

Vamos agora **alterar** o nosso último programa (**Prog02\_07.c**) para permitir a **leitura dos valores**, a partir da **shell**, para a sua execução:

- Começemos por **ler** o **valor inicial** das polegadas (**08**):  

```
printf("Qual o primeiro valor para as polegadas que deseja?");  
scanf ("%f", &p);
```
- Podemos também querer **variar** o **incremento da variável** (**09**):  

```
printf ("Qual o incremento que deseja? ");  
scanf ("%f", &dp);
```
- Podemos ainda querer **alterar** o **valor final** da tabela (**10**):  

```
printf ("Qual o ultimo valor para as polegadas que deseja?");  
scanf ("%f", &pf);
```



## Leitura de Dados a Partir da *Shell*

Vamos agora **alterar** o nosso último programa (**Prog02\_07.c**) para permitir a **leitura dos valores**, a partir da **shell**, para a sua execução:

- Começemos por **ler** o **valor inicial** das polegadas (**08**):  

```
printf("Qual o primeiro valor para as polegadas que deseja?");  
scanf ("%f", &p);
```
- Podemos também querer **variar** o **incremento da variável** (**09**):  

```
printf ("Qual o incremento que deseja? ");  
scanf ("%f", &dp);
```
- Podemos ainda querer **alterar** o **valor final** da tabela (**10**):  

```
printf ("Qual o ultimo valor para as polegadas que deseja?");  
scanf ("%f", &pf);
```
- Finalmente, podemos querer juntá-los numa única linha (**11**, **12**):

## Leitura de Dados a Partir da *Shell*

Vamos agora **alterar** o nosso último programa (**Prog02\_07.c**) para permitir a **leitura dos valores**, a partir da **shell**, para a sua execução:

- Começemos por **ler** o **valor inicial** das polegadas (**08**):

```
printf("Qual o primeiro valor para as polegadas que deseja?");  
scanf ("%f", &p);
```

- Podemos também querer **variar** o **incremento da variável** (**09**):

```
printf ("Qual o incremento que deseja? ");  
scanf ("%f", &dp);
```

- Podemos ainda querer **alterar** o **valor final** da tabela (**10**):

```
printf ("Qual o ultimo valor para as polegadas que deseja?");  
scanf ("%f", &pf);
```

- Finalmente, podemos querer juntá-los numa única linha (**11**, **12**):

```
printf ("Valor (inicial, fim, incremento): ");  
scanf ("%f %f %f", &p, &pf, &dp);
```