



DEI
DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
TÉCNICO LISBOA

Introdução à Programação em C

**Tipos e Operadores Elementares
Estruturas de Controlo**

IAED

Resumo do 1º problema

- 1º problema: programa de conversão de temperaturas
- O que vamos aprender:
 - Definição de variáveis e atribuições
 - Ciclos utilizando a estrutura de controlo **while**
 - Ciclos utilizando a estrutura de controlo **for**
 - Definição de constantes usando **#define**
 - Introdução ao *Input/Output*

Conversão de Temperaturas

- Problema: construir um programa que escreve uma tabela de conversão de temperaturas da escala Fahrenheit para escala Celsius

$$T_{Celsius} = \frac{5}{9}(T_{Fahrenheit} - 32)$$

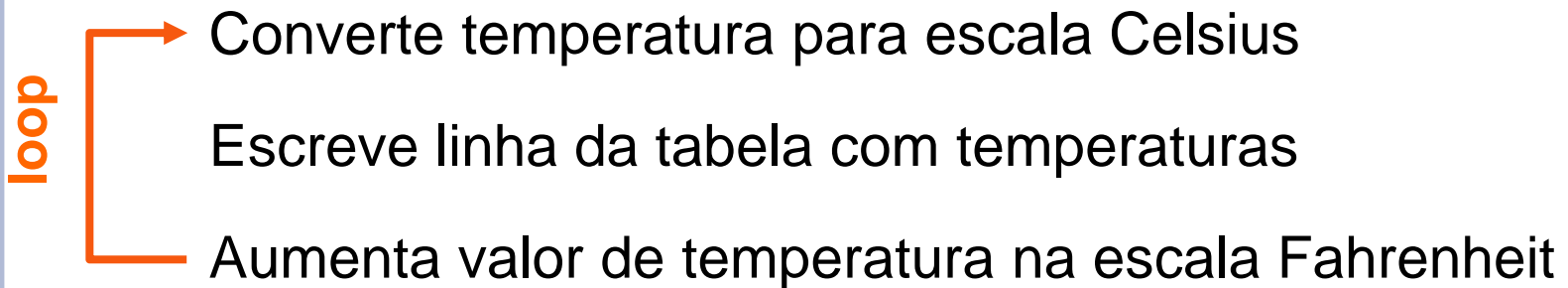
0	-17
20	-6
40	4
...	
260	126
280	137
300	148

Conversão de Temperaturas

Algoritmo

Inicia no limite inferior de temperatura na escala Fahrenheit

Enquanto temperatura não for superior ao limite superior



Conversão de Temperaturas

Algoritmo

Inicia no limite inferior de temperatura na escala Fahrenheit

fahr = inferior;

Enquanto temperatura não for superior ao limite superior

while (fahr <= superior)

{

→ Converte temperatura para escala Celsius

celsius = 5 * (fahr-32) / 9;

Escreve linha da tabela com temperaturas

printf("%d\t%d\n", fahr, celsius);

→ Aumenta valor de temperatura na escala Fahrenheit

fahr = fahr + passo;

}

Conversão de Temperaturas

```
#include <stdio.h>

/* Conversao Fahrenheit-Celsius */

int main ()
{
    int fahr, celsius;
    int inferior, superior, passo;

    inferior = 0;
    superior = 300;
    passo = 20;

    fahr = inferior;
    while (fahr <= superior)
    {
        celsius = 5 * (fahr-32) / 9;
        printf("%d\t%d\n", fahr, celsius);
        fahr = fahr + passo;
    }
    return 0;
}
```

**Vamos analisar o código
em detalhe**



`/* Comentários */`

```
#include <stdio.h>
```

```
/* Conversao Fahrenheit-Celsius */
```

```
int main ()
```

```
{
```

```
    int fahr, celsius;
```

```
    int inferior, superior, passo;
```

- Comentários:

Texto entre `/*` e `*/`

Declaração de variáveis

```
int main ()  
{  
    int fahr;  
    int celsius;  
    int inferior;  
    int superior;  
    int passo;  
  
    inferior = 0;  
    superior = 300;  
    passo = 20;
```


Declaração de variáveis

```
int main ()  
{  
    int fahr, celsius;  
    int inferior, superior, passo;  
  
    inferior = 0;  
    superior = 300;  
    passo = 20;
```

- Declaração de variáveis

Atribuição de valor a uma variável

```
int main ()  
{  
    int fahr, celsius;  
    int inferior, superior, passo;  
  
    inferior = 0;  
    superior = 300;  
    passo = 20;
```

- **Sintaxe:** <variável> = <expressão>

Atribuição de valor a uma variável

```
int main ()  
{  
    int fahr, celsius;  
    int inferior=0, superior=300, passo=20;
```

- Sintaxe: <variável> = <expressão>

Ciclo while

Algoritmo

Inicia no limite inferior de temperatura na escala Fahrenheit

fahr = inferior;

Enquanto temperatura for inferior ao limite superior

while (fahr <= superior)

{

→ Converte temperatura para escala Celsius

celsius = 5 * (fahr-32) / 9;

Escreve linha da tabela com temperaturas

printf("%d\t%d\n", fahr, celsius);


→ Aumenta valor de temperatura na escala Fahrenheit

fahr = fahr + passo;

}

Ciclo while

```
fahr = inferior;
while (fahr <= superior)
{
    celsius = 5 * (fahr-32) / 9;
    printf("%d\t%d\n", fahr, celsius);
    fahr = fahr + passo;
}
```

- Ciclo while
- **Síntaxe:** while (<expressao>) <instrução>
- Indentação ajuda a perceber estrutura do programa
- Erro frequente: while (i >= 0);
 i = i-1;
 

Output: printf

```
fahr = inferior;
while (fahr <= superior)
{
    celsius = 5 * (fahr-32) / 9;
    printf("%d\t%d\n", fahr, celsius);
    fahr = fahr + passo;
}
```

- printf permite efectuar escrita formatada
- %d inteiro decimal, \t carácter de tabulação
- Melhorar formatação:

```
printf("%3d\t%6d\n", fahr, celsius);
```

- **Problema: 0°F são -17,8°C e não -17°C**

Divisão inteira

```
fahr = inferior;
while (fahr <= superior)
{
    celsius = 5 * (fahr-32) / 9;
    printf("%d\t%d\n", fahr, celsius);
    fahr = fahr + passo;
}
```

- Divisão entre inteiros é divisão inteira
- Resultado é o quociente da divisão (número inteiro)
- Exemplo: $5/9$ é 0

Conseguimos resolver o problema da divisão inteira?

```
#include <stdio.h>

/* Conversao Fahrenheit-Celsius */

int main ()
{
    int fahr, celsius;
    int inferior, superior, passo;

    inferior = 0;
    superior = 300;
    passo = 20;

    fahr = inferior;
    while (fahr <= superior)
    {
        celsius = 5 * (fahr-32) / 9;
        printf("%3d\t%6d\n", fahr, celsius);
        fahr = fahr + passo;
    }
    return 0;
}
```


Conseguimos resolver o problema da divisão inteira?

```
#include <stdio.h>

/* Conversao Fahrenheit-Celsius */

int main ()
{
    float fahr, celsius;
    int inferior, superior, passo;

    inferior = 0;
    superior = 300;
    passo = 20;

    fahr = inferior;
    while (fahr <= superior)
    {
        celsius = (5.0/9.0) * (fahr-32);
        printf("%3.0f\t%6.1f\n", fahr, celsius);
        fahr = fahr + passo;
    }
    return 0;
}
```

...e será que precisamos de todas estas **variáveis** que, na realidade, são **constantes**?

```
#include <stdio.h>

/* Conversao Fahrenheit-Celsius */

int main ()
{
    float fahr, celsius;
    int inferior, superior, passo;

    inferior = 0;
    superior = 300;
    passo = 20;

    fahr = inferior;
    while (fahr <= superior)
    {
        celsius = (5.0/9.0) * (fahr-32);
        printf("%3.0f\t%6.1f\n", fahr, celsius);
        fahr = fahr + passo;
    }
    return 0;
}
```

...e será que precisamos de todas estas **variáveis** que, na realidade, são **constantes**?

```
#include <stdio.h>

/* Conversao Fahrenheit-Celsius */

int main ()
{
    float fahr, celsius;
int inferior, superior, passo;

inferior = 0;
superior = 300;
passo = 20;

    fahr = 0;
    while (fahr <= 300)
    {
        celsius = (5.0/9.0) * (fahr-32);
        printf("%3.0f\t%6.1f\n", fahr, celsius);
        fahr = fahr + 20;
    }
    return 0;
}
```

Definição de constantes

```
#include <stdio.h>

/* Conversao Fahrenheit-Celsius */

int main ()
{
    float fahr, celsius;

    fahr = 0;
    while (fahr <= 300)
    {
        celsius = (5.0/9.0) * (fahr-32);
        printf("%3.0f\t%6.1f\n", fahr, celsius);
        fahr = fahr + 20;
    }
    return 0;
}
```

- Má prática utilizar valores explícitos
- Alternativa: utilização de constantes, com `#define`

Definição de constantes: #define

```
#include <stdio.h>

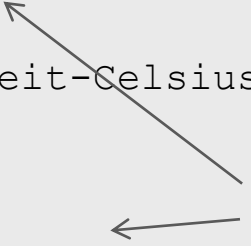
/* Conversao Fahrenheit-Celsius */

#define INFERIOR 0
#define SUPERIOR 300
#define PASSO 20

int main ()
{
    float fahr, celsius;

    fahr = INFERIOR;
    while (fahr <= SUPERIOR)
    {
        celsius = (5.0/9.0) * (fahr-32);
        printf("%3.0f\t%6.1f\n", fahr, celsius);
        fahr = fahr + PASSO;
    }
    return 0;
}
```

Directivas para o pre-compilador não têm “;”



Ciclos *while* vs *for*

```
fahr = INFERIOR;
while (fahr <= SUPERIOR)
{
    celsius = (5.0/9.0) * (fahr-32);
    printf("%3.0f\t%6.1f\n", fahr, celsius);
    fahr = fahr + PASSO;
}
```

```
for (fahr = INFERIOR; fahr <= SUPERIOR; fahr = fahr + PASSO)
{
    celsius = (5.0/9.0) * (fahr-32);
    printf("%3.0f\t%6.1f\n", fahr, celsius);
}
```

- Inicialização de variáveis: **fahr = INFERIOR**
- Teste: **fahr <= SUPERIOR**
- Incremento: **fahr = fahr + PASSO**

O que sabemos até agora:

- O programa começa sempre pela função **main()**
- Inclusão de funções adicionais ao C através da directiva **#include**
- Cada bloco é definido por { }
- Tipos de dados básicos: **char**, **int**, **float** e **double**
- Definição de variáveis e atribuições
- Ciclos utilizando a estrutura de controlo **while**
- Ciclos utilizando a estrutura de controlo **for**
- Definição de constantes usando directivas de pré-compilador **#define**
- Já sabemos escrever para o terminal (stdout) através do comando **printf**.
Mas como consigo interagir com o utilizador?

scanf: Leitura de valores do *standard input*

```
#include <stdio.h>

/* Objectivo: vamos pedir um inteiro ao utilizador e devolver o
   quadrado desse numero */

int main ()
{
    int x;
    printf("Introduza um valor inteiro:\n");
    scanf("%d",&x); ←
    printf("O quadrado de %d é %d\n", x, x*x);
    return 0;
}
```

O programa fica à espera
até que o utilizador
introduza algo no terminal

- %d para ler um inteiro
`scanf ("%d", &x) ;`
- Se pretendessemos ler dois inteiros seguidos:
`scanf ("%d%d", &x, &y) ;`
- Necessário colocar **&** antes do nome da variável



*Dennis Ritchie e Ken Thompson
a programarem um computador
Digital PDP-11, utilizando dois
terminais Teletype 33*

Alguma Dúvida ?