

## Programação Imperativa – Alguns elementos da sintaxe do C

Usaremos o seguinte programa como exemplo ilustrativo<sup>1</sup>.

```
1  //-- função que calcula o factorial
2  int factorial(int m){
3      int i, p=1;
4      for(i=0;i<m;i++)
5          p=p*i;
6      return(p);
7  }
8
9  main(){
10     int n;
11     printf("Valor de n? ");
12     scanf("%d",&n);
13     printf("Fact. de %d = %d\n",n,factorial(n));
14 }
```

---

- Um programa em linguagem C contém a definição de uma ou mais funções<sup>2</sup> tendo uma das funções que se chamar main. É por esta função que a execução do programa se inicia. No exemplo há 2 funções definidas: `fact` (linhas 2–7) e `main` (linhas 9–14).
- A definição de uma função tem a seguinte forma (podendo alguns dos elementos ser omitidos).

```
<tipo> <nome> ( <parametros> ) {
    <definicoes de variáveis>
    <instruções>
}
```

Para a função `fact` temos

```
<tipo>: int           <nome>: fact
<parametros>: int n   <definicoes de variáveis>: int i,p=0;
<inicialização>: p=0  <instrução>: for(i=0;i<m;i++) p=p*i;
<instrução>: return(p); <instrução>: p=p*i;
```

Quando não houver parâmetros (ver `main`), pode escrever-se “()” ou “(void)”.

---

<sup>1</sup>A numeração das linhas não faz parte do programa!

<sup>2</sup>E é essencialmente constituído por essas definições, pela definição de variáveis globais e por alguns outros elementos; a divisão em *funções* é fundamental para a estruturação do programa.

- Uma instrução pode ser simples (termina em ";" ) ou composta – zero ou mais instruções (simples ou compostas) dentro de chavetas. Uma instrução simples pode ser nula (consta apenas do ";" ).
- São exemplos de instruções simples

```

Var = Exp; ..... Atribuição
return Exp; ..... Retorno de função
if(Cond) Instr ..... Instrução condicional
if(Cond) Inst1 ..... Instrução condicional com alternativa
    else Inst2
while(Cond) Instr ..... "enquanto ... faça" (instrução de ciclo)
do Instr while(Cond);.. "faça ... enquanto" (instrução de ciclo)
for(Exp1; Exp2; Exp3)
    Instr ..... Instrução de ciclo "for"
switch(Expr){ ..... Instrução de selecção
    case Exp1: Insts1
    .....
    default: Instsd}

```

Note que as instruções "Instr" podem ser simples ou compostas.

A instrução de atribuição pode tomar outras formas, nomeadamente

- "Var <op>= Exp (onde <op> é um operador; é sinónimo de "Var = Var <op> Exp.
- "Var ++; é sinónimo de "Var = Var + 1.
- "Var --; é sinónimo de "Var = Var - 1.

---

### Exemplos

```

x=x*2;          return y%5;    for(i=0;i<=n;i++){
if(x<0){x=5;if(y==6 || y==7) x=y;}
if(x<0) printf("Erro!\n") else x='a';
while(x<n) x+=2;
switch(n){
    case 5: x=6; break;
    case 9: x=y; break;
    default: printf("Opcao errada\n");
}

```

---

**Exercício** Diga, justificando, se cada uma das seguintes instruções está sintaticamente correcta ou não.

1. x+5=x;
2. if(x>2?);
3. while(x>5) while(x<5) while(x<x) {x=2+x x=3;}
4. for(i=1;i<=10) printf("Outra vez!\n\n");
5. if(x==2); else{}