

**1.1** Se ainda não o fez, comece por ativar a sua conta nos computadores dos laboratórios do DCC (LabCC).

Os computadores dos laboratórios correm uma sistema operativo GNU/Linux com ambiente gráfico. Comece por procurar nos menus algumas aplicações que poderão ser úteis:

**Chromium/Firefox:** navegadores de páginas *web*;

**Adobe Acrobat Reader:** um visualizador de ficheiros PDF (*Portable Document Format*);

**Terminal:** um interpretador de comandos (“*shell*”);

**Atom:** outro editor de texto.

**1.2** Usado o editor de texto gedit, ou o Atom, passe o seguinte programa com uma mensagem de boas-vindas e grave-o num ficheiro `exemplo.c`

```
main() {  
    printf("Ser ou não ser, eis a questão.\n");  
}
```

(a) Compile o programa com o seguinte comando de “*shell*” de Linux:

```
gcc -Wall -o exemplo exemplo.c
```

(b) A compilação produz um ficheiro executável `exemplo` no diretório corrente (e possivelmente algumas mensagens). Execute o programa com o seguinte comando de “*shell*”:

```
./exemplo
```

(c) Baseado-se no exemplo apresentado na aula teórica, corrija o programa de forma a que a compilação deixe de produzir mensagens (“*warnings*”).

**1.3** Considere a versão final do programa de cálculo de volume de uma caixa apresentado nos slides.

(a) Use um editor de texto (e.g. gedit, ou Atom) para passar cuidadosamente o programa para um ficheiro `volume.c` (para se familiarizar com a sintaxe não use simplesmente “copy-paste”, componha mesmo o programa).

(b) Compile o programa com `gcc -Wall -o volume volume.c` e corrija eventuais erros. Deve obter um ficheiro executável `volume` no diretório corrente.

(c) Experimente correr o executável `volume` para calcular alguns volumes:

| L    | W    | H    | Volume |
|------|------|------|--------|
| 11   | 5    | 6    | 330    |
| 5    | 6    | 11   | ?      |
| 11   | 12   | 13   | ?      |
| 120  | 560  | 1700 | ?      |
| 1500 | 1500 | 1500 | ?      |

**1.4** Considere ainda o programa da questão anterior. Para  $L = W = H = 1500$ , podemos constatar que o programa não calcula o volume correto (porquê?).

Qual será o lado do maior cubo para o qual o programa dá resultados corretos? Justifique a sua resposta. *Sugestão:* nas arquiteturas Intel x86 (como os PCs no laboratório) o tipo `int` representa valores inteiros apenas entre  $-2^{31}$  e  $2^{31} - 1$ .

**1.5** Considere as declarações de variáveis.

```
int a, b, c;
float x, y;
```

Identifique e corrija os erros nas instruções de escrita e leitura seguintes.

- |   |                                      |
|---|--------------------------------------|
| (a) <code>printf("d%d%d\n", a, b, c);</code>    | (e) <code>printf("%.2\n", x);</code> |
| (b) <code>printf("%d;%d\n", x, y);</code>       | (f) <code>scanf("d", &amp;a)</code>  |
| (c) <code>printf("%d,%d\n", a, b, c);</code>    | (g) <code>scanf("%f",&amp;b);</code> |
| (d) <code>printf("%f,%f,%d\n", a, b, x);</code> | (h) <code>scanf("%f",y)</code>       |

**1.6** Modifique o programa de cálculo de volume de uma caixa para operar com dimensões de tipo `float`. Deve alterar não apenas as declarações de variáveis, mas também a leitura de dados e escrita de resultados.

**1.7** Simule a execução passo-a-passo das seguintes atribuições e indique os resultados finais das variáveis. Pode usar o simulador de C em <http://www.pythontutor.com/c.html#mode=display> para conferir os resultados.

- |   |  |   |
|---|--|---|
| (a) <code>int a, b, t;</code><br><code>a = 121;</code><br><code>b = 45;</code><br><code>t = a;</code><br><code>a = b;</code><br><code>b = t;</code> | <code>p = p * 2;</code><br><code>p = p * 3;</code><br><code>p = p * 4;</code>  | (d) <code>int a, b, r;</code><br><code>a = 54;</code><br><code>b = 24;</code><br><code>r = a%b;</code><br><code>a = b;</code><br><code>b = r;</code><br><code>r = a%b;</code><br><code>a = b;</code><br><code>b = r;</code> |
| (b) <code>int p;</code><br><code>p = 1;</code>  | (c) <code>int n;</code><br><code>n = 1;</code><br><code>n = n*10 + 2;</code><br><code>n = n*10 + 3;</code><br><code>n = n*10 + 4;</code> |   |

**1.8** Simule a execução dos seguintes programas indicando os valores das variáveis após cada passo.

|   |  |  |
|---|--|--|
| (a) <code>int s = 0;</code><br><code>s += 1;</code><br><code>s += 2;</code><br><code>s += 3;</code><br><code>s += 4;</code> | (c) <code>int n=0, s=0;</code><br><code>s += ++n;</code><br><code>s += ++n;</code><br><code>s += ++n;</code> | <code>y *= x;</code><br><code>y *= x;</code><br><code>y *= x;</code><br><code>y *= x;</code><br><code>y *= x;</code>               |
| (b) <code>int r = 1;</code><br><code>r *= 2;</code><br><code>r *= 3;</code><br><code>r *= 4;</code>                         | (d) <code>int n=0, s=0;</code><br><code>s += n++;</code><br><code>s += n++;</code><br><code>s += n++;</code> | (f) <code>int x = 3, y = 1;</code><br><code>y *= x;</code><br><code>y *= y;</code><br><code>y *= y;</code><br><code>y *= x;</code> |
|   | (e) <code>int x = 3, y = 1;</code>   |  |

**1.9** Escreva um programa que leia o valor do raio e calcule o volume de uma esfera usando a seguinte fórmula:  $V = 4/3\pi r^3$ .

*Sugestão:* na linguagem C não existe operação pré-definida para cálculo de potências; pode usar multiplicações repetidas, e.g. `r*r*r` para calcular  $r^3$ . Tenha ainda o cuidado de escrever a fração  $4/3$  como `4.0/3.0` (o que acontece se usar `4/3`?).

**1.10** Escreva um programa que lê um valor em euros e cêntimos (um `float`), acrescenta a taxa de IVA (23%) e imprime o resultado. Exemplo:

```
Valor sem IVA? 100
Valor com IVA: 123.00
```

**1.11** Escreva um program que lê uma quantia inteira de euros e mostra como pagar essa quantia em notas de €20, €10, €5 e moedas de €1. Exemplo:

```
Quantia em EUR? 93
notas EUR 20: 4
notas EUR 10: 1
notas EUR 5: 0
moedas EUR 1: 3
```

*Sugestão:* O quociente da divisão da quantia total por 20 dá-nos o número de notas de €20. Repita o processo para a quantia sobranete (o resto da divisão por 20) usando as notas mais pequenas (€10 e €5). Tenha o cuidado de usar variáveis inteiras e não virgula-flutuante.

**1.12** Escreva um programa para calcular a soma de duas frações. Deve ler o numerador e denominador de cada fração (valores inteiros) e imprime o numerador e denominador da soma. Exemplo:

```
Primeiro numerador? 3
Primeiro denominador? 4
Segundo numerador? 1
Segundo denominador? 2
3/4 + 1/2 = 10/8
```

Note que o resultado pode não ficar simplificado (como no exemplo acima).

*Sugestão:* Comece por simplificar no papel a soma de duas frações genéricas  $a/b + c/d$ .

**1.13** Escreva um programa `maior.c` que lê três valores inteiros da entrada-padrão e imprime o maior desses valores. Por exemplo: se introduzimos os números 45, 73 e 28, o programa deve imprimir 73.

Para submeter ao sistema de testes automáticos, o programa deve imprimir *apenas* o maior valor e uma mudança de linha `\n`; *não* deve imprimir mensagens extra (e.g. “Introduza um número”). Eis um “esqueleto” duma solução:

```
... // directivas include
int main(void) {
    int a, b, c, maior;
    scanf("%d %d %d", &a, &b, &c); // ler 3 valores
    ... // determinar o maior
    printf("%d\n", maior); // imprimir o resultado
}
```

**1.14** Escreva um programa que lê um valor  $x$  em vírgula flutuante, calcula a expressão  $3x^5 + 2x^4 - 5x^3 - x^2 + 7x - 6$  e imprime o resultado.

Como não existe em C uma operação pré-definida para potências deve usar multiplicações repetidas.

**1.15** Modifique o programa da questão anterior para calcular a expressão usando a seguinte fórmula equivalente:  $((((3x + 2)x - 5)x - 1)x + 7)x - 6$ . Note que desta forma não usamos potências e, portanto, o programa efectua menos multiplicações.<sup>1</sup>

---

<sup>1</sup>Esta técnica para calcular polinómios é designada *método de Horner*: [https://en.wikipedia.org/wiki/Horner%27s\\_method](https://en.wikipedia.org/wiki/Horner%27s_method).