



Programação de Computadores (2013-2014)

Sétima folha de Problemas

Matéria abordada: funções (passagem de parâmetros por valor e por referência; passagem de tabelas), alcance das variáveis, ciclos (while, do while, for).

Bibliografia: Capítulo 5, 6 e 8 de [Oua03].

Da folha 6 em diante estão proibidas as variáveis globais!

1. (a) Escreva o protótipo de uma função de nome `fatorial` que dado um inteiro n calcula:

$$n! = \prod_{i=1}^n i$$

Se o número dado for 0 a função devolve 1 ($0!=1$). Se o número dado for negativo a função devolve 0 (o que equivale a um valor incorreto). Implemente essa função.

- (b) Escreva um programa principal que lhe permita testar a função. O programa deve reportar uma mensagem de erro se a função devolver 0.

2. (a) Nos programas que tem escrito precisa com frequência de solicitar valores ao utilizador que devem satisfazer determinados requisitos, sendo um dos mais frequentes a necessidade de o número dado ser estritamente positivo.

Defina então uma função para pedir ao utilizador um número e retornar um número real positivo, de acordo com o seguinte protótipo:

```
float pede_real_pos(const char msg[], const char msg_erro[]);
```

Neste protótipo `msg` é a mensagem de pedido a apresentar ao utilizador antes de ele escrever o número e `msg_erro` é a mensagem de erro a apresentar sempre que o utilizador dá um número nulo ou negativo. Neste caso apresenta-se repetidamente `msg` até se ler um número positivo, sendo este o valor retornado pela função.

Verifique a correção sintática da função compilando o ficheiro (g++ -c f7e2.cpp).

Nota: Recordar-se que, sendo `tab` uma tabela de caracteres, ao fazermos

```
cout << tab;
```

o compilador assume que `tab` é uma string C e vai apresentar como texto os octetos que se encontram em memória desde `tab[0]` até encontrar um `'\0'` (ou terminar com erro se `tab` não estiver devidamente terminada com `'\0'`).

- (b) Acrescente ao ficheiro da função anterior (f7e2.cpp) a função de protótipo

```
float area_esfera(float raio);
```

que dado o raio de uma esfera, r , devolve a sua área ($A = 4\pi r^2$). Defina a constante PI com o valor 3.1416.

Verifique a correção sintática da função compilando o ficheiro (g++ -c f7e2.cpp).

- (c) Escreva no ficheiro (f7e2.cpp) o corpo da função `main`, que lhe permita testar as duas funções implementadas (`pede_real_pos` e `area_esfera`).

Para passar as mensagens à função `pede_real_pos`, defina duas variáveis na função `main`, como por exemplo:

```
char pede_raio[] = "De-me a medida do raio de uma esfera (>0): ";
```

```
char erro_raio[] = "A medida do raio nao pode ser nula ou negativa!\n";
```

e utilizá-las como parâmetros na chamada a `pede_real_pos`:

```
raio = pede_real_pos(pede_raio, erro_raio);
```

3. Copie do ficheiro `f7e2.cpp` para o ficheiro `f7e3.cpp` a função de protótipo

```
float pede_real_pos(const char msg[], const char msg_erro[]);
```

e a definição da constante `PI`.

- (a) Escreva, no novo ficheiro (`f7e3.cpp`), um programa principal em que usando a função anterior solicita um número real positivo (o raio duma esfera), calcula a sua área e apresenta esse valor ao utilizador.
- (b) Selecione de entre os protótipos seguintes, justificando, quais os que são adequados para uma função que, dado o raio r , de uma esfera, calcula o seu volume ($V = 4\pi r^3/3$) e a sua área ($A = 4\pi r^2$):

```
float area_volume_esfera(float raio, float volume, float area);  
void area_volume_esfera(float raio, float volume, float area);  
void area_volume_esfera(float raio, float &volume, float &area);  
float area_volume_esfera(float raio, float &area);
```

- (c) Implemente as duas funções cujos protótipos selecionou na alínea anterior (mantendo o nome proposto).

Em C++ não existe dificuldade em distinguir as duas funções que acabou de implementar. Porquê?

- (d) Escreva um programa principal que pede um número real estritamente positivo (a medida do raio), usando a função de protótipo

```
pede_real_pos(const char msg[], const char msg_erro[]);
```

e mostra a medida do volume e da área da esfera.

Teste no mesmo programa a utilização das duas funções.

4. O produto interno de dois vetores é dado pela soma dos produtos dos seus elementos.

- (a) Escreva o protótipo de uma função que dadas duas tabelas de `double`, `u` e `v` (que representam os vetores \vec{u} e \vec{v}) e a sua dimensão, n , calcule o seu produto interno:

$$\vec{u} \cdot \vec{v} = \sum_{i=0}^{n-1} v[i] \times u[i]$$

- (b) Implemente a função cujo protótipo escreveu.
- (c) Escreva um programa que lhe permita testar a função implementada.

Sugestão: Declare e inicialize duas tabelas para as coordenadas de dois vetores de dimensão 3. Mostre os vetores ao utilizador e o respetivo produto interno.

5. Um número N é divisível por 3 se a soma dos seus algarismos for também ela divisível por 3. Por exemplo, o número $N = 111111111$ é divisível por 3, porque a soma dos seus algarismos é igual a $1+1+1+1+1+1+1+1+1 = 9$. Se por sua vez a soma dos algarismos tiver mais do que um algarismo, o processo recomeça. Por exemplo, o número 1986 é divisível por 3, porque $1+9+8+6 = 24$ e $2+4 = 6$ e portanto, sabe-se que 6 é divisível por 3. Os algarismos divisíveis por 3 são: 3, 6 e 9. Esta propriedade é útil para testar a divisibilidade por 3 de grandes números formados por *centenas de algarismos*.

- (a) Implemente a função cujo protótipo é: `bool divisivel_por_3(int num);`

A função devolve `true` se `num` for divisível por 3 e `false` caso contrário. Use o resto da divisão por 10 para obter o algarismo mais à direita de um número e a divisão inteira por 10 para reter os restantes algarismos à esquerda.

- (b) Uma vez que a implementação anterior apenas permite testar a divisibilidade de números na gama de representação `int`, implemente uma outra função cujo protótipo seja:

```
bool divisivel_por_3(const char num[], int nalg);
```

onde `ndig` é o número de algarismos, ou:

```
bool divisivel_por_3(const string & num);
```

em que os algarismos do número se encontra armazenado na cadeia de caracteres `num`.

- (c) Escreva um programa que peça ao utilizador um número e/ou uma sequência de algarismos, de forma a testar cada uma das funções que implementou. Deve escrever no ecrã a conclusão acerca da divisibilidade por 3.

6. (a) Escreva uma função que dado um carácter, `c` e dois números, `a` e `b`, compreendidos entre 1 e 20 desenhe no monitor um retângulo de comprimento `a` e largura `b`, utilizando esse carácter. Exemplo, para `a=6`, `b=4` e `c='R'`:

```
RRRR
R    R
R    R
RRRR
```

A função não faz nada se algum dos números, `a` ou `b`, estiver fora do intervalo dado, devolvendo nesse caso o valor `true`; caso desenhe o quadrado devolve `false`.

- (b) Escreva um programa principal que lhe permita testar a função da alínea anterior: chame a função e desenhe o retângulo dado no exemplo e seguidamente peça ao utilizador os parâmetros de um retângulo a desenhar e apresente esse novo retângulo. No final da execução da função apresente uma mensagem de erro se o valor devolvido for `true`.

7. (a) Escreva uma função que dado um carácter, c e um número n compreendido entre 3 e 9 (inclusive) desenhe no monitor um Δ de altura n e largura $2n - 1$ utilizando esse carácter. Por exemplo, para $n = 4$ e $c = 'D'$, obtemos o seguinte resultado (os espaços após o último carácter não precisam de ser ecoados para o ecrã):

	Coluna j =						
	0	1	2	3	4	5	6
linha i = 0				D			
linha i = 1			D		D		
linha i = 2		D				D	
linha i = 3	D	D	D	D	D	D	D

A função não faz nada se o número n estiver fora do intervalo dado.

- (b) Escreva um programa principal que lhe permita testar a função da alínea anterior desenhando o Δ dado no exemplo. Peça ao utilizador os parâmetros necessários para desenhar um Δ e apresente-o no ecrã.

TPC Considere que deseja apresentar por ordem crescente 3 números inteiros no ecrã. Para isso:

- (a) Escreva uma função que dados 3 números inteiros os apresenta no ecrã por ordem crescente, separados por $<$. A função deve ainda devolver o número 0 se os números dados forem todos diferentes, o número 1 se houver exactamente 2 números iguais entre os três dados ou o número 2 se forem os três iguais.
- (b) Escreva um programa principal que lhe permita testar a função da alínea anterior, o qual deve apresentar uma frase com base no valor devolvido pela função.

Referências

[Oua03] S. Oualline. *Practical C++ Programming*. O'Reilly, 3rd edition, 2003.