Instituto Superior de Engenharia de Coimbra

Departamento de Engenharia Informática e Sistemas

Ano Letivo 2017/2018

Introdução à Programação

Ficha Laboratorial 5

Tópicos da matéria:

Utilização de funções em linguagem C.

Conversões implícitas e explícitas.

O tipo de dados char.

Variáveis locais e globais.

Nota: Antes da implementação deve desenvolver o algoritmo para cada um dos exercícios propostos.

1. Desenvolva uma função que devolva o quadrado de um número real passado como argumento.

Escreva um programa que, utilizando a função definida no ponto anterior, leia números e imprima os seus quadrados até que seja introduzido o valor zero.

2. Desenvolva uma função que verifique se um número inteiro está entre dois limites (também inteiros). Os três valores são passados como argumento. A função deve devolver 1, se o número estiver dentro dos limites, ou 0, caso contrário.

3. Construa uma função que escreva uma linha com n asteriscos. O valor de n é passado como argumento. A declaração da função poderá ter o seguinte formato: void linha (int n)

4. Escreva um programa que, utilizando a função da pergunta anterior, desenhe um quadrado no monitor. A dimensão do lado do quadrado deve ser especificada pelo utilizador

Exemplo: *quadrado com lado 3*

```
***
***
***
```

5. Escreva dois programas semelhantes ao da pergunta anterior, mas que desenhem um triângulo e um losango respectivamente (a altura do losango tem que ser ímpar).

Exemplo:

```
*
**
***
```

```
  *
 ***
*****
 ***
  *
```

6. Desenvolva uma função que desenhe um triângulo de asteriscos invertido e oco. O número de asteriscos da linha superior é passado como argumento da função. É garantido que o seu valor é ímpar e está compreendido entre 5 e 11. Na figura pode ver-se um exemplo para um triângulo em que a linha inicial tem 9 asteriscos. No final, a função deve devolver o número de asteriscos escritos.

7. Desenvolva uma função que desenhe no monitor um triângulo de números, invertido. O número de linhas é passado como argumento da função. É garantido que o seu valor é superior a 1 e inferior ou igual a 9. Na figura pode ver-se um exemplo para um triângulo com 5 linhas. No desenho do triângulo, em cada uma das linhas o valor do número aumenta até à coluna central e a partir daí diminui.

```
1 2 3 4 5 4 3 2 1
  1 2 3 4 3 2 1
    1 2 3 2 1
      1 2 1
        1
```

8. Desenvolva uma função que receba, como argumento, um valor inteiro positivo e devolva o numero de dígitos do valor recebido.

9. Desenvolva uma função que devolva o inteiro que mais se aproxima da média de dois reais positivos que são passados como parâmetro.

Nota: Partir do princípio que não existe nenhuma função de arredondamento em C.

10. Desenvolva as funções especificadas nas alíneas seguintes:

a) Função que devolva o cubo de um número inteiro e positivo passado como argumento;

b) Função que obtenha do utilizador um número inteiro compreendido entre 100 e 999. O valor obtido deve ser devolvido como resultado final;

c) Função que verifique se um determinado número inteiro N, recebido como argumento, obedece à seguinte propriedade: N é igual à soma do cubo dos seus algarismos. Um exemplo de um número que satisfaz esta propriedade é o $371 = 3^3 + 7^3 + 1^3$. É garantido que o valor recebido como argumento é um número com três dígitos. A função deve devolver 1 se a propriedade se verificar, ou 0, no caso contrário.

d) Construa um programa que obtenha do utilizador vários números inteiros pertencentes ao intervalo [100, 999] e verifique quais os que satisfazem a propriedade indicada na alínea c). A introdução de números termina quando o utilizador assim o desejar.

11. Desenvolva uma função que leia um conjunto de números inteiros e devolva o número de vezes que o valor máximo surgiu. A dimensão da sequência é passada como argumento.

12. Desenvolva uma função que devolva o mínimo múltiplo comum de dois números inteiros positivos passados como argumento.

13. Pretende-se desenvolver um programa que permita fazer um pequeno jogo entre duas pessoas. O programa escolhe um número entre 0 e 500, que os dois jogadores tentam acertar alternadamente. De cada vez que um dos jogadores indica um número, o programa deve informar se:

- O número está abaixo da solução
- O número está acima da solução
- O jogador acertou.

O programa deve identificar os jogadores como sendo jogador1 e jogador2.

Neste programa vamos utilizar duas funções (além da função void main(void) que controlará todo o jogo):

- Função que pede uma nova aposta a um dos jogadores. Recebe, como argumento, o número do jogador e devolve a sua nova aposta;
- Função que compara dois números passados como argumento. Se o primeiro argumento for menor do que o segundo devolve –1, se o segundo for menor do que o primeiro devolve 1, e se forem iguais devolve 0.

Após desenvolver estas funções, podemos criar o programa completo. No início, o programa deve escolher um número aleatório entre 0 e 500. Após isso, vai perguntando a cada um dos jogadores as suas apostas. O programa termina quando um dos jogadores acertar no número.

Exemplo de uma fase do jogo:

E a vez do jogador 1!

Qual a sua aposta? **4**

O valor 4 esta abaixo do numero certo

E a vez do jogador 2!

Qual a sua aposta? **18**

O valor 18 esta acima do numero certo

E a vez do jogador 1!

Qual a sua aposta? **6**

O valor 6 esta abaixo do numero certo

E a vez do jogador 2!

Qual a sua aposta? **9**

O valor 9 esta acima do numero certo

E a vez do jogador 1!

Qual a sua aposta? **7**

O jogador 1 acertou!

14. Desenvolva uma função que devolva o número de divisores de um valor inteiro e positivo, passado como argumento.

Após isso, escreva um programa que leia uma sequência de números inteiros. Sempre que, nessa sequência, surgir um número primo deve ser escrita no monitor a seguinte mensagem:

"O número que introduziu é primo!".

O programa termina quando for introduzido um número negativo ou nulo.

15. A conjectura de Goldbach, um matemático do século XVIII, afirma que qualquer número par (superior a 4) é igual à soma de dois números primos.

Exemplos:

80 = 7 + 73; 34 = 3 +31; 1256 = 7 + 1249

Desenvolva uma função que receba como argumento um número inteiro par, superior a 4, e escreva no monitor os dois números primos que verificam a referida afirmação. A função devolve 1, no caso de ter encontrado os dois números primos. Se não encontrar (o que é improvável, uma vez que ninguém conseguiu provar que a conjectura é falsa), devolve 0.

Nota: ao implementar o seu código pode utilizar a função desenvolvida na questão anterior.

16. Desenvolva uma função que receba um carácter passado como argumento, e, caso esse carácter seja uma vogal, devolva a sua ordem ('a', 'A' têm ordem 1, enquanto que 'u', 'U' têm ordem 5). Se o carácter não for uma vogal deve ser devolvido 0.

Nota: pode utilizar as funções existentes na biblioteca <ctype.h>.

17. Desenvolva uma função que leia uma palavra (terminada com '\n') e a reproduza no monitor, com todas as letras transformadas em maiúsculas. A função deve devolver o número de caracteres lidos.

Nota: não se pretende que utilize tabelas de nenhum tipo.

18. Desenvolva uma função que escreva um conjunto de caracteres da tabela ASCII e os respectivos códigos numéricos entre dois limites, passados como argumento.

Exemplo: se os limites forem 62 e 65 a tabela deverá ter o seguinte formato:

Código	**Carácter**
62	>
63	?
64	@
65	A

19. Realize as seguintes tarefas:

a) Elabore o algoritmo de uma função que permita calcular o factorial de um determinado número inteiro positivo.

b) Utilizando a função elaborada na alínea anterior, desenvolva um algoritmo que calcule a seguinte expressão:

$$\frac{n!}{p!(n-p)!}$$

Conversões entre tipos

20. Indique qual o resultado da execução dos seguintes problemas:

a)

```
#include <stdio.h>

void main(void)
{
   double d;
   float f;
   long int l;
   int i;

   i = l = f = d = 100/3;
   printf("%d\t%ld\t%f\t%f\n", i, l, f, d);
   d = f = l = i = 100/3;
   printf("%d\t%ld\t%f\t%f\n", i, l, f, d);
   i = l = f = d = 100/3.0;
   printf("%d\t%ld\t%f\t%f\n", i, l, f, d);
   d = f = l = i = 100/ 3.0;
   printf("%d\t%ld\t%f\t%f\n", i, l, f, d);
}
```

b)

```
#include <stdio.h>
void main(void)
{
   double y = 3.2, x;
   int i = 2, j;

   x = (j = y/i) * 2;
   printf("%f\t%d\n", x, j);

   j = (x = y/i) * 2;
   printf("%f\t%d\n", x, j);

   x = y * ( j = ((int)2.9 + 1.1)/y);
   printf("%f\t%d\n", x, j);
}
```

Variáveis automáticas, estáticas e globais

21. Para cada um dos seguintes programas, indique o resultado da sua execução:

a)

```
#include <stdio.h>

int n=4;

void f1(void)
{
      printf("%d\n",n);
      n++;
      printf("%d\n",n);
}
```

```
void f2(void)
{
      int n=-1;
      printf("%d\n",n);
      n=30;
      printf("%d\n",n);
}

void main(void)
{
      printf("%d\n",n);
      n=10;
      f1();
      printf("%d\n",n);
      f2();
      printf("%d\n",n);
}
```

b)

```
#include<stdio.h>

void f1(int n)
{
      static int x = 0;
      int y = 0;
      printf("%d\t%d\t%d\n",x, n, y);
      x ++;
      n ++;
      y ++;
}

void main(void)
{
      int i;
      for(i=0;i<2;i++)
      {
            printf("\nIteracao %d:\n",i);
            f1(i);
      }
}
```

c)

```
#include <stdio.h>

int i = 1;

int start(void)
{
   return i;
}

int next(int j)
{
   j = i++;
   return j;
}
```

```
int last(int j)
{
   static int i = 10;
   j = i--;
   return j;
}

void main(void)
{
   int i, j;

   i = start();

   for(j = 1; j <= 3; j++)
   {
      printf("%d\t%d\n", i, j);
      printf("%d\n", next(i));
      printf("%d\n", last(i));
   }
}
```