Ficha Prática nº2 Programação C

Programação III

Paul Crocker e Simão Melo de Sousa

23 de Março de 2001

1 Introduções

Exercício 1:
Sejam as seguintes declarações C:

```
int n=10; p = 4;
long q = 2;
float x = 1.75;
```

dar o tipo e o valor das seguintes expressões:

```
1)  n + q
2)  n + x
3)  n % p + q
4)  n < p
5)  n >= p
6)  n > q
7)  q + 3 * (n > p)
8)  q && n
9)  (q -2 ) && ( n - 10 )
10) x * (q == 2)
11) x * (q = 5)
```

Exercício 2:
Explique o comportamento do seguinte programa quando não é introduzido um número.

```
main()
{
 int n;
 do {
    printf("Dar um numero: ");
    scanf("%d", &n);
```



```
    printf("aqui vai o seu quadrado");
    }
 while(n);
}
```

Exercício 3: (*Output*)
Quais são os resultados fornecidos por este programa:

```
#include <stdio.h>

main()
{
    int n = 543;
    int p = 5;
    float x = 34.5678;
    printf("A: %d %f\n", n x);
    printf("B: %4d %10f\n", n ,x);
    printf("C: %2d %3f\n", n ,x);
    printf("D: %10.3f %10.3e\n", x , x);
    printf("E: %*d\n", p, n);
    printf("F: %*.*f\n", 12, 5, x);
}
```

Exercício 4: (*Input*)
Quais são os valores introduzidos nas variáveis `n` e `p`, de tipo `int`, pelo comando `scanf("%4d %2d", &n, &p);` quando são introduzidas as seguintes sequências de caracteres (ˆ para um espaço, # para o "newline"):

- 12ˆ45#
- 123456#
- 123456ˆ7#
- 1ˆ458#
- ˆˆˆ4567ˆˆ8912#

Exercício 5: (*Mensagens de erro*)
Copie o programa "lab011.c"presente em `~crocker/prog3` para a sua área e modifique este programa C da seguinte forma. Compila o programa com o erro e anota a mensagem de erro produzido ou a descrição da anomalia de funcionamento, explique. A seguir a cada alínea, corrija o programa, compile e verifique que o seu programa voltou a funcionar.

1. apague o /* do comentário.
2. apaga o */ do comentário.
3. tire a directiva ao pre-processador #include $< stdio.h >$.
4. tire os parêntesis triangulares a volta do stdio.h.
5. Acrescente um "ponto e vírgula"no fim da directiva ao pre-processador.
6. Acrescente um "ponto e virgula"depois da última chaveta
7. Apague o ponto e virgula a seguir à última chamada ao printf.
8. Tire a letra "f"do comando printf.
9. Utilize apóstrofos em vez das plicas na primeira chamada ao printf.
10. Troca as chamadas a printf.

Qual destas erros eram erros de sintaxe ?

Exercício 6:
2) Objectivo : I/O erros Copie o programa "lab012.c"presente em `~crocker/prog3` para a sua área.

1. Explique o que faz o programa.
2. Compile e execute o programa varias vezes
3. Execute o programa e introduza números inteiro maiores do que o maior inteiro aceite pelo sistema (p.ex 5000000000). Tente a seguir com números menores do que o menor número aceite pelo sistema (p.ex -500000000000).
4. Execute o programa e introduza um inteiro que contêm uma virgula (p.ex 123,45). Qual é o inteiro que a maquina escreve? Explique.
5. Execute o programa e introduza um número decimal (p.ex 123.88).
6. remove o & em frente a referência à variável "num"no scanf. Compile e execute o programa, qual é o resultado?
7. Remove a referência à variável "num"no scanf.
8. Restabeleça a chamada ao scanf. Insere um & em frente do "num"na linha do printf. Compile e Execute. Qual é o resultado?
9. Restabeleça a referência ao "printf". Remove o %d ne referência ao "printf". Compile e Execute. Qual é o resultado?

Exercício 7:
Copie o programa "lab013.c"presente em `~crocker/prog3` para a sua área.

1. ver o conteúdo do programa e explicar.
2. compile e corrija os erros.

Exercício 8: (*IRS*)
Escreva um programa que peça o salario mensal ilíquido de uma pessoa e que consoante os valores da tabela 1 indiqua o valor de IRS a pagar.

Rendimento Anual Bruto	**Taxa**
até 2000 *contos*	20%
2001 – 2500 *contos*	30%
2501 – 3000 *contos*	35%
> 3000 *contos*	40%

Figura 1: Pseudo-tabela do IRS

Exercício 9: (*Aço*)
Um certo aço é classificado de acordo com o resultado de três testes abaixo indicados, que devem determinar, se o mesmo aço satisfaz as seguintes especificações:

- conteúdo de carbona abaixo de 70%;
- dureza de Rokwell maior do que 50;
- resistência a tracção maior do que 8000 psi.

Ao aço é atribuído o grau

- 10 se passou todos os testes;
- 9 se passa somente os testes 1 e 2;
- 8 se passa só o teste 1;
- 7 nos outros casos.

Escreva um programa que peça os três valores ao utilizador, e que devolva a classificação do aço. Um cuidado particular terá de ser dado na eficiência dos testes envolvidos nas condicionais.

Exercício 10: (*O grande Gauss*)
Escreva um programa C que permita calcular a soma dos n primeiros inteiros.

- Utilizando um ciclo;
- Utilizando a formula do Gauss;

Exercício 11: (*Menu*)
Escreva um programa C que consoante a escolha do utilizador calcula:

1. A área de um triângulo;
2. A área dum esfera;
3. Se um número introduzido é negativo, positivo ou nulo;
4. Se um número introduzido é primo ou não. (usar o crivo de Eratostene);

Exercício 12: (*A praga do coelho*)
Escreva um programa que permita calcular para um dado x o valor de $Fib(x)$, sem usar recursividade, tendo em conta que :

$$\begin{cases} Fib(0) = 1 \\ Fib(1) = 1 \\ Fib(n+2) = Fib(n) + Fib(n+1) \end{cases}$$

Exercício 13:

1. Escreva um programa que iniciliza uma variável para um número de três dígitos (100-999) e depois imprime a sua expansão em base 10. exemplo 145 = 1 * 100 + 4* 10 + 5*1.
2. Escreva um programa que faça a expansão dum número de três dígitos em qualquer base b (2-16)

Exercício 14: (*Capicua*)
Escreva um programa C que permita verificar se um dado número inteiro n é capicua ou não.

Exercício 15:
Faça programas para

1. Copiar o standard input para o standard output
2. Contar o número de linhas e e caracteres de espaço do input
3. Contar o número de palavras (uma palavra é uma cadeia de caracteres seperada por whitespace : espaço, tabulação ou caracteres de "nova linha"(newline)).
4. Contar o número de o número de digitos 0,1, ... ,9,

Para Contar o número de palavras (P). O Algoritmo é:
whitespace é espaço em branco ' ', tab ´\t´ e carriage return '\n'
set numP=zeroset
set dentroP=false
ler : character c : com getchar(c)até o fim do ficheiro

```
if (c não é whitespace)then
  if (dentroP é false )
     then => entramos dentro duma cadeia de caracteres : numP++ e set dentroP=true
     else => já estavamos dentro duma cadeia/palavara : portanto fazer nada
else
  => c é whitespace : set dentroP=false
```

Para contar o número de digitis utiliza um Array[10], e manipule o índice do array assim: ler character com getchar(c). Depois if (c é digito)then Array[c-'0']++
(ver o "Kernighan and Ritchie"capítulo 1)

Exercício 16:
Escreva um programa que mostra a tabuada da multiplicação dos números de 1 a 10 da seguinte forma:

```
      |   1   2   3   4   5   6   7   8   9  10
-----------------------------------------------
   1  |   1   2   3   4   5   6   7   8   9  10
   2  |   2   4   6   8  10  12  14  16  18  20
   3  |   3   6   9  12  15  18  21  24  27  30
   4  |   4   8  12  16  20  24  28  32  36  40
   5  |   5  10  15  20  25  30  35  40  45  50
   6  |   6  12  18  24  30  36  42  48  54  60
   7  |   7  14  21  28  35  42  49  56  63  70
   8  |   8  16  24  32  40  48  56  64  72  80
   9  |   9  18  27  36  45  54  63  72  81  90
  10  |  10  20  30  40  50  60  70  80  90 100
```

---

2 Primeira abordagem aos subprogramas em C

Exercício 17:
Resolva os exercícios anteriores usando funções.

Exercício 18: (*Elevações*)
Escreva uma função C que, dado dois parâmetros x e y inteiros, permita calcular x^y. Uma generalização dessa função, na libraria matemática (math.h), tem por declaração `double pow(double x,y)`.

Exercício 19: (*Fahrenheit 451*)

1. Escreva uma função C que permita converter uma quantidade em graus Fahrenheit em graus Celcius, utilizando a fórmula $^oC = \frac{5}{9}(^oF - 32)$.

2. Escreva uma função que permita devolver a tabela de conversão de Fahrenheit para Celcius do graus 0 até o grau 500 com intervalo de 10.

Exercício 20: (*Horner*)
Podemos representar um polinómio P por um vector p de reais em que uma célula de índice i representa o coeficiente associado à potência de grau i. Escreva um programa que inicializa P e que dado um x, calcula $P(x)$ usando o método de Horner:

$$P_n(x) = (p_n x + p_{n-1})x + p_{n-2})x + \cdots + p_1)x + a_0$$

Sugestão: escreva a função `float Horner(float x,float p[N])`

Exercício 21: (*Integral pelo métodos dos rectângulos*)
O calculo de um integral $\int_a^b f(t)dt$ pode ser visto como um "simple"cálculo de área. Efectivamente, dado um intervalo delimitado por a e b, o integral de f neste intervalo é o valor da área compreendida entre a curva representativa da função e o eixo do x. Assim sendo, um médodo intuitivo do calculo deste valor poderia ser aproximar a área por um conjunto de rectângulos, cujas áreas são facilmente computáveis. Os rectângulos são calculados da seguinte forma:

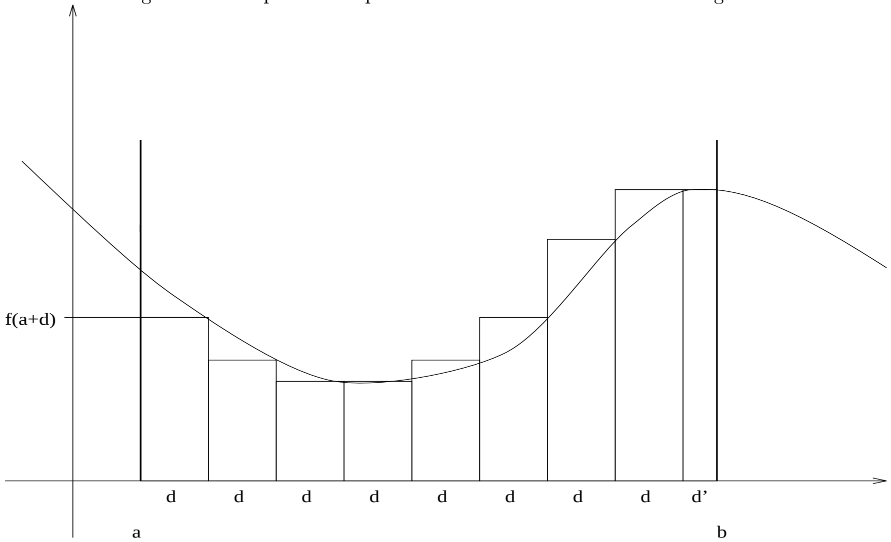
1. a, b são valores fornecidos portanto lidas pela programa.

2. f é uma função fornecida - codificado dentro do programa.

3. d é fornecido, e representa o comprimento da base dos rectângulos para construir.

4. Constroi-se recursivamente os rectângulos começando pelo ponto a até o ponto b. Isto é, o primeiro rectângulo tem por coordenadas {(a,0),(a+d,0),(a,f(a+d)),(a+d,f(a+d))}, o segundo tem por coordenadas {(a+d,0),(a+2d,0),(a+d,f(a+2d)),(a+2d,f(a+2d))}, etc ... De tal forma que $a + kd$ ($k \in \mathbb{N}$) não ultrapassa b. Trata-se por isso o último rectângulo de forma particular, caso seja necessário.

O valor do integral é entao aproximado pela soma das áreas de todos os rectângulos construídos.



Escreva um programa que permita aproximar o calculo do integral num intervalo duma função pelo método acima descrito.
