Departamento de Matemática — Universidade de Coimbra

Ano Lectivo de 2015/2016 | Métodos de Programação I | Exame 1/4/2016

Leia atentamente o enunciado de cada pergunta antes de iniciar a sua resolução.

Duração da prova: 1h30min | Os programas devem ser escritos em *C*

---

1. (a) Diga o que se entende por *metodologia de programação estruturada e descendente* (máximo 10 linhas);

   (b) Nesta metodologia qual é a importância dos sub-programas serem unidades funcionalmente independentes? Justifique (máximo 10 linhas).

2. Elabore um programa que calcule as raízes da equação quadrática de coeficientes inteiros: $a_0x^2 + a_1x + a_3 = 0$

3. Construa um programa, que leia uma sequência de caracteres, representando um número em notação romana, e que indique como resposta o correspondente número em notação decimal.

   Considere a notação romana na forma estritamente aditiva, isto é, `IIII` $= 4$.

   Os símbolos da numeração romana, e sua correspondência na forma decimal, são os seguintes:

`M`=1000	`D`=500	`C`=100	`L`=50
`X`=10	`V`=5	`I`=1	

4. Construa um sub-programa que, dado um número natural verifique se ele é um Número Primo, ou não, escrevendo uma mensagem correspondente.

5. Considere a seguinte função inteira de variável inteira:

$$f(n) = \begin{cases} 1 & \text{se } n < 0 \\ 1 & \text{se } n = 0 \\ \frac{2}{1} \times \frac{2}{3} \times \frac{4}{3} \times \frac{4}{5} \times \frac{6}{5} \times \frac{6}{7} \times \cdots \times \frac{2n}{2n-1} \times \frac{2n}{2n+1} & \text{se } n > 0 \end{cases}$$

   escreva um programa completo que:

   - peça e leia o valor de n;
   - usando um sub-programa, calcule o valor de $f(n)$;
   - escreva o valor de $f(n)$.

Resolução — A1

1. (a) Decompor o problema global em sub-problemas específicos (por tarefa bem determinada), provar que se cada sub-problema é resolvido correctamente, e se as várias soluções parciais podem ser combinadas de forma correcta, então o problema global será resolvido correctamente.

   Repetir este processo até atingir sub-problemas simples.

   (b) Os sub-programas como unidade funcionalmente independentes permite resolver cada um deles de forma independente.

2.
```c
#include <stdio.h>
#include <math.h>

int main() {
  int a0,a1,a2;
  float bd;
  float r1,r2,i1,i2;

  printf("Introduza três inteiros: ");
  scanf("%d%d%d",&a0,&a1,&a2);

  bd = a1*a1 - 4.0*a0*a2;

  printf("valor de bd %f\n",bd);
  if (bd < 0) { // duas raízes imaginárias
    r1 = -a1 / (2*a0);
    i1 = sqrt(-bd)/(2*a0);
    r2 = -a1 / (2*a0);
    i2 = -sqrt(-bd)/(2*a0);
    printf("Duas raízes imaginárias (%f,%f); (%f,%f)\n",r1,i1,r2,i2);
  }
  else {
    if (bd == 0) { // raiz real única
      r1 = -a1/(2*a0);
      printf("Raiz real dupla %f\n",r1);
    }
    else { // raízes reais
      r1 = (-a1 + sqrt(bd))/(2*a0);
      r2 = (-a1 - sqrt(bd))/(2*a0);
      printf("Duas raízes reais %f,%f\n",r1,r2);
    }
  }
  return(0);
}
```

3.
```c
#include <stdio.h>

int main() {
  char c;
  int res;
  int emDecimal;

  printf("Introduza um número em notação romana, terminando por '.'\n");
  scanf("%c",&c);

  emDecimal = 0;
  while (c != '.') {
    switch (c) {
    case 'v':
      emDecimal = emDecimal + 5000; break;
    case 'M':
      emDecimal = emDecimal + 1000; break;
    case 'D':
      emDecimal = emDecimal + 500; break;
    case 'C':
      emDecimal = emDecimal + 100; break;
    case 'L':
      emDecimal = emDecimal + 50; break;
    case 'X':
      emDecimal = emDecimal + 10; break;
    case 'V':
      emDecimal = emDecimal + 5; break;
    case 'I':
      emDecimal = emDecimal + 1; break;
    }
    scanf("%c",&c);
  }
  printf("O resultado é %d\n",emDecimal);
  return(0);
}
```

4.

```c
#include <stdio.h>


int primo(int n) {
  int i=2;
  int divisor=0;

  while (!divisor && i<n) {
    if (n % i == 0) divisor=1;
    i++;
  }
  return(!divisor);
}

int main() {
  int n;

  printf("Introduza um inteiro: ");
  scanf("%d",&n);

  if (primo(n)) {
    printf("O número %d é um número primo\n",n);
  }
  else {
    printf("O número %d não é um número primo\n",n);
  }

  return(0);
}
```