

CCM0118 — Computação I
CURSO DE CIÊNCIAS MOLECULARES — TURMA 22 — SEGUNDO SEMESTRE DE 2012
Segunda Prova — 25 de outubro de 2012

Nome: _____

Assinatura: _____

Nº USP: _____

Instruções:

1. Preencha o cabeçalho acima.
2. Não destaque as folhas deste caderno.
3. A prova tem cinco questões. Antes de começar a trabalhar verifique se o seu caderno de questões está completo.
4. A prova pode ser feita a lápis. Cuidado com a legibilidade.
5. A prova deve ser resolvida individualmente. Não é permitida a consulta a livros, apontamentos ou colegas.
6. Não é permitido o uso de folhas avulsas para rascunho.
7. A duração da prova é **1 hora e 40 minutos**.

Boa prova!

Questão	Valor	Nota
1	1,0	
2	2,0	
3	3,0	
4	3,0	
5	1,0	
Total	10,0	

Questão 1

(1,0 pontos)

Considere esta implementação recursiva da função de Fibonacci:

```
int fib(int n)
{
    if (n == 0)
        return 0;
    else if (n == 1)
        return 1;
    else return fib(n-1) + fib(n-2);
}
```

O cálculo de `fib(3)` provocará a seguinte sequência de chamadas da função:

```
fib(3)
  fib(2)
    fib(1)
    fib(0)
  fib(1)
```

Qual a sequência de chamadas provocada por `fib(5)`? Use indentação para representar o aninhamento de chamadas.

Questão 2

(2,0 pontos)

Escreva uma função com protótipo

```
float raiz_cubica(float x, float epsilon);
```

que recebe como parâmetros números reais $x > 0$ e $\varepsilon > 0$. A função devolve uma aproximação de raiz cúbica de x com precisão ε . A aproximação deve ser obtida por meio da sequência $r_0, r_1, \dots, r_n, \dots$, onde $r_0 = x$ e

$$r_{n+1} = \frac{1}{3} \left(2r_n + \frac{x}{r_n^2} \right)$$

A aproximação a ser devolvida será o primeiro valor r_{n+1} para o qual $|r_{n+1} - r_n| < \varepsilon$.

Questão 3

(1,5 + 1,5 pontos)

- (a) Escreva uma função `converte`, que recebe um intervalo de tempo dado em segundos e o converte para dias, horas, minutos e segundos. Essa função tem um parâmetro de entrada e quatro parâmetros de saída. O parâmetro de entrada é um inteiro não negativo t com o valor de um intervalo de tempo em segundos. Os parâmetros de saída são quatro inteiros não negativos com os valores de dias, horas, minutos e segundos correspondentes ao valor de t . Por exemplo, para $t = 100000$, a saída deve ser 1 (dia), 3 (horas), 46 (minutos) e 40 (segundos), pois $1 \times 24 \times 60 \times 60 + 3 \times 60 \times 60 + 46 \times 60 + 40 = 100000$.

A função `converte` deve ser declarada como `void` (ou seja, ela não deve devolver nada como valor) e deve receber cinco parâmetros. O primeiro desses parâmetros é o de entrada, que deve ser declarado como `int t`. Os quatro parâmetros seguintes são os endereços de (ponteiros para) quatro variáveis tipo `int` nas quais a função `converte` colocará os dias, as horas, os minutos e os segundos.

(b) Escreva um programa que resolva o seguinte problema:

Dados um número inteiro não negativo n e uma sequência com n números inteiros não negativos que representam intervalos de tempo expressos em segundos, converter cada um desses intervalos para dias, horas, minutos e segundos.

Utilize obrigatoriamente a função `converte` do item (a), mesmo que você não tenha escrito essa função.

Questão 4

(1,5 + 1,5 pontos)

Nesta questão você usará vetores para representar polinômios de coeficientes reais. Um polinômio $p(x) = a_0 + a_1x + \dots + a_nx^n$ (de grau n) será representado por um vetor de `floats` com os $n + 1$ coeficientes do polinômio (o primeiro elemento do vetor é o a_0 , o último elemento é o a_n).

Considere dada uma função com protótipo

```
float avalia_polinomio(int n, float a[], float x);
```

que devolve o valor do polinômio $p(x) = a_0 + a_1x + \dots + a_nx^n$ num ponto especificado. A função recebe como parâmetros o grau do polinômio, o vetor de coeficientes do polinômio e o ponto x no qual o valor do polinômio será calculado. Você não precisa escrever essa função. Suponha que ela já foi escrita e use-a à vontade.

- (a) Escreva uma função `integra_polinomio`, que recebe como parâmetro um inteiro não-negativo k e um vetor a com os $k + 1$ coeficientes reais de um polinômio de grau k . A função altera k e a de modo que eles passem a armazenar o grau e os coeficientes do polinômio com termo independente nulo ($a_0 = 0$) que é uma primitiva (integral indefinida) do polinômio de entrada. Suponha que o vetor a tem espaço suficiente para mais um coeficiente. (Essa hipótese é necessária porque o grau do polinômio integrado é maior que o do polinômio original.)

- (b) Escreva um programa que lê um inteiro positivo n , lê os coeficientes reais a_0, a_1, \dots, a_n de um polinômio de grau n , que por hipótese são todos maiores ou iguais a zero, e lê dois reais x_1 e x_2 , que por hipótese satisfazem as desigualdades $0 \leq x_1 \leq x_2$. O programa calcula (e imprime) o valor da área delimitada superiormente pelo gráfico do polinômio, delimitada inferiormente pelo eixo dos x , delimitada à esquerda pela reta vertical $x = x_1$ e à esquerda pela reta vertical $x = x_2$.

Utilize obrigatoriamente a função do item (a), mesmo que você não a tenha escrito. Caso você não tenha feito o item (a), escreva o protótipo da função `integra_polinomio`.

Questão 5

(1,0 pontos)

Escreva um programa que leia um inteiro positivo n , leia uma sequência de n números inteiros e imprima um segmento dessa sequência que tenha soma máxima. (Atenção: o n não faz parte da sequência!)

Exemplo: Considere a sequência 5, 2, -2, -7, 3, 14, 10, -3, 9, -6, 4, 1. O segmento sublinhado tem soma 33. Esse segmento tem soma máxima, pois a sequência não contém nenhum segmento com soma maior que 33.

Rascunho

(não destacar esta página do caderno de questões)