

MAC 115 – Introdução à Computação
Instituto de Física – Segundo Semestre de 2005

Prova 3 – 2/12/2005

Instruções:

1. Não destaque folhas do caderno de soluções.
2. A prova pode ser feita a lápis. Cuidado com a legibilidade.
3. Não é permitido o uso de folhas avulsas para rascunho.
4. Não é necessário apagar rascunhos no caderno de soluções.
5. A duração da prova é de 1 hora e 40 minutos.

Questão 1 (valor: 2 pontos)

Considere a seguinte função

```
double p(double x, double a[], int n)
{
    double temp = 0; int k;
    for (k = n; k >= 0; k--)
        temp = temp * x + a[k];
    return temp;
}
```

- (a) Suponha que temos um vetor c com $c[0] = -2$, $c[1] = -1$ e $c[2] = 1$. Qual é o valor devolvido pelas chamadas $p(2, c, 2)$, $p(-1, c, 2)$ e $p(1, c, 2)$?
- (b) Suponha que os elementos de um vetor c são tais que $c[i] = i + 1$ para todo $0 \leq i \leq n$. Encontre uma fórmula para o valor devolvido pela chamada $p(x, c, n)$. [Ignore erros de arredondamento.]

Questão 2 (valor: 4 pontos)

Nesta questão, vamos representar inteiros (possivelmente com um número grande de dígitos) em vetores. Por exemplo, para representar o inteiro

31415926535897932384626433832795028841971693993751,

armazenamos em um vetor, digamos v , os dígitos 1, 5, 7, 3, 9, 9, etc, nesta ordem, isto é, $v[0] = 1$, $v[1] = 5$, $v[2] = 7$, e assim por diante (é mais conveniente termos os dígitos na ‘ordem inversa’). A idéia é escrever um programa que soma dois inteiros (grandes) dados. Você deve supor que, nesta questão, os inteiros a serem somados têm no máximo 1000 dígitos.

- (a) Escreva uma função de protótipo

```
void leia_vetor(int v[], int n);
```

que lê as entradas de um vetor inteiro v de n elementos. Escreva também uma função de protótipo

```
void imprima_vetor(int v[], int n);
```

que imprime o vetor inteiro v de n elementos. Sua função `imprima_vetor()` deve imprimir os elementos de v na ordem ‘reversa’: primeiro $v[n-1]$, depois $v[n-2]$, etc, sem deixar espaço entre estes inteiros.

(b) Escreva uma função de protótipo

```
int some(int s[], int a[], int n_a, int b[], int n_b);
```

que recebe em a e, respectivamente, em b as seqüências de dígitos dos inteiro a e b (como explicado no início desta questão), e que devolve em s uma representação da soma $a + b$. A sua função deve receber em n_a e em n_b o número de dígitos de a e b , respectivamente, e deve devolver o número de dígitos na soma $a + b$.

Exemplo. Suponha que $a = 82434256$ e $b = 33752337$. Então, a sua função será chamada com

```
a = {6, 5, 2, 4, ... }  
b = {7, 3, 3, 2, ... }
```

e $n_a = n_b = 8$. A sua função deve então produzir o vetor

```
s = {3, 9, 5, 6, ... }
```

pois $a + b = 116186593$, e o valor devolvido por `some()` deve ser 9 ($a + b$ tem 9 dígitos).

(c) Escreva um programa que recebe dois inteiros positivos a e b com n_a e n_b dígitos, respectivamente, e determina a soma $a + b$. Os valores de n_a e n_b são dados pelo usuário, e os números a e b são fornecidos dígito por dígito, na ordem reversa (unidades primeiro, depois dezenas, depois centenas, etc).

Exemplo. Suponha que $a = 82434251$ e $b = 33752337$. Então, a entrada do seu programa será

```
8  
1 5 2 4 3 4 2 8  
8  
7 3 3 2 5 7 3 3
```

e a saída deve ser

```
82434251  
+  
33752337  
=  
116186588
```

Questão 3 (valor: 4 pontos)

Dada uma matriz inteira, definimos o *peso* de uma linha dessa matriz como sendo a soma do *menor* e do *maior* elemento dessa linha. Por exemplo, na matriz

$$A = \begin{pmatrix} -4 & 7 & 8 & 10 & -17 & 28 \\ 2 & 3 & -5 & 8 & 15 & 25 \\ -8 & 10 & 14 & -16 & 20 & 48 \\ 7 & 8 & 9 & 12 & 18 & -45 \end{pmatrix},$$

as linhas têm peso 11, 20, 32, e -27 .

(a) Escreva uma função de protótipo

```
void leia_matriz(int a[][NMAX], int m, int n);
```

que lê as entradas da matriz $m \times n$ inteira a . Escreva também uma função de protótipo

```
void imprima_matriz(int a[][NMAX], int m, int n);
```

que imprime a matriz $m \times n$ inteira a .

(b) Escreva uma função de protótipo

```
void menor_maior(int v[], int comp, int *pmin, int *pmax);
```

que recebe um vetor inteiro v e um inteiro $comp$ (o número de elementos de v), e devolve o menor e o maior elemento do vetor v em $*pmin$ e $*pmax$.

(c) Escreva uma função de protótipo

```
int processe_matriz(int a[][NMAX], int m, int n);
```

que recebe uma matriz $m \times n$ inteira a , e imprime o seguinte:

- o menor e o maior elemento de cada linha e o peso desta linha,
- o índice da linha de maior peso,
- o peso máximo encontrado.

(Em caso de haver mais de uma linha de peso máximo, `processe_matriz()` deve imprimir o índice de qualquer uma delas.) Ao ser chamada com a matriz A acima, sua função deve imprimir algo como

```
Pesos:
0: -17 28 peso = 11
1: -5 25 peso = 20
2: -16 48 peso = 32
3: -45 18 peso = -27
Linha de maior peso: 2
Peso maximo: 32
```

Finalmente, `processe_matriz()` deve devolver o índice da linha de maior peso (2, no exemplo acima). Você deve obrigatoriamente usar a função `menor_maior()` acima, mesmo que você não a tenha feito.

(d) Escreva um programa que recebe como entrada dois inteiros m e n (com $0 < m, n \leq 50$) e uma matriz $m \times n$ inteira A , e que, primeiro, imprime a matriz lida. Uma vez impressa a matriz, o seu programa deve chamar a função do item (c) para esta matriz. Seu programa deve ainda imprimir os elementos da linha de A com peso máximo.

Exemplo. Suponha que a entrada de seu programa é

```
4 6
-4 7 8 10 -17 28
2 3 -5 8 15 25
-8 10 14 -16 20 48
7 8 9 12 18 -45
```

A saída de seu programa poderia então ser algo como

Matriz lida:

```
-4  7  8  10 -17 28
  2  3 -5   8  15 25
 -8 10 14 -16  20 48
  7  8  9  12  18 -45
```

Pesos:

0: -17 28 peso = 11

1: -5 25 peso = 20

2: -16 48 peso = 32

3: -45 18 peso = -27

Linha de maior peso: 2

Peso maximo: 32

Elementos da linha de peso maximo:

```
-8 10 14 -16 20 48
```