Departamento de Informática - PUC-Rio
INF 1007 – Programação 2
P1 – 23/09/2010

Aluno: _____

Matrícula: _____ Turma: _____

Instruções:

1) Escreva seu nome completo, matrícula e turma em todas as folhas desta prova;
2) A prova deve ser completamente resolvida nas folhas que constam deste caderno, utilizando-se frente e/ou verso;
3) As questões podem ser resolvidas em qualquer ordem;
4) As soluções que não forem apresentadas nas páginas a elas destinadas devem ser identificadas com o número da questão a que se referem;
5) A prova pode ser feita utilizando-se lápis ou caneta (azul ou preta);
6) Todos os dispositivos eletrônicos (celulares, i-pods, etc) devem ser desligados.

Pontuação:

Questão	Item	Valor	Nota
1	A	2,0	
	B	2,0	
2	A	1,0	
	B	2,0	
	C	3,0	
Total		10,0	

INF 1007 – P1 – 23/09/10	
Nome:	
Matrícula:	Turma:

Questão 1

Sobre o valor bruto de uma venda têm que ser recolhidos ao governo vários impostos, que são percentuais definidos pela legislação. Os percentuais desses impostos são representados segundo a estrutura a seguir:

```
struct perctributacao {
      float perc_icms;
      float perc_pis;
      float perc_cofins;
};
typedef struct perctributacao PercTributacao;
```

a) **[Valor: 2,0 pontos]** Escreva em C a função *calcularImposto*, que recebe o valor bruto de uma venda (*vr_bruto*) e uma estrutura contendo os percentuais de impostos (*trib*), cada um deles devendo ser aplicado sobre o valor bruto da venda, e retorna uma estrutura com o valor dos impostos a serem pagos e o valor líquido da venda (*vr_liquido*). Um exemplo de uso pode ser visto na questão B, a seguir.

Os valores dos impostos a serem pagos sobre a venda são representados segundo a estrutura a seguir e são calculados aplicando-se os respectivos percentuais sobre o valor bruto da venda:

```
struct imposto {
      float vr_icms;
      float vr_pis;
      float vr_cofins;
};
typedef struct imposto Imposto;
```

Considere que a função sempre recebe valores válidos. O protótipo da função *calcularImposto* é o seguinte:

```
Imposto calcularImposto (float vr_bruto, PercTributacao trib,
                         float *vr_liquido);
```

b) **[Valor: 2,0 pontos]** Escreva um programa principal que leia do teclado o valor bruto da venda e os percentuais dos impostos cofins, icms e pis, nessa ordem, chame a função `calcularImposto`, criada anteriormente, para calcular a tributação relativa a venda e exiba na tela o valor dos impostos a serem recolhidos e o valor líquido da venda, todos com duas casas decimais. Ao final da execução do programa, a aparência da tela deve ser a seguinte:

```
100.00
0.076
0.02
0.0165

cofins = 7.60
icms = 2.00
pis = 1.65
valor liquido = 88.75
```

Observação: No item B, utilize a função desenvolvida no item A.

INF 1007 – P1 – 23/09/10	
Nome:	
Matrícula:	Turma:

Solução da Questão 1:

INF 1007 – P1 – 23/09/10	
Nome:	
Matrícula:	Turma:

Questão 2

Escreva em C as seguintes funções:

a) **[Valor: 1,0 ponto]** A função `Tamanho` recebe uma cadeia de caracteres (*s*) e retorna a quantidade total de caracteres da cadeia, INCLUINDO qualquer caractere de controle que faça parte da cadeia de caracteres. Seu funcionamento é DIFERENTE do funcionamento da função *strlen* da biblioteca padrão do C. A função `Tamanho` tem o seguinte protótipo:

```
int Tamanho (char *s);
```

b) **[Valor: 2,0 pontos]** A função `ConcatenaEspecial` recebe duas cadeias de caracteres (*s1* e *s2*) e copia a segunda cadeia (*s2*) para o final da primeira cadeia (*s1*) recebida como parâmetro, acrescentado UM espaço em branco entre as duas palavras. Ou seja, *ConcatenaEspecial ("Boa", "prova")* fará com que a primeira cadeia passe a ser "Boa prova" ao final de sua execução. Considere que a primeira cadeia (*s1*) tem o tamanho necessário para armazenar os caracteres relativos a segunda cadeia (*s2*). A função `ConcatenaEspecial` tem o seguinte protótipo:

```
void ConcatenaEspecial (char *s1, char *s2);
```

c) **[Valor: 3,0 pontos]** A função *`MontaCadeia`* recebe um vetor de ponteiros para palavras (cadeias de caracteres) e retorna uma cadeia de caracteres composta por todas as palavras do vetor, separadas por um espaço em branco. A cadeia de caracteres retornada tem que ter o exato tamanho necessário.

- A função *`MontaCadeia`* TEM que usar as funções desenvolvidas nos itens A) e B).
- A função *`MontaCadeia`* NÃO pode usar funções da biblioteca padrão do C relativas a *string.h*.
- A função *`MontaCadeia`* tem o seguinte protótipo:

```
char* MontaCadeia (char **palavras, int n);
```
onde:

  - → *palavras* é um vetor de ponteiros para cadeias de caracteres;
  - → *n* é a quantidade de elementos do vetor *palavras*. Considere que sempre n >= 1;
  - → A função deve retornar uma cadeia de caracteres ou NULL, caso não consiga alocá-la.

Se a função *`MontaCadeia`* recebesse um vetor de ponteiros para as cadeias de caracteres a seguir:

```
"Primeira"
"prova"
"Prog2"
"-"
"Boa"
"Sorte"
"!"
```

geraria a seguinte cadeia de caracteres:

```
"Primeira prova Prog2 - Boa Sorte !"
```

INF 1007 – P1 – 23/09/10	
Nome:	
Matrícula:	Turma:

Solução da Questão 2:

INF 1007 – P1 – 23/09/10	
Nome:	
Matrícula:	Turma:

Solução da Questão 2:

Protótipos de funções que podem ser úteis:

stdio.h:

```
int scanf (char* formato, ...);
int printf (char* formato, ...);
FILE* fopen (char* nome, char* modo);
int fclose (FILE* fp);
int fscanf (FILE* fp, char* formato, ...);
int fprintf (FILE* fp, char* formato, ...);
char* fgets(char* str, int size, FILE* fp));
int sscanf(char* str, char* formato, ...);
```

math.h:

```
double sqrt (double x);
double pow (double x, double exp);
double cos (double radianos);
double sin (double radianos);
```

stdlib.h:

```
void* malloc (int nbytes);
void* realloc (void *p, int nbytes);
void  free (void* p);
```

Não separe as folhas deste caderno.