



FACULDADE DE EDUCAÇÃO TECNOLÓGICA DO ESTADO DO RIO DE JANEIRO

PROVA

x	AV1	AV2	AVS	AVF
Professor: <i>Leonardo Soares Vianna</i>		Disciplina: <i>Algoritmos e Linguagens de Programação II</i>		Data: <i>24/10/2016</i>
Aluno:		Matrícula:		Turma: <i>A – Noite</i>
Nota:		Nota revista:		Visto:

Questão 01 [1,5 pontos]:

O código fornecido a seguir apresenta a implementação de uma função recursiva que manipula dois parâmetros inteiros:

```
int funcao (int A, int B) {  
    if (A >= B) {  
        return A + funcao (A-1, B+1);  
    } else {  
        return B-A;  
    }  
}
```

Considerando que na *main()* é executado o comando

x = funcao (15,3);

fornecer a sequência de chamadas realizadas à função, incluindo os parâmetros fornecidos e os valores retornados. Ao final, informar o valor da variável *x*.

Questão 02 [1,5 pontos]:

Considerando que *v* é um vetor contendo números reais ordenados crescentemente, analisar as funções apresentadas abaixo e **informar o objetivo** de cada uma delas:

```
int funcao1 (float v[], float a, int b, int c)  
{  
    int d;  
    if (b <= c) {  
        d = (b+c)/2;  
        if (v[d] == a) {  
            return d;  
        } else {  
            if (v[d] > a) {  
                return funcao1 (v, a, b, d-1);  
            } else {  
                return funcao1 (v, a, d+1, c);  
            }  
        }  
    } else {  
        return -1;  
    }  
}
```

```
int funcao2 (float v[], float n, int x)  
{  
    int t, w, z, y = funcao1 (v, n, 0, x-1);  
    if (y >= 0) {  
        w = 1;        z = v[y];        t = y-1;  
        while ((t >= 0) && (v[t] == n)) {  
            w++;        t--;  
        }  
        t = y+1;  
        while ((t < x) && (v[t] == n)) {  
            w++;        t++;  
        }  
        return w;  
    } else {  
        return 0;  
    }  
}
```

Questão 03 (ENADE 2014 – adaptada) [1,0 ponto]:

Observe o programa classificador ("sort"), em pseudocódigo, apresentado abaixo.

```
1  início  
2  variável texto nome[5]  
3  variável real nota[5]  
4  variável inteiro i, j  
5  variável real aux  
6  variável texto naux  
7  para i de 1 até 5  
8      escrever "Nome ", i, " = "  
9      ler nome[i-1]  
10     escrever "Nota ", i, " = "  
11     ler nota[i-1]  
12 fimpara  
13 para i de 0 até 4  
14     para j de i+1 até 4  
15         se nota[i] <= nota[j] então  
16             aux <- nota[i]  
17             nota[i] <- nota[j]  
18             nota[j] <- aux  
19             naux <- nome[i]  
20             nome[i] <- nome[j]  
21             nome[j] <- naux  
22 fimse  
23 fimpara  
24 fimpara  
25 para i de 1 até 5  
26     escrever nome[i-1], " : ", nota[i-1], "\n"  
27 fimpara  
28 fim
```

- i. Este programa classifica, em ordem:
- a () decrescente, notas de alunos e nomes de alunos de mesma nota.
 - b () alfabética crescente, nomes e notas de alunos de mesmo nome.
 - c () decrescente, notas de alunos.
 - d () alfabética crescente, nomes de alunos.
 - e () crescente, notas de alunos.
- ii. Analisando o funcionamento do algoritmo, de qual método de ordenação abordado nos seminários de AL2 esse código mais se aproxima? Justifique a sua resposta.

Questão 04 [2,0 pontos]:

Desenvolver uma função que, dadas duas *strings* *s1* e *s2*, determine o número de caracteres que são comuns às duas cadeias de caracteres.

Nota: sua solução deve, obrigatoriamente, fazer uso de recursividade.

Questão 05 [2,0 pontos]:

O histórico de um aluno é representado por um vetor de *structs* onde cada posição armazena o código da disciplina cursada, semestre e ano que a cumpriu e a média final na disciplina. Implementar uma função que exiba o histórico do aluno com o seguinte formato:

NomeDisciplina1 (código1)	MédiaDisciplina1
NomeDisciplina2 (código2)	MédiaDisciplina2
⋮	
NomeDisciplinaN (códigoN)	MédiaDisciplinaN

Coefficiente de rendimento: **CR**

Observações:

1. Para obter os dados da disciplina, um outro vetor de *structs* deve ser consultado. Este, por sua vez, armazena para cada disciplina do curso as seguintes informações: *código*, *nome* e *número de créditos*;
2. O coeficiente de rendimento consiste em uma média ponderada de todos os graus atribuídos às disciplinas cursadas, onde os pesos são representados pelo *número de créditos* da disciplina;
3. Todos os dados em negrito apresentados no formato do histórico devem ser obtidos a partir dos vetores.

Questão 06 [2,0 pontos]:

Fazer uma função que, dado um arquivo texto, determine quantas vezes a *string* *S* encontra-se no arquivo.