MC102—Algoritmos e Programação de Computadores, turmas A e B

Prova 3

Prof. Juliana de Santi

19 de Junho de 2012

Nome:	
RA:	Nota:

Instruções: Você pode fazer a prova a lápis (desde que o resultado final seja legível). Todas as respostas devem ser colocadas na folha de almaço. Não é permitida consulta a qualquer material manuscrito ou impresso. Em caso de fraude, todos os envolvidos receberão nota zero.

Dica: Os cabeçalhos das funções para arquivo são: fprintf(ponteiro-arq, string-formato, lista-variáveis), fscanf(ponteiro-arq, string-formato, &lista-variáveis), fopen(nome-arq, modo-abertura), fclose(ponteiro-arq).

1. (**2.0**) Mostre o que será impresso na tela, quando for executado o programa abaixo. No lugar de **D** na função **f2** use o último dígito do seu RA.

```
#include <stdio.h>

struct S1{
  int a;
  int b;
};

int f1(struct S1 v[], int n);
int f2(struct S1 *x);

int main(){
  struct S1 v[] = { {8,2}, {3,4}, {5,6}, {1,8} };
  int i;

  f1(v, 4);
  for(i=0; i<4; i++)
    printf("a=%d, b=%d \n",v[i].a, v[i].b);
}

int f1(struct S1 v[], int n){
  int i,j,m;
  struct S1 x;

  for(i=0; i<n ;i++){
    m = i;
    for(j=i+1; j<n; j++){
      if(v[i].a < v[j].a)
```

```
        m=j;
    }
    x = v[m];
    v[m] = v[i];
    v[i] = x;
  }

  for(i=0; i<n; i++)
    f2( &v[i] );
}//fim de f1

int f2(struct S1 *x){
  int a = D; //Subistitua D pelo último dígito do seu RA.
  (*x).b = (*x).a + a;
}//fim de f2
```

2. (**3.0**)

   (a) (**1.5**) Explique o funcionamento do algoritmo **bubbleSort**, utilizando o vetor $(15, 45, 1, -10, 67, 80, 4, 16)$ como exemplo. Mostre como o algoritmo procede para ordenar este vetor.

   (b) (**1.5**) Abaixo está o código de uma função em C que implementa o algoritmo **bubbleSort**.

```
void bubbleSort(int vet[], int tam){
  int i, j, aux;
  for(i=tam-1 ; i>0; i--){
    for(j = 0; j < i; j++){
      if(vet[j] > vet[j+1]){
        aux = vet[j];
        vet[j] = vet[j+1];
        vet[j+1] = aux;
      }
    }
  }
}
```

   Suponha que estamos trabalhando com dados de motoristas, armazenando o nome, o CPF, o RG e o número de pontos na carteira. Crie um tipo **struct** para representar um motorista. Altere o código do **bubbleSort** acima para que este receba um vetor de motoristas, e ordene este vetor em ordem **decrescente** de número de pontos na carteira.

3. (**3.0**) Dado um vetor **v** de inteiros com **n** elementos queremos encontrar a soma de todos os elementos deste vetor.

   (a) (**1.5**) Escreva um função recursiva em C que computa a soma dos elementos do vetor **v** contendo **n** inteiros.

   (b) (**1.5**) Mostre a memória durante a execução da sua função considerando que **v=(4,-1, 6, -10, 4)** e $\mathbf{n = 5}$.

4. (**2.0**) Suponha um **arquivo texto** contendo um inteiro por linha. Faça um programa que lê este arquivo texto chamado *entrada.txt* para a memória. O programa deve então criar um novo arquivo texto chamado *parOrdem.txt* contendo apenas os números **pares** e ordenados por valor. Use o **bubbleSort** da questão 2. Assuma que o arquivo *entrada.txt* possui no máximo 1000 números. **Dica:** Lembre-se que você pode usar o formato **%d** para ler/escrever números em arquivo texto.