



Instituto Federal de Educação, Ciência e Tecnologia de Brasília – Câmpus Taguatinga
Ciência da Computação – Programação de Computadores I
Prova II – 2º/2017 – Matrizes, ponteiros, registros, arquivos, recursividade
Prof. Daniel Saad Nogueira Nunes

Aluno: _____

Matrícula: _____

Data: 5 de junho de 2017

Duração da prova: 180 minutos

Tabela de notas (uso exclusivo do professor)

Questão	Pontos	Nota
1	2	
2	2	
3	2	
4	2½	
5	2	
6	2½	
Total	13	

Observações

- Esta prova tem o total de 3 páginas (incluindo a capa) e 6 questões.
- O número total de pontos é 13.
- Certifique-se de assinar todas as folhas de resposta bem como a capa da prova.
- Leia atentamente todas as questões da prova. A interpretação do problema é crucial para o desenvolvimento correto da resposta.
- Resoluções sem justificativa não serão consideradas.
- É vedado o uso de equipamentos eletrônicos, como celulares, notebooks entre outros.
- A prova será **anulada** e medidas disciplinares serão tomadas para os alunos que “colarem” durante a avaliação.

★ Certifique-se de assinar todas as folhas de resposta.

Questão 1 (2 pontos)

Desenvolva uma função que verifica se uma matriz de inteiros $B_{k \times l}$ é uma submatriz de $A_{n \times m}$, também uma matriz de inteiros. Ou seja, a sua função deverá retornar **verdadeiro** caso B seja submatriz de A e **falso**, caso contrário. Ela deverá possuir a seguinte assinatura:

```
int submatriz(int** A, int n, int m, int** B, int k, int l);
```

Questão 2 (2 pontos)

Um polígono é uma figura geométrica fechada com um número arbitrário de lados. Faça um programa completo que:

- (a) (0.5 pontos) Defina a estrutura de um polígono através de uma **struct**.
- (b) (0.5 pontos) Leia um polígono do teclado com um número arbitrário de lados do usuário (deve usar alocação dinâmica de memória).
- (c) (1 ponto) Calcule a razão perímetro vs número de lados utilizando uma função específica para este fim.

Questão 3 (2 pontos)

Elabore um programa completo que leia um inteiro n do usuário e imprima na tela a quantidade mínima de bits necessária para representar n . A solução adotada deverá obrigatoriamente utilizar **recursividade**.

Questão 4 (2½ pontos)

Suponha que um arquivo **binário** da Receita Federal esteja populado por vários registros de pessoa física, que consistem dos seguintes campos:

```
typedef struct pessoa_fisica_t{
    char nome_completo[50];
    char cpf [20];
    double renda_media;
    int nro_bens;
    int nro_dependentes;
} pessoa_fisica_t;
```

Faça um programa completo que:

- (a) (0.5 pontos) Realize a abertura deste arquivo no modo leitura + atualização. Suponha que ele se chama `dados.dat` e se encontra no diretório `/home/RF/2017/`.
- (b) (1 ponto) Imprima na tela o **nome** e **cpf** das pessoas que possuem uma renda familiar *per capita* (considerando o número de dependentes) superior a R\$ x reais. Você deverá implementar uma função com a seguinte assinatura:

```
void imprime_pessoas_renda_per_capita(FILE* arq, double x);
```

- (c) (1 ponto) Altere no arquivo o número de bens de todas as pessoas chamadas **str** para 0. Você deverá implementar uma função com a seguinte assinatura:

```
void altera_nro_bens(FILE* arq, char* str);
```

Dica: você pode utilizar a função fseek para posicionar a cabeça de leitura/escrita no registro correto.

Exemplo:

```
(...)  
fseek(arq,-20,SEEK_CUR) // Recua 20 bytes da posição atual  
fseek(arq,20,SEEK_SET) // Avança 20 bytes do início do arquivo  
fseek(arq,-20,SEEK_SET) // Recua 20 bytes do fim do arquivo  
(...)
```

Questão 5 (2 pontos)

De acordo com ponteiros:

- (a) (1 ponto) Faça uma função que aloque dinamicamente uma matriz $A_{n \times m}$ do usuário e retorne a matriz alocada. Ela deve possuir a seguinte assinatura:
- ```
double** aloca_matriz(int n,int m);
```
- (b) (1 ponto) Crie uma função que, dado um vetor contendo as médias finais dos alunos de PC1, determine quais foram a maior e menor notas. Estes dois valores devem ser modificados por “referência”, isto é, a sua função deverá possuir a seguinte assinatura:
- ```
void maior_menor_nota(double* notas,int tam, double* menor, double* maior);
```

Questão 6 (2½ pontos)

Desenvolva um programa completo que receba um caminho de arquivo texto através de parâmetros de linha de comando e modifique o arquivo convertendo todas as suas vogais minúsculas para maiúsculas. Seu programa não deverá criar um novo arquivo, isto é, as modificações devem ser feitas no arquivo passado como parâmetro.

Dica: É permitida a utilização das funções:

```
int toupper(char c); // retorna a versão maiúscula do caracter 'c'  
int getc(FILE* arq); // retorna o próximo caractere do arquivo  
int ungetc(int c,FILE* arq); // Coloca o caractere 'c' na posição  
                           //imediatamente anterior. Inverso de getc.  
int putc(char c, FILE* arq); // Coloca o caractere 'c' no arquivo
```

I have no idols. I admire work,
dedication and competence

Ayrton Senna