

Universidade Federal Fluminense
Departamento de Ciência da Computação
Programação Estruturada
Abril de 2018

Lista de Exercícios I

1) Escreva um programa em C que receba um array de inteiros e gere uma representação compacta usando o método RLE (*Run Length Encoding*). Para cada ocorrência de um número inteiro o método RLE armazena o valor correspondente juntamente com o número de ocorrências consecutivas. Exemplo:

```
1 2 3 5 5 5 7 7 7 7 7 2 2 2 0 0 0 0 9 9 9 12 12 12 4 4  
1 1 2 1 5 3 7 5 2 3 0 5 9 3 12 3 4 2
```

2) Escreva funções em linguagem C que efetue as seguintes operações sobre cadeias de caracteres:

- Calcular o comprimento da cadeia de caracteres *s*.
- Copiar uma cadeia de caracteres *s1* para uma cadeia *s2*.
- Inverter uma cadeia de caracteres *s*.
- Inserir uma cadeia de caracteres *s1* a partir da *i*-ésima posição da cadeia de caracteres *s2*, se *s2* possuir espaço suficiente para a inserção.

3) Escreva um programa em C que receba uma cadeia de caracteres representando um número em ponto flutuante e o converta para um valor de tipo float.

4) Escreva uma função em C que, para uma dada matriz com ***n* × *m*** números inteiros entre 0 e 99, construa um histograma de frequência de ocorrência de cada elemento representada por um array de inteiros com 100 elementos.

5) Escreva um programa que leia um arquivo texto e imprima cada uma de suas linhas.

6) Escreva um programa que leia um arquivo texto e armazene em um tipo estruturado as seguintes medidas.

- a) Número de letras do arquivo
- b) Total de palavras
- c) Número de espaços em branco
- d) Número de linhas
- e) Tamanho médio das palavras em número de caracteres.
- f) Maior palavra.

```
struct estatistica_texto {  
    int num_letras;  
    int num_palavras;  
    int num_espacos;  
    int num_linhas;  
    float tam_medio_palavras;  
    char maior_palavra[MAX_PALAVRA];  
};
```

7) Descreva em C um tipo estruturado que represente um conjunto de inteiros e forneça funções para união, interseção e diferença entre conjuntos.

```
typedef struct conjunto_int {int num_elementos;  
    int elementos[MAX_ELEMENTOS];} conjunto_int_t;  
void uniao(conjunto_int_t c1, conjunto_int_t c2, conjunto_int_t * uniao);  
void intersecao(conjunto_int_t c1, conjunto_int_t c2, int m, conjunto_int_t *  
intersecao);  
void diferenca(conjunto_int_t c1, conjunto_int_t c2, conjunto_int_t *diferenca);
```

8) No jogo conhecido como “Conway's Game of Life” é possível verificar a criação automática de padrões a partir de um sistema computacional. A ideia é que padrões estáticos ou oscilatórios (desenhos, formas geométricas, etc) são automaticamente criados após uma certa

quantidade de gerações de evolução no jogo. Hoje em dia, o jogo é explorado para a descoberta automática de evoluções de formas de vida simples, e também para composição automática de músicas, entre outras tarefas.

Implemente um programa simplificado que torne possível a criação de tais padrões. Para tanto, assuma que seu programa manipulará uma matriz binária (seus elementos são apenas 0 ou 1) de duas dimensões, lida de um arquivo texto. Na matriz, o valor 1 indica uma “célula viva” e o valor 0 indica uma “célula morta”. A matriz binária evoluirá segundo as regras abaixo:

- 1 – Qualquer célula viva com menos do que dois vizinhos vivos deve morrer (*subpopulação*).
- 2 – Qualquer célula viva com dois ou três vizinhos vivos, continua viva.
- 3 – Qualquer célula viva com mais do que três vizinhos vivos, morre (*superpopulação*).
- 4 – Qualquer célula morta com exatamente três vizinhos vivos se torna uma célula viva (*reprodução*).

Uma célula em qualquer matriz sempre terá oito vizinhos, que são as células que estão na sua posições adjacentes, na vertical, horizontal e nas diagonais, considerando uma matriz circular. Entretanto, para facilitar, **considere que as células das bordas não sofrerão modificações**. Suponha, por exemplo, que uma matriz 3x4, foi inicializada da seguinte forma:

```
1 1 0 0
0 1 0 0
1 1 0 0
```

Após uma geração, teremos a seguinte matriz, uma vez que o elemento vivo na posição (1,1) possui quatro vizinhos vivos e o elemento morto na posição (1,2) possui 3 vizinhos vivos:

```
1 1 0 0
0 0 1 0
1 1 0 0
```

a) Descreva os tipos estruturados apropriados para armazenar as informações necessárias à simulação do jogo.

b) Suponha a existência de um arquivo texto contendo na primeira linha as duas dimensões da matriz binária e nas linhas seguintes a enumeração das linhas da matriz, contendo zeros e uns. Escreva uma função em C que leia os dados do arquivo texto para a representação dos dados na memória principal.

c) Escreva uma função em C que execute um passo da simulação, isto é dada uma configuração, na geração $k-1$, descrita por uma matriz M , produza uma nova configuração, na geração k , usando as regras estabelecidas .

9) Considere o programa abaixo. Sem executá-lo, escreva o que será impresso na tela do console.

```
#include <stdio.h>
#include <stdlib.h>

struct ponto{float a;
             float b;};

int main()
{
    int a = 10;
    int b = 5;
    int *c = &a;
    int **d = &c;
    char *e = "inconstitucionalidade";
    char *f = &e[*c];
    struct ponto p={5.0,3.0};
    struct ponto *p1=&p;
    float *g = (float*)p1;
    unsigned int v[12] = {0,1,2,4,8,16,32,64,128,256,512,1024};
    unsigned char * w = (unsigned char*)v;

    printf("%d %d\n",a,b);
    printf("%d\n",a+*c);
    printf("%d %d\n",*c+**d);
    printf("%s\n",f);
    printf("%s\n",(f+b));
    printf("%f\n",p1->a);
    printf("%f %f\n",g[0],g[1]);

    for (int i=0;i<12*sizeof(int)/sizeof(unsigned char);i++)
        printf("%d ",*(w+i));
    printf("\n");

    return 0;
}
```

10) Escreva na linguagem C um programa que implemente o jogo de cartas 21 (Blackjack)
<https://pt.wikipedia.org/wiki/Blackjack>.