Programação de Computadores I (ICP131)

Prof. Ronald Souza – IC/UFRJ

Segunda Prova – 10/07/2023

Questão 1) (1.0 ponto) O que o código abaixo imprimirá? No seu caderno de respostas, **informe claramente** a alternativa que você escolheu dentre as alternativas a seguir.

a) vexame c) exam e) maxe
b) exame d) emaxe

```
#include <stdio.h>
void imprimir(char s[], int pos) {
    if (!pos)
       return;
    imprimir(s, pos - 1);
    printf("%c", s[pos]);
}
int main() {
    char palavra[] = "vexame";
    imprimir(palavra, sizeof(palavra) - 1); return 0;
}
```

Questão 2) (2.0 pontos) Considere um mapa de ligações entre cidades representado em uma matriz M quadrada, de dimensão NxN, onde o conteúdo da célula M(i,j) indica se existe ou não uma estrada que liga a cidade *i* à cidade *j* **nessa direção** (i.e., de *i* para *j*). Quando uma estrada da cidade *i* para a cidade *j* existe, a célula M(i,j) recebe valor 1; caso contrário, M(i,j) recebe valor 0.

Escreva uma função em C que receba como entrada uma matriz M de ligações e sua dimensão N, e retorne o índice da matriz referente à cidade com o **menor** número de ligações que **chegam até ela**. Em caso de empate, basta retornar qualquer um dentre os índices válidos.

Questão 3) (1.5 ponto) Considere a função definida abaixo. Essa função recebe dois vetores como entrada e deveria imprimir na tela os elementos desses dois vetores de forma intercalada, i.e., o primeiro elemento do primeiro vetor, depois o primeiro elemento do segundo vetor e assim sucessivamente. Por exemplo, se entrarmos com os vetores vetA = [1, 2, 3, 4, 5, 6, 7, 8, 9] e vetB = [0, 0, 0, 0, 0, 0, 0, 0, 0], a sequência de saída será: 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0

```
void intercala(int vetA[], int tamA, int vetB[], int tamB) {
    int a=0, b=0; //indices dos vetores
    for(int i=0; i<(tamA+tamB); i++)
        if(i%2 == 0) { printf("%d ", vetA[a]); a++;} //prox elemento vem de vetA
        else         { printf("%d ", vetB[b]); b++;} //prox elemento vem de vetB
}
```

Ocorreu, entretanto, que ao ser chamada com os vetores de entrada: vetA = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1] e vetB = [2, 2, 2, 2, 2], a sequência impressa na tela foi: 1 2 1 2 1 2 1 2 1 2 1 0 1 4196381 1.

Explique o que está errado na função acima e possíveis formas de se obter uma solução correta.

Questão 4 (2.0 pontos) A série infinita mostrada abaixo pode ser usada para estimar o valor de **log(1 + *x*),** sendo **−1 < *x* <= 1**. Usando essa série, escreva uma **função recursiva** em C para estimar **log(1 + *x*)**.

Tanto o valor de ***x*** quanto o número ***n*** de **elementos da série** deverão ser passados como argumentos para a função. Considere que ***x*** e ***n*** de entrada já são valores defendidos, e portanto válidos.
É permitido usar a função pow() da biblioteca <math.h>.

$$\log(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \frac{x^5}{5} ...$$

Questão 5) (3.5 pontos) Considere um jogo de cartas para N jogadores, N > 1, onde

- são utilizadas **somente** as cartas que são numeradas, ou seja, as cartas de 2 até 10;
- uma carta vale, em pontos, **exatamente o seu número** se o seu naipe **não** for copas**.**
- **cartas de copas** valem **duas vezes o seu número**. Por exemplo, um "3 de espadas" vale 3 pontos; um "3 de copas" vale 6 pontos.
- a cada rodada, **um jogador por vez**, do primeiro ao N-ésimo, revelará uma única carta para os demais jogadores.
- o jogador cuja carta revelada é a de maior valor recolherá **todas** as N cartas reveladas **naquela rodada** e conquistará seus respectivos pontos.
- uma carta já revelada **não pode ter sua pontuação repetida** por outro jogador. Por exemplo, se o primeiro jogador revela um "8 de ouros", sua carta vale 8 pontos. Portanto, nenhum outro jogador subsequente poderá, **naquela rodada**, revelar outra carta de 8 pontos (nesse caso, "8 de espadas", "8 de paus" ou "4 de copas").

O jogo será transmitido online e deseja-se informar quantos pontos cada jogador revelou a cada rodada.

a) (2.0 pontos) Implemente uma função em C que receba um vetor com todas as N Cartas reveladas, e forneça o status da rodada. Especificamente, sua função deverá:

- utilizar a **estrutura de dados** `Carta` **já fornecida** mais abaixo.
- ter a seguinte assinatura (onde N é o tamanho do vetor '`reveladas`'):

```
void statusRodada(Carta reveladas[], int N);
```

- **ordenar** o vetor '`reveladas`' em ordem crescente de **pontos**
- para cada carta do vetor (agora já ordenado!), imprimir o **nome do jogador** que a revelou e **quantos pontos ela vale**.

DICA: Uma função auxiliar que, dada uma carta, retorna quantos pontos ela vale será bem útil!

```
typedef struct {
    char naipe   //'o','e','c' ou 'p'(ouros, espadas, copas ou paus, respec.).
    int numero;  //Número da carta, entre 2 e 10.
    char jogador[100]  //Nome do jogador que a revelou;
} Carta;
```

b) (1.5 pontos) Do 2º jogador em diante, cada jogador deve levar em conta se a carta que pretende jogar pode de fato ser revelada, pois a mesma **não** pode valer a mesma quantidade de pontos de nenhuma das cartas já reveladas até o momento. **Escreva uma função** que recebe uma carta que o jogador da vez pretende revelar e o conjunto de cartas reveladas até o momento **já ordenado por pontos**, e retorna 1 caso a carta intencionada possa de fato ser jogada ou 0 caso contrário. A assinatura da sua função é:

```
int possivel(Carta intencao, Carta jaReveladas[], int t);
```
//t é o total de cartas já reveladas.