



PUC Minas

Pontifícia Universidade Católica de Minas Gerais
Instituto de Ciências Exatas e Informática (ICEI)
Engenharia de Computação / Engenharia de Software
Disciplina: Algoritmos e Estruturas de Dados I (AED I)

Segunda Prova Individual – 25 pontos

Nome:

Matrícula:

Instruções para os alunos:

- **A prova é individual e sem consulta;**
- **A interpretação faz parte da prova;**
- Deve ser feita, de preferência, a caneta;
- Caso use lápis, não cabe recurso de revisão após a prova ser entregue;
- Valor de cada questão apresentado na mesma;
- Lembre-se de assinar a lista de presença.

BOA PROVA!

1) (5 pontos) Considerando o código abaixo, mostre o que será impresso na tela após a execução do programa.

```
int a = 3;
float b = 1;

void func2(int b, float c, float *m){
    b = func3(a, &c);
    (*m) = a + b;
    printf("(func2) a=%.2f b=%.2f c=%.2f m=%.2f \n",
           (float)a, (float)b, (float)c, (float)(*m));
}

int func3(float a, float *m){
    b += (*m) + 1;
    (*m) += 3;
    printf("(func3) a=%.2f b=%.2f m=%.2f \n",
           (float)a, (float)b, (float)(*m));
    return a + b;
}

int main(){
    int a = 10;
    float res;
    func2(a, b, &res);
    printf("(main) a=%.2f b=%.2f res=%.2f \n", (float)a, (float)b, (float)res);
}
```

Apresente um algoritmo em linguagem de Programação C para cada um dos **4 (quatro)** problemas propostos a seguir. A correção irá considerar:

- o atendimento ao problema proposto;
- a qualidade da solução lógica;
- a codificação do programa e suas bibliotecas;
- a endentação (alinhamento) do código e comentários pontuais nos algoritmos.
- a escolha adequada da estrutura de repetição e recursos de modularização;

2) (5 pontos) Implemente uma função que recebe um vetor v com n inteiros e retorna um vetor p , que contém os valores de v sem repetição. Além do retorno padrão (return) sua função deve retornar **por referência** a quantidade de valores em p .

3) (5 pontos) Implemente a função abaixo, que receba como parâmetro uma string e converte, individualmente, cada caractere para maiúsculo. A função deverá retornar a string convertida para maiúsculo. A string a ser retornada deverá ser criada utilizando o comando malloc.

```
char* capitalizeString(char *vetor, int tamanho);
```

4) (5 pontos) Implemente uma função/procedimento *transposta* que recebe **uma matriz** M de valores reais e dimensão linhas x colunas. Sua função/procedimento deve retornar uma matriz T de dimensões colunas x linhas, cujos valores correspondem à **matriz transposta** de M . Lembre-se que uma matriz transposta corresponde àquelas cujas linhas se transformam em colunas e cujas colunas se transformam em linhas.

Exemplo: Supondo $M = \begin{matrix} 1 & 2 & 3 & 4 \end{matrix}$, então $T = \begin{matrix} 1 & 5 \\ 5 & 6 & 7 & 8 \\ 2 & 6 \\ 3 & 7 \\ 4 & 8 \end{matrix}$

5) (5 pontos) Implemente uma função/procedimento que encontra o maior prefixo comum em um conjunto de palavras, armazenadas em uma matriz de caracteres (cada linha da matriz representa uma palavra). A função/procedimento deve retornar um vetor com o maior prefixo. Caso não seja encontrado nenhum prefixo comum, a função/procedimento deve retornar um vetor de caracteres vazio.

Considerações:

- Considere que todas as palavras são compostas de caracteres de 'a' a 'z';
- Cada linha da matriz `palavras` contém uma única palavra com pelo menos uma letra;
- Considere que as palavras na matriz estão ordenadas em ordem alfabética;
- Considere que a quantidade de palavras na matriz seja sempre 3.

Exemplos:

Entrada	Saída
palavras = ["abacate", "abaco", "abóbora"]	"ab"
palavras = ["abacate", "abaco", "acordo"]	"a"
palavras = ["abacate", "abaco", "brasil"]	""