

# FOLHA DE QUESTÕES

|  |  |  |                    |   |              |  |                                |  |                                |  |                   |  |  |
|--|--|--|--------------------|---|--------------|--|--------------------------------|--|--------------------------------|--|-------------------|--|--|
| <b>CURSO:</b><br>CIÊNCIA DA COMPUTAÇÃO   |  |  |                    | <b>DISCIPLINA:</b><br>ESTRUTURAS DE DADOS 1 |              |  |                                |  |                                |  |                   |  |  |
| <b>ASS.:</b>                             |  |  |                    | <b>NOME:</b>                                |              |  |                                |  |                                |  |                   |  |  |
| <b>Professor: MATHEUS BANDINI</b>        |  |  |                    |   |              |  |                                |  |                                |  |                   |  |  |
| <b>DATA LIMITE:</b><br><b>30/04/2021</b> |  |  | <b>Nº de ordem</b> |   | <b>GRAU:</b> |  | <b>PROVA:</b><br><b>A1-AIA</b> |  | <b>TURMA</b><br><b>1INF34A</b> |  | <b>MATRÍCULA:</b> |  |  |
|  |  |  |                    |   |              |  |                                |  |                                |  |                   |  |  |

**LEIA ATENTAMENTE ÀS INSTRUÇÕES:**

- 1) A prova deve ser entregue em um arquivo único prova\_a1.c com um arquivo auxiliar prova\_a1.h. O arquivo .c deve possuir um menu no qual permite-se escolher qual questão deve-se executar, além de uma opção para sair do programa.
- 2) A entrega deve ser feita exclusivamente pelo ambiente do Canvas

**Questão 1 (1,0 ponto)** Crie uma função chama *reports* que receba um vetor de estruturas de alunos de um determinado curso. A estrutura deve conter o código do aluno, o seu nome, a nota da primeira prova, a nota da segunda prova e nota da terceira prova. A função deverá:

- Permitir ao usuário entrar com os dados de 5 alunos.
- Encontrar e imprimir os dados do aluno com maior nota da primeira prova.
- Encontrar e imprimir os dados do aluno com maior media geral.
- Encontrar e imprimir os dados do aluno com menor media geral.
- Para cada aluno diga se ele foi aprovado ou reprovado, considerando o valor 6 para aprovação. Apresente esta informação na forma de uma tabela.

Sugere-se que seja criada uma função para popular os dados dos alunos. A estrutura e as assinaturas das funções devem ser as seguintes:

```
typedef struct
{
    long code;
    char name[100];
    float grade1;
    float grade2;
    float grade3;
} Student;

void populate_students (Student *students)
void reports (Student *students)
```

**Dica:** Para ler strings contendo espaços, utilize a instrução: `scanf("%[A-Z a-z]",str);`

**Questão 2 (1,0 ponto)** Crie uma função chamada *tournament* que recebe um vetor de estruturas como parâmetro. Cada elemento da estrutura irá conter o nome de um clube de futebol e as respectivas quantidades de vitórias, empates e derrotas e saldo de gols. A função deve retornar o índice do clube campeão. Considere que cada vitória vale 3 pontos e que cada empate vale 1 ponto. Clubes derrotados não somam pontos. Caso haja empate em pontuação, o desempate deve ser feito pelo saldo de gols. Persistindo o empate, o vencedor é o primeiro clube verificado. Considere ainda que uma constante P representa o número de partidas. Logo, a soma da quantidade de vitórias, empates e derrotas deve ser igual a P. Sugere-se que seja criada uma função para popular os dados dos times. A estrutura e as assinaturas das funções devem ser as seguintes:

```
typedef struct
{
    char name[100];
    int victories;
    int draws;
    int defeats;
    int goal_diff;
} Team;

void populate_teams (Team *teams)
int tournament (Team *teams)
```

**Dica:** Para ler strings contendo espaços, utilize a instrução: `scanf("%[A-Z a-z]",str);`

**Questão 3 (1,0 ponto)** Escreva uma função que recebe como parâmetros um vetor de inteiros `v`, o número de elementos dele `N` e ponteiros para variáveis nas quais devem ser armazenados os valores máximo e mínimo do vetor. Lembre-se que, como resultado do processamento em questão, a obtenção dos valores máximo e mínimo devem ser visíveis de dentro da função principal. A assinatura da função deve ser:

```
void max_min (int *v, int N, int *max, int *min)
```

**Questão 4 (1,0 ponto)** A Busca Binária (*Binary Search*) é um algoritmo que utiliza a técnica de Divisão e Conquista em um vetor ordenado para encontrar um valor. O seu funcionamento é análogo ao de uma busca por uma palavra em um dicionário. A figura 1 ilustra um exemplo de busca binária em um vetor ordenado pelo número 80.

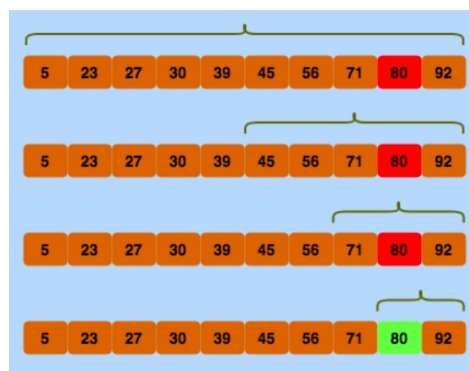


Figura 1: Exemplo de funcionamento do algoritmo Busca Binária

Implemente uma função recursiva que realize a Busca Binária em um vetor ordenado de valores inteiros. A função deve retornar o índice na qual o elemento buscado for encontrado, ou 1, caso o elemento não esteja presente no vetor e recebe, como argumentos, o vetor de inteiros, o início e o fim do intervalo analisado a cada recursão, além do valor a ser buscado. A assinatura da função deve ser:

```
int bin_search (int *v, int begin, int end, int value)
```

**Questão 5 (1,0 ponto)** Implemente uma função recursiva que, dados dois números inteiros `x` e `n`, calcula o valor do produto  $x \cdot n$ .