

ALUNOS:

1) Relacione as colunas, sendo que os números da primeira coluna podem aparecer mais de uma vez na segunda coluna.

1) Função recursiva	Tempo de Execução $O(n)$ – Resp:
2) Busca sequencial	Tempo de Execução $O(\log n)$ – Resp:
3) Busca binária	Executada sobre vetores ordenados. Resp:
	A implementação dela pode acarretar gasto maior de memória, já que durante o processo de execução da função muitas informações devem ser guardadas na pilha de execução. Resp:
	Executada para qualquer vetor. Resp:

2)

```
#define DIM 5
int main() {
    int v[DIM] = {6, 76, 2, 12, 5}, i, soma, *p;

    soma = 0;
    p = v;
    for (i = 0; i < DIM; i++)
        soma = soma + p[i];

    printf("\n soma: %d", soma);
    return 0;}
```

O código apresentado está errado, pois a variável *p* é um ponteiro e, por isso, o acesso aos elementos do vetor deve ser feito pela instrução **(p+i)* e não *p[i]*.

Escolha uma opção: () Verdadeiro () Falso

3) Suponha que você vá executar um programa cujo executável tem o nome de **MAT2023**. Esse programa foi escrito com a função principal *main* recebendo parâmetros *argc* (arguments count) e *argv* (arguments vector). Se você executá-lo como mostrado na linha a seguir, qual o valor de **argc**? Desenhe o parâmetro **argv** com o seu conteúdo.

```
> ./MAT2023 2 7
```

4) Suponha que *v* é um vetor. Descreva a diferença conceitual entre as expressões *v[3]* e *v+3*. Resposta:

5) Suponha que você tenha um vetor de inteiros *A* já preenchido e precisa imprimir o seu conteúdo. Qual trecho de código é mais eficiente para realizar tal tarefa, *A* ou *B*? Justifique sua resposta. Você pode usar exemplos, instruções para explicar.

Trecho A
`int i;
int vetor[5]={2,4,6,8,10};`

Trecho B
`int vetor[5]={2,4,6,8,10};int*ptr;
int *fim = vetor+5;`

```
for(i = 0; i < 5; i++)
    printf("%d ", vetor[i]);
```

```
for(ptr = vetor; ptr < fim; ptr++)
    printf("%d ", *ptr);
```

6) (Apostila UFMG) Após analisar a sequência de instruções abaixo de um programa em C, escolha a alternativa **falsa**:

```
int *pti;
int i = 10;
pti = &i;
```

- a. pti armazena o endereço de i
- b. *pti é igual a 10
- c. ao se executar *pti = 20; i passará a ter o valor 20
- d. ao se alterar o valor de i, *pti será modificado
- e. pti é igual a 10

7) (Apostila UFMG) Se i e j são variáveis inteiras e pi e pj são ponteiros para inteiro, qual atribuição **é ilegal**?

- a. pi = &i;
- b. *pj = &j;
- c. pj = *&j;
- d. i = *&j;
- e. i = (*pi)+++*pj

8) Na sequência de sequência de instruções abaixo:

```
float f;
float *pf;
pf = &f;
scanf("%f", pf);
```

- a. Efetuamos a leitura de f
- b. Não efetuamos a leitura de f
- c. Temos um erro de sintaxe
- d. Deveríamos estar usando &pf no scanf
- e. Nenhuma das opções anteriores

9) Complete o trecho de código abaixo que soma todos os elementos de uma matriz de inteiros $M_{DIM \times DIM}$, utilizando ponteiros e APENAS UMA estrutura de repetição, de modo que o acesso a uma posição da matriz seja feita com o menor número de processamentos aritméticos. Assuma que a matriz é estática, de ordem DIM e está completamente preenchida.

```
int soma = 0;
int* ptr;
for(_____; _____; _____)
{
    soma = soma + _____;
}
```

10) (Apostila UFMG) Analise o conjunto de instruções abaixo de um programa C:

```
int *pti;
int veti[]={10,7,2,6,3};
pti = veti;
```

Qual afirmativa **é falsa**?

- a. *pti é igual a 10
- b. *(pti+2) é igual a 2
- c. pti[4] é igual a 3
- d. pti[1] é igual a 10
- e. *(veti+3) é igual a 6

