

Pergunta 1

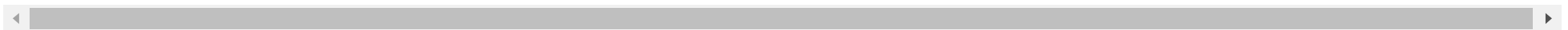
1,44 em 1,44 pontos

Seja f a função de espalhamento ou mapeamento e x a chave, o endereço de memória será atribuído por $f(x)$. Os valores serão distribuídos em um vetor de N posições, sendo formado em um intervalo entre 0 e $N-1$.

Sobre a utilização da função de mapeamento ou função *hash*, avalie se são (V) verdadeiras ou (F) falsas as afirmativas a seguir.

- I. () Utilizada para guardar uma coleção de dados.
- II. () Utilizada para obter os registros de maneira rápida.
- III. () Utilizada para acessar os arquivos no computador.
- IV. () Utilizada para ter acesso a uma determinada aplicação.

Assinale a alternativa que apresenta a sequência CORRETA.

**Pergunta 2**

1,44 em 1,44 pontos

É dada a implementação da função de hash abaixo apresentada na videoaula. Considere que nesta implementação estamos simplesmente garantindo que não colocaremos um registro fora dos limites do vetor mas não há mecanismos de controle de colisões. Indique qual é alternativa correta que exemplifica porque Pedro e Paulo não aparecem no vetor depois de inserir todos os alunos, como mostra a Figura 1.

```
int main(){
    Hash alunosHash(10);
    int   ras[7]   = {
        12704, 31300, 1234,
        49001, 52202, 65606,
        91234};
    string nomes[7] = {
        "Pedro", "Raul", "Paulo",
        "Carlos", "Lucas", "Maria",
        "Samanta"};
```

Apresentação do vetor após as inserções:

```
0:31300, Raul
1:49001, Carlos
2:52202, Lucas
3:-1,
4:91234, Samanta
5:-1,
6:65606, Maria
7:-1,
8:-1,
9:-1,
```

Figura 1. Resultado das inserções

Pergunta 3

1,44 em 1,44 pontos

É dada a implementação da função de hash abaixo (incompleta) apresentada na videoaula. Considere que nesta implementação estamos simplesmente garantindo que não colocaremos um registro fora dos limites do vetor (considera a não existência de colisões). Indique qual é alternativa correta para a linha 15 do código a seguir:

```
14 - int Hash::getHash(Aluno aluno){  
15 |  
16 }
```

Pergunta 4

1,42 em 1,42 pontos

Um procedimento natural para resolver os problemas de colisões consiste em guardar as chaves sinônimas em listas encadeadas. Existem duas opções: as listas podem se localizar no exterior da tabela ou compartilhar o mesmo espaço da tabela.

O encadeamento exterior consiste em manter _____, uma para cada endereço-base possível. Os _____ correspondentes aos endereços-base serão apenas os principais dessas listas. Um campo para o encadeamento deve ser adicionado a cada nó. A _____ interna consiste nos nós que correspondem a cada endereço de encadeamento possível.

Preencha as lacunas escolhendo a alternativa CORRETA:

Pergunta 5

1,42 em 1,42 pontos

Dada as propriedades de estruturas de dados a seguir:

Estrutura 1: estrutura que mapeia a chave de busca diretamente para um endereço de memória (endereço base).

Estrutura 2: estrutura linear em que o primeiro elemento a entrar tem que ser o primeiro a sair.

Estrutura 3: estrutura linear em que as inserções e remoções ocorrem na mesma posição.

Assinale a alternativa que apresenta, em ordem, as estruturas para as quais se referem as definições.

Pergunta 6

1,42 em 1,42 pontos

Uma tabela recebe chaves do tipo string e armazena os dados internamente como um vetor. A função de espalhamento da tabela Hash utiliza o seguinte procedimento para mapear as strings em inteiros:

1 – Mapeamento de caracteres: os três primeiros caracteres são mapeados em inteiros da forma:

De a até f: mapeado para 1

De g até n: mapeado para 3

De o até s: mapeado para 5

De t até z: mapeado para 7

2 – Os inteiros associados a cada um dos três primeiros caracteres são multiplicados entre si.

3 – O resto da divisão por 11 é computado, dado que o vetor possui tamanho 11.

Dadas as seguintes strings: ULISSES, DANIELLE e LARISSA, aplicando a função de espalhamento apresentada, indique a alternativa correta que apresenta a string e a posição obtida.

Pergunta 7

1,42 em 1,42 pontos

Um método de busca bastante utilizado, conhecido como hash, baseia-se na utilização que mapeia chaves em endereços de memória, de modo que os dados associados a cada chave possam ser rapidamente localizados e lidos. Quando há conflitos de localização, algum algoritmo de separação é adotado.

Considere uma tabela hash armazenada em um arquivo no disco rígido. Supondo-se que a mesma possua uma função de *hash* razoavelmente protegida de conflitos, o número médio de acessos ao disco, necessários para localizar uma chave em um universo de N chaves, é mais próximo de: