

Recuperação P1 de Estrutura de Dados

PROFESSOR: __Renato de Pierri ____ CURSO: _Tecnologia em Análise e Desenvolvimento de Sistemas_
 Sigla _IED001_ TURMA: __A____ PERÍODO: MAT () NOT ()
 DISCIPLINA: _Estrutura de Dados____ DATA: ____ / ____ / ____
 NOME: _____ RA: _____

Argumente com fundamentando as questões da avaliação indicando porque a questão do gabarito está correta e porque as outras opções não podem ser consideradas corretas. Trabalho individual para imprimir e responder as questões à mão. Não será aceito trabalho impresso. Prazo de entrega é dia 13 de Abril de 2017 até as 22:00hs.

- 1)- Marque a alternativa que melhor descreve a definição do que é uma estrutura de dados homogênea:
- a)- É uma variável que permite o agrupamento de vários tipos de informações e seus valores podem ser misturados como: Inteiro, Float e Char e Struct tudo junto, por exemplo.
 - b)- É uma variável que não permite o agrupamento de um mesmo tipo de informação. Seus valores devem ser de tipos diferentes: Só pode utilizar os tipos Inteiro, Float e Char por exemplo.
 - c)- É uma variável que permite o agrupamento de um mesmo tipo de informação. Na linguagem C, isso é feito criando-se arrays de objetos.
 - d)- É uma variável que permite o agrupamento de um mesmo tipo de informação e seus valores devem ser do mesmo tipo: Inteiro, Float e Char por exemplo.**
 - e)- É uma variável que permite o agrupamento de um mesmo tipo de informação. Na linguagem C, isso é feito criando-se arrays de objetos que usam valores apenas do tipo Inteiro, Float e Char.

- 2)- Descartando a possibilidade do uso de funções de alocação dinâmica de memória, marque verdadeiro (V) ou falso (F) para as afirmações listadas abaixo:
- (F)- Array é um arranjo de elementos colocados em uma região segmentada de memória.
 - (V)- Todos elementos de um vetor possuem o mesmo nome e são referenciados por um subscrito.
 - (F)- O índice de um array só pode ser um número inteiro. Utilizar uma expressão no lugar do índice de um array causa erro de compilação.
 - (V)- A declaração "int dados[10];" corresponde a um array de nome "dados", com 10 posições, sendo que o índice do vetor vai de 0 até 9.
 - (F)- Na linguagem C, um array pode ser redimensionado em tempo de execução sem o uso de funções alocação dinâmica de memória.

Recuperação P1 de Estrutura de Dados

PROFESSOR: __Renato de Pierri ____ CURSO: _Tecnologia em Análise e Desenvolvimento de Sistemas_
NOME: _____ RA: _____

3)- Abaixo há duas inicializações de arrays. Comente o código descrevendo o que está sendo feito.

```
int valor[5] = {10,20,30,40,50};
```

```
//O array 'valor' de variável tipo inteiro está sendo inicializado. Este vetor tem 5 posições indo de 0 à 4 e está sendo inicializado com os //valores 10, 20, 30, 40 e 50
```

```
//=====
```

```
int semInicializador[] = {33, 44, 55};
```

```
//Declara e inicializa o array semInicializador como um array uni-dimensional com três membros, uma vez que o tamanho do array não //foi especificado e há três inicializadores (The C Programming Language – Brian W. Kernighan e Dennis Ritchie. 1988)
```

Considerando as inicializações acima, responda: As duas inicializações vão compilar? (SIM) (NÃO). Por quê?

4)- O que faz o trecho de código abaixo?

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int i;
    int valor[5] = {10,20,30,40,50};
    for (i = 0; i < 5; i++){
        printf("vetor valor[%i] = %i\n", i, valor[i]);
    }
    printf("\n");
    return 0;
}
```

- a)- A inicialização do vetor "valor" está errada.
- b)- Inicializa e exibe o conteúdo do vetor "valor".**
- c)- O printf só funciona com "%d" para inteiros.
- d)- Vai ficar em loop eterno.
- e)- Vai fazer a entrada de valores no vetor.

Avaliação de Estrutura de Dados

PROFESSOR: __Renato de Pierri ____ CURSO: _Tecnologia em Análise e Desenvolvimento de Sistemas_
NOME: _____ RA: _____

5)- Abaixo tem duas inicializações de array, ambas corretas e inicializando o array com o caractere '\0'. Relacione o código com o comentário

```
//(_A_)
char leitura[]={'\0','\0','\0','\0','\0','\0','\0','\0','\0','\0'};
//(_B_)
char invertido[10];
for (controle = 0; controle < 10;controle++){
    invertido[controle] = '\0';}
```

(A) Inicialização simples, funciona com pequenos arrays mas seu uso é discutível.

(B) Considerado uma boa prática de programação e é adequada para a inicialização de arrays de qualquer tamanho

6)- O código abaixo, quando executado carrega o vetor "matriz" com os caracteres "A", "B", "C", "D", "E" e "F". Assinale a alternativa que corresponda à constante eu devo alterar para que o código passe a chamar a função "lerLetra(0)":

a)- linha	b)- coluna	c)- LIN
d)- COL		e)- AUT

```
void desafio2(){
    //Definindo a constante LIN para linha e COL para coluna
    const LIN = 2;
    const COL = 3;
    const AUT = 1;
    //Declarando a matriz
    char matriz[LIN][COL];
    int linha, coluna, controle;
    //Pula uma linha e inicializa 'controle'
    printf("\n");
    controle = 0;
    //Populando a matriz
    for (linha = 0; linha < LIN; linha++){
        //Percorrendo as colunas da matriz
        for (coluna = 0; coluna < COL; coluna++){
            //Pega o valor de controle e coloca na matriz
            if(AUT == 1){
                matriz[linha][coluna] = 65 + controle;
            }else{
                matriz[linha][coluna] = lerLetra(0);
            }
            //Incrementa o valor de controle
            controle ++;
        }
    }
    // --- Continua o código ...
}
```

Avaliação de Estrutura de Dados

PROFESSOR: __Renato de Pierri ____ CURSO: _Tecnologia em Análise e Desenvolvimento de Sistemas_
 NOME: _____ RA: _____

<p>7)- Considerando o código abaixo, assinale a alternativa que corresponda ao conteúdo do vetor "invertido" ao término do processamento, imediatamente antes de executar o 'return'.</p>	<pre>int main() { size_t tam_Leitura; int controle = 0; int leitura[5]={1,2,3,4,5}; int invertido []={'\0','\0','\0','\0','\0'}; tam_Leitura = sizeof(leitura); tam_Leitura = (tam_Leitura/4)-1; for (controle = 0; controle <= 4; controle++){ invertido[-1*(controle - tam_Leitura)] = leitura[controle]; } return 0; }</pre>				
a)- 5, 4, 3, 2, 1	b)- 4, 3, 2, 1, 1	c)- 5, 5, 4, 3, 2	d)- 1, 2, 3, 4, 5	e)- 4, 4, 4, 4, 4	

<p>8)- Ainda considerando o trecho de código da questão 7, marque a alternativa que melhor informa quais linhas de código estão calculando a quantidade de elementos existentes no array "leitura":</p>	<p>a)- int invertido []={'\0','\0','\0','\0','\0'}; tam_Leitura = sizeof(leitura);</p> <p>b)- tam_Leitura = sizeof(leitura); tam_Leitura = (tam_Leitura/4)-1;</p> <p>c)- int leitura[5]={1,2,3,4,5}; tam_Leitura = sizeof(leitura);</p> <p>d)- int leitura[5]={1,2,3,4,5}; tam_Leitura = (tam_Leitura/4)-1;</p> <p>e)- tam_Leitura = (tam_Leitura/4)-1; invertido[-1*(controle - tam_Leitura)]</p>
---	---

Avaliação de Estrutura de Dados

PROFESSOR: __Renato de Pierri ____ CURSO: _Tecnologia em Análise e Desenvolvimento de Sistemas_
NOME: _____ RA: _____

9)- O trecho de código abaixo é de um algoritmo de ordenação com baixa eficiência. Analisando o código e comentários, assinale a alternativa que melhor corresponda ao nome do algoritmo de ordenação.

- a)- Buble Sort
- b)- Quick Sort
- c)- Shell Sort
- d)- Merge Sort
- e)- Insertion Sort

```
printf("\nIniciando a classificacao:\n");
for (controle = 0; controle < qtd_Elementos - 1; controle++){
    for (j = qtd_Elementos - 1; j > controle; --j){
        if(elemento[j] < elemento[j-1]){

            //Trocando os dados de lugar
            temp = elemento[j];
            elemento[j] = elemento[j-1];
            elemento[j-1] = temp;

            //Imprimindo o processo de classificacao
            printf("|");
            for (lista=0; lista < qtd_Elementos; lista++){
                printf(" %d |", elemento[lista]);
            }
            printf("\n");
        }
    }
}
printf("Fim da classificacao\n");
```

10)- O método de classificação Quick Sort, nasceu empregando a recursão. O trecho de código a seguir é a função Quick Sort implementada em sala de aula. Analisando o código, assinale a alternativa que mostra especificamente a chamada recursiva da função quickSort.

- a)- 10 e 11
- b)- 11 e 12**
- c)- 12 e 13
- d)- 09 e 10
- e)- 09 e 12

```
1 // Função quickSort é recursiva.
2 void quickSort(int esquerda, int direita){
3     int pivo, pontoParticao;
4
5     if(direita - esquerda <= 0){
6         return;
7     }
8     else{
9         pivo = intArray[direita];
10        pontoParticao = particao(esquerda, direita, pivo);
11        quickSort(esquerda, pontoParticao-1);
12        quickSort(pontoParticao+1, direita);
13    }
14 }
15 // Fim da função quickSort
16
```

Avaliação de Estrutura de Dados

PROFESSOR: __Renato de Pierri ____ CURSO: _Tecnologia em Análise e Desenvolvimento de Sistemas_

NOME: _____ RA: _____

11)- A respeito do Quick Sort, assinale Verdadeiro (V) ou Falso (F):

-)- Quick sort é um algoritmo de classificação altamente ineficiente.
-)- É baseado no particionamento de um array de dados em arrays menores
-)- O particionamento é baseado em um valor chamado pivô
-)- Uma matriz irá conter valores menores que o valor do pivô escolhido
-)- Outra matriz irá conter valores maior ou igual ao valor do pivô escolhido
-)- Ao particionar um array, o Quick sort faz uma chamada recursiva aos dois sub arrays resultantes.

12)- O trecho de algoritmo a seguir pertence a qual método de classificação?

- a)- Buble Sort
- b)- Merge Soft
- c)- Selection Sort
- d)- Quick Sort**
- e)- NDA

- Escolher o maior índice do array como 'pivô'.
 - Criar duas variáveis para apontar a 'Esquerda' e a 'Direita' da lista, excluindo o 'pivô'.
 - 'Esquerda' vai apontar para o índice baixo.
 - 'Direita' vai apontar para o índice alto.
 - Enquanto o valor da 'Esquerda' for menor que o do 'pivô', 'Esquerda' avança para lado direito.
 - Enquanto o valor da 'Direita' for maior que o do 'pivô', 'Direita' avança para lado esquerdo.
 - Se a condição dos passos 5 e 6 for falsa, trocar o valor da 'Esquerda' com o da 'Direita'.
- Se 'Esquerda' for \geq 'Direita', o ponto em que eles se encontram é o novo pivô.*



GOVERNO DO ESTADO
SÃO PAULO

Avaliação de Estrutura de Dados

PROFESSOR: __Renato de Pierri ____ CURSO: _Tecnologia em Análise e Desenvolvimento de Sistemas_
NOME: _____ RA: _____

13)- A linguagem C possui na biblioteca 'stdlib.h' a função 'qsort' que implementa o quick sort.

void qsort(void *base, size_t nitems, size_t size, int (*compar)(const void *, const void*))

Relacione a descrição em inglês desses parâmetros com a descrição equivalente em português.

- (A) base -- This is the pointer to the first element of the array to be sorted.
- (B) nitems -- This is the number of elements in the array pointed by base.
- (C) size -- This is the size in bytes of each element in the array.
- (D) compar -- This is the function that compares two elements.

- (B)- Este é o número de elementos no array apontados pela base
- (C)- Este é o tamanho em bytes de cada elemento no array
- (A)- É o ponteiro para o primeiro elemento do array a ser classificado
- (D)- Esta é a função que compara dois elementos
