

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA - IFMA
Curso: Sistemas de Informação
Algoritmos e Estrutura de Dados – Prof. Gentil Cutrim Serra Jr
Avaliação I - 16/março/2023

Aluno: _____

1) [1pt] [Adaptada - UFSM - 2022 - UFSM - Analista de Tecnologia da Informação]
Considere a seguinte função **recursiva** em Python:

```
def func(arg):  
    if arg == 0:  
        return 0  
    else:  
        return (arg % 2) + 10 * func(arg//2)
```

Equivalente em Linguagem C:

```
int func (int arg) {  
    if (arg == 0)  
        return 0;  
    else  
        return (arg % 2) + 10 * func(arg / 2);  
}
```

Qual o valor retornado pela função acima, quando recebe como parâmetro o número 5?

Resposta: 101

Para func (5): return 1 + 10 * func (2)

Para func (2): return 0 + 10 * func (1)

Para func (1): return 1 + 10 * func (0)

Para func (0): return 0

Para func (1): return 1 + 10 * func (0) → return 1 + 10*0 = return 1

Para func (2): return 0 + 10 * func (1) → return 0 + 10*1 = return 10

Para func (5): return 1 + 10 * func (2) → return 1 + 10*10 = **return 101**

2) [1pt] Analise as afirmativas referentes às estruturas de dados básicas:

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA - IFMA
Curso: Sistemas de Informação
Algoritmos e Estrutura de Dados – Prof. Gentil Cutrim Serra Jr
Avaliação I - 16/março/2023

- I – Uma lista linear é uma sequência de zero ou mais itens x_1, x_2, \dots, x_n , em que x_i é de um determinado tipo e n representa o tamanho da lista linear.
- II – Uma pilha é uma lista linear em que todas as inserções, retiradas e geralmente todos os acessos, são feitos em apenas um extremo da lista.
- III – Uma fila é uma lista linear em que todas as inserções são realizadas em um extremo da lista e todos os acessos e retiradas são realizados **no mesmo extremo da lista**.

Estão CORRETAS as afirmativas:

- a) **I e II, apenas.**
- b) I e III, apenas.
- c) II e III, apenas.
- d) I, II e III.

3) [1pt] [CESPE / CEBRASPE - 2023 - PO-AL - Perito Criminal - Especialidade: Análise de Sistemas] Julgue o próximo item, a respeito de tipos de dados elementares e estruturados. Em uma estrutura de dados tipo lista, o elemento eliminado do conjunto é o mais recentemente inserido. A lista implementa uma política em que **o último a entrar é o primeiro a sair**.

- () Certo
(**) Errado**

4) [1pt] Escreva uma **função de UTILIZAÇÃO** que remove a segunda metade dos elementos de uma TAD lista (não importa o tipo de alocação de memória).

```
int excluir_2ametade(tipo_lista *le);
```

```
int excluir_2ametade(tipo_lista *le) {  
    int i; elemento e;  
    int metade = tamanho(*le)/2;  
    for (i=1;i<=metade;i++) { // metade vezes  
        excluir_elemento(*le,metade,&e); // sempre deleta da posição metade  
    }  
}
```

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA - IFMA
Curso: Sistemas de Informação
Algoritmos e Estrutura de Dados – Prof. Gentil Cutrim Serra Jr
Avaliação I - 16/março/2023

5) [2pt] Escreva uma **função de implementação** que remove a segunda metade dos elementos de uma lista **alocada dinamicamente**, sem utilizar a função **excluir_elemento**. Utilize a função **free** para liberar espaços de memória.

```
int excluir_2ametade(tipo_lista *le);

int excluir_2ametade(tipo_lista *le) {
    ptr_nodo pnode_excluido, pnode_anterior;
    int metade = tamanho(le)/2;
    pnode_excluido = le->lista;
    for (int i=1; i<=metade; i++) { // posiciona na 2a metade da lista
        pnode_anterior = pnode_excluido;
        pnode_excluido = pnode_excluido->prox;
    }
    pnode_anterior->prox = NULL; // passa a ser o final da lista
    while (pnode_excluido!=NULL) { // deleta os nodos da 2a metade
        pnode_anterior = pnode_excluido;
        pnode_excluido = pnode_excluido->prox;
        le->tamanho--;
        free(pnode_anterior);
    }
}
```

6) [2pt] Dada uma lista sequencial **LstFornecedores**, onde cada elemento da lista é implementado como um registro que contém os campos: **codigo**, **nome**, **dtCadastro**, considere que essa lista possui vários elementos incluídos. Faça uma **rotina de utilização** para realizar uma cópia dessa lista para uma outra lista **Lst2023**, copiando apenas a primeira metade dos elementos e que tenham o ano de cadastro igual a 2023.

<pre>typedef struct { int codigo; char nome[40]; data dtCadastro; } tipo_fornec;</pre>		<pre>typedef struct { int dia; int mês; int ano; } data;</pre>
--	--	--

```
void copia_lista_camada_utilizacao () {
    elemento e;
    int metade = tamanho(LstFornecedores)/2;
    for (int i=1; i<=metade; i++) {
        obter_elemento(LstFornecedores, i, &e);
        if (e.dtCadastro.ano == 2023)
            incluir_elemento(&LstF2023, tamanho(LstF2023)+1, e);
    }
}
```

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA - IFMA
Curso: Sistemas de Informação
Algoritmos e Estrutura de Dados – Prof. Gentil Cutrim Serra Jr
Avaliação I - 16/março/2023

}

7) [2pt] Dada uma lista de clientes (codigo, nome, telefone), implementada como um TAD, deseja-se acrescentar um novo registro à lista. Faça uma **rotina de utilização** (considere as operações predefinidas para uma TAD Lista – alocação estática ou dinâmica) que pesquisa se o nome já está incluído na lista (strcmp), se já estiver informe para o usuário em qual posição está, caso contrário, inclua na última posição da lista.

```
int inclusao_registro_camada_utilizacao (tipo_lista lista,
elemento cliente) {
    int i, elemento e;
    for (i=1;i<=tamanho(lista); i++) {
        obter_elemento(lista, i, &e);
        if (strcmp(e.nome,cliente.nome) == 0) {
            printf("elemento já existe na posição %d da lista", i);
            return 0;
        }
    }
    incluir_elemento(&lista, tamanho(lista)+1, e);
    return 1;
}
```