Escola Superior de Tecnologia e Gestão de Viseu

Engenharia Informática

Exame (Ép. Recurso) de **Estruturas de Dados**

Ano lectivo 2022/2023 – 1º Ano/2º Semestre

19.07.2023 **Duração:** 120 Minutos Sem Consulta

Nome: _____ *N.º* | | | | | | |

Nota: Nesta prova todas as secções de código pedidas devem ser escritas na linguagem de programação C.

I (1.0 V cada pergunta)

1. Considere uma lista de inteiros L, como definida nas aulas. O seguinte código permite:

```
NO *p = L->inicio;
if (p) return;
while (!p)
{
  printf("Info= %d\n",p->info);
  p = p->prox;
};
```

☐ Mostrar todos os elementos da lista;

☐ Mostra somente o primeiro elemento da lista;

☐ Entra em ciclo infinito;

☐ Nenhuma das anteriores ou existem erros.

2. Considere uma lista de PESSOAS, Assuma que existem as funções Add (para inserir) e DestruirLista (que vai destruir a lista e todo o seu conteudo).

```
Lista *L1 = (Lista *)malloc(sizeof(Lista));
Lista *L2 = (Lista *)malloc(sizeof(Lista));
PESSOA *P = (PESSOA *)malloc(sizeof(PESSOA));
Add(L1, P);
Add(L2, P);
DestruirLista(L1); DestruirLista(L2);
```

☐ Cria duas Listas e cria duas PESSOAS;

☐ Cria duas Listas e cria duas PESSOAS, destruindo de seguida as listas;

☐ A(s) instruções têm erro de compilação;

☐ Podem acontecer resultados inesperados na execução do último DestruirLista.

3. . O código:

```
int *Q = (int *)malloc(sizeof(int));
int *L = (int *)malloc(sizeof(int));
*Q = 10;
*L = 20;
L = Q;
Q = L;
printf("L = [%d]; Q = [%d];\n", *L, *Q);
free (L);
```

Qual o resultado?

☐ L =[20]; Q = [10];

☐ L =[10]; Q = [20];

☐ L =[10]; Q = [10];

☐ Nenhuma das anteriores ou tem erros de **sintaxe**;

4. Considere uma árvore binária K ordenada (**decrescente**), onde foram inseridos os valores (nesta sequência) 50; 20; -10; 5; 6; 30; 10.

```
int Pert(NO *p, int X)
{ if (!p) return 0;
  if (p->info != X) return 1;
  if (X>p->info) return Pert(p->Dir,X);
  if (X<p->info) return Pert(p->Esq,X);
}
```

A seguinte chamada:

```
printf("Valor = %d",Pert(K->raiz, 10));
```

☐ Tem como output 1;

☐ Tem como output 0;

☐ Entra em ciclo infinito;

☐ Nenhuma das anteriores ou existem erros de sintaxe;

5. Considere o seguinte código (a estrutura Livro tem os campos **codigo**(int) e **preco**(float)):

```
Livro *VP;
VP = (Livro *)malloc(200*sizeof(Livro));
for (int i = 0; i<200; i++)
  VP[i].preco = i;
```

Pretende-se aumentar o preço do livro em 10%, qual a instrução correcta?

☐ for (i=0; i<200; i++) VP[i].preco *= 0.1;

☐ for (i=0; i<200; i++) VP[i]->preco *=1.1;

☐ for (i=0; i<200; i++) VP[i].preco -= 10/100;

☐ Nenhuma das anteriores ou existem erros de sintaxe;

6. Um restaurante pretende implementar um sistema na cozinha que gerir os pedidos (pratos) dos seus clientes. É normal que a primeira pessoa a pedir, seja a primeira a ser atendida!. Qual a estrutura que mais de adequa?

☐ Lista;

☐ Fila;

☐ Pilha

☐ Árvore Binária;

7. O código:

```
Caixa M = (Caixa *)malloc(sizeof(Caixa));
free (M);
```

☐ cria um novo ponteiro M alocando espaço para uma caixa e elimina-a;

☐ cria um novo ponteiro M para uma caixa e elimina-o;

☐ Tem erro(s) de compilação

☐ Nenhuma das anteriores;

8. Um algoritmo recursivo, pode ser implementado iterativamente. Qual a estrutura de dados que mais se adequa?

☐ Lista;

☐ Fila;

☐ Pilha

☐ Árvore Binária;

9. O seguinte código permite:

```
ListaGeral *L = CriarLista(); // Assumir que a função existe!
free(L);
```

☐ Criar uma Lista e depois destruir a lista;

☐ Criar uma Lista, mas a destruição não está correta;

☐ Não está correto o modo de criar a lista, mas está correto a sua destruição;

☐ Nenhuma das anteriores;

10. Considere a função "Func"com o código abaixo. Assumindo que a função é chamada tendo como parâmetro o seu N.º mecanográfico, qual o valor de retorno?

```
int Func(int Nmec)
{
  if (Nmec == 0) return 0;
  return 1 + Func(Nmec / 10);
};
```

☐ 5;

☐ 25;

☐ 31;

☐ Nenhuma das anteriores;

II (Responda somente a 3 perguntas)

Considere as estruturas de dados **usadas no seu trabalho prático!**

Implemente **três** das seguintes funções:

a) Considerando o trabalho prático, implemente a função para determinar qual a Caixa que tem **MENOS** Clientes em espera: CAIXA ***Get_Caixa_Menos_Clientes(Supermercado *S)**.

b) Verificar se uma dada pessoa está em espera em alguma caixa, dado o código de cliente; **int PesquisarPessoaCaixa(Supermercado *S, int Cod_Cliente)** ; Se existir o cliente **retorna 1**; caso contrário **retorna 0.** Podem assumir que o Supermercado tem uma lista(ou array) de caixas.

c) Implemente uma (ou mais) função(ões) para inverter a ordem de ordenação de uma árvore.
void InverterOrd(ArvBinaria *A)

d) Determinar o numero de nós da árvore binária. Terá de implementar o algoritmo que percorra toda a árvore. Assumir que a estrutura ABinaria, não tem o atributo/campo NELementos

int ContarNos (ABinaria *A)

e) Implemente o destruir de uma dada árvore binária.

void DestruirArvore(ABinaria *A)