

Prova 1

- 1) Considere os tipos de alocação de arranjos em C++ sendo executados dentro de uma função. Informe para cada um: (1) o tipo de vinculação do intervalo de índices (fixo ou não); (2) o tipo de vinculação de memória (automático/dinâmico/estático) e; (3) o local no qual a memória é alocada (pilha/heap). Informe também, (4) o tempo em que ocorrem os itens (1) e (3) (carga/compilação/execução).
 - a) `int b[10];`
 - b) `int *d = new int[15];`
`delete[] d;`
 - c) `vector e;`
`e.push_back(52);`
 - d) `static int a[10];`
 - e) `int *d = new int[n];`
`delete[] d;`
 - f) `int c[n];`
- 2) Explique e diferencie alocação manual, estática e automática de variáveis. Qual o tempo de vida de cada tipo de alocação?
- 3) Explique e exemplifique os seguintes tipos de agregados homogêneos e heterogêneos comuns à linguagens imperativas: Registro, União, Tupla e Lista. Os exemplos podem ser fornecidos em sua linguagem de preferência.
- 4) Considerando Ponteiros e Referências em Heap, explique e apresente ao menos uma vantagem e uma desvantagem de termos um Coletor de Lixo (Java/C#/Python) vs Alocação Manual (C/C++)
- 5) Considerando expressões aritméticas em linguagens de programação, apresente um exemplo para cada notação (com 5 operandos): pré-fixa, in-fixa e pós-fixa. Em qual(is) é necessário usar parentização para forçar a ordem das operações? Em qual(is) ocorre associatividade de operadores?
- 6) Explique e exemplifique as conversões de estreitamento e ampliação. Existe perda de valores quando essas conversões são aplicadas?
- 7) Considerando expressões em linguagens de programação imperativas, explique e exemplifique operadores relacionais, booleanos e binários. Utilize a linguagem de sua preferência.
- 8) Explique e exemplifique os seguintes métodos de passagem de parâmetros: in, out, inout. Utilize a linguagem de sua preferência.
- 9) Considerando a linguagem C, qual o resultado impresso na última linha do seguinte código? Reescreva o código usando delimitadores de blocos { e }.

```
int a = 1, b = 3, c = 99;
if (a == 0)
    if (b == 0)
        c = 0;
else
    if (b > 2)
        c = 1;
printf("%d", c);
```