

Nome: \_\_\_\_\_ Código: \_\_\_\_\_

**Notas:**

- Responda às questões seguintes, indicando a opção correta (em maiúsculas)
- Cada resposta errada vale -15% da cotação da pergunta

1. Considere a função `printN` apresentada:

```
template <class T> void printN(T element, int n = 1) {  
    while (n--)  
        cout << element;  
};
```

- A. `T` tem de ser obrigatoriamente um dos tipos de dados pré-definidos em C++
- B. `T` tem de incluir apenas a implementação do operador `<<`
- C. `T` tem de incluir a implementação operador de igualdade e do operador `<<`
- D. `T` tem de incluir a implementação do construtor de cópia e do operador `<<`
- E. Nenhuma das possibilidades anteriores

Resposta:     D    

2. Considere a definição das classes `Box` e `OneBox`:

```
template <class T> class Box{};  
template <class T> class OneBox: public Box<T> {};
```

Que atribuições são corretas?

- I. `Box<string> b1 = OneBox<string>();`
- II. `OneBox<string> b2 = Box<string>();`
- III. `Box<string> b3 = OneBox<int>();`

- A. I, II e III
- B. II apenas
- C. I apenas
- D. I e II apenas
- E. Nenhuma das possibilidades anteriores

Resposta:     C    

3. Considere o código seguinte em C++ e indique quando é escrita a mensagem "exception found".

```
try {  
    // code  
} catch (ExceptA e) { cout << "exception found"; }
```

- A. Quando em `code` é lançada uma exceção da classe `ExceptA` apenas
- B. Quando em `code` é lançada uma exceção da classe `ExceptA` ou classes derivadas de `ExceptA`
- C. Quando em `code` é lançada uma exceção da classe `ExceptA` apenas e esta inclui o operador `<<`
- D. Nunca é escrita a mensagem "exception found"
- E. Nenhuma das possibilidades anteriores

Resposta:     B

4. Considere a função `maximum`, que determina o maior valor de um vetor passado como argumento.

```
template <class T> T maximum(const vector<T> &v)
{
    if ( v.size() == 0 ) throw EmptyVector();
    T max=v[0];
    for (int i=1; i<v.size(); i++)
        if ( v[i] > max ) max=v[i];
    return max;
}
```

- A. A função lança uma exceção sempre que o vetor `v` está vazio
- B. A função lança uma exceção se o operador `>` não está definido no tipo de dados `T`
- C. A função lança uma exceção apenas quando o vector `v` está vazio e `T` é um tipo de dados pré-definido em C++
- D. A função lança uma exceção sempre que `T` não é um tipo de dados pré-definido em C++
- E. Nenhuma das possibilidades anteriores

Resposta:     A    

5. Suponha um algoritmo de complexidade temporal  $O(N^{1/2})$ . Para  $N=100$ , o tempo de execução do algoritmo é de 1 minuto. Qual o tempo aproximado de execução do algoritmo para  $N=1000$ ?

- A. Cerca de 1 minuto
- B. Cerca de 3 minutos
- C. Cerca de 10 minutos
- D. Cerca de 30 minutos
- E. Nenhuma das possibilidades anteriores

Resposta:     B    

6. Indique a complexidade temporal da função `methodX`:

```
int methodX(int n) {
    int s=0;
    while ( n > 1 ) {
        n = n/2;
        s++;
    }
    return s;
};
```

- A.  $O(n \log n)$
- B.  $O(n)$
- C.  $O(\log n)$
- D.  $O(n^2)$
- E. Nenhuma das possibilidades anteriores

Resposta:     C

Nome: \_\_\_\_\_ Código: \_\_\_\_\_

7. Suponha que estava a verificar a implementação do algoritmo `quickSort` que efetua a ordenação de um vetor em ordem ascendente. Após o término do primeiro passo, o conteúdo do vetor é o seguinte: [3, 9, 1, 8, 17, 24, 22, 20]. Qual o pivot usado neste passo?
- A. 17
  - B. 8
  - C. 24
  - D. 1
  - E. Nenhuma das possibilidades anteriores

Resposta:     A    

8. O algoritmo de ordenação `insertionSort`: mantém os primeiros  $k$  elementos do vetor ordenados; para o próximo elemento ( $k+1$ ), procura a posição correta para o inserir entre os primeiros  $k$  de modo a manter a ordenação. Neste passo de procurar a posição onde colocar o elemento  $k+1$ , é usado um método de pesquisa sequencial. Suponha que o algoritmo `insertionSort2` usa, neste passo, um método de pesquisa binária. A complexidade temporal do algoritmo `insertionSort2` é:
- A.  $O(n \log n)$
  - B.  $O(n)$
  - C.  $O(\log n)$
  - D.  $O(n^2)$
  - E. Nenhuma das possibilidades anteriores

Resposta:     D    

9. Pretende-se pesquisar um elemento  $x$  num vetor que se sabe estar ordenado. É correto afirmar (no caso médio) que a pesquisa sequencial:
- A. Possui melhor desempenho temporal que a pesquisa binária, no caso de o elemento  $x$  não existir
  - B. Possui melhor desempenho temporal que a pesquisa binária, no caso do elemento  $x$  existir
  - C. Possui desempenho temporal semelhante à pesquisa binária, no caso de o elemento  $x$  não existir
  - D. Possui pior desempenho temporal que a pesquisa binária, quer o elemento  $x$  exista ou não
  - E. Nenhuma das possibilidades anteriores

Resposta:     D    

10. Na linguagem C++, um membro-dado estático (`static`):
- A. Só pode ser acedido por membros-função estáticos
  - B. Só pode ser acedido depois de ser criado o primeiro objeto da classe
  - C. Pode ter um valor mesmo que não existam objetos da classe
  - D. Não pode ser alterado
  - E. Nenhuma das possibilidades anteriores

Resposta:     C