

**1) Lista de exercícios:** Resolva os exercícios abaixo como se pede.

- a) Escreva um programa em C++ para calcular o n-ésimo elemento da série de Fibonacci. O programa deve possuir um arquivo principal que solicita o índice n do elemento ao usuário e depois exibe na tela o resultado. Para isso, é necessária uma função para calcular e retornar o n-ésimo elemento da série, tendo como entrada o índice. O protótipo dessa função deve ser disponibilizado no arquivo `fibonacci.h` e o código fonte da função, no arquivo `fibonacci.cpp`. Não há o uso de classes.
- b) Repetir o exemplo anterior criando a classe `Fibonacci`. O método `setFibonacci`, entretanto, recebe o índice da série, mas não retorna o resultado. Assim, é necessária a criação de um método de acesso `private` que calcule o elemento da série, chamado `computeFibonacci`, invocado pelo próprio método `setFibonacci`. O resultado é armazenado como um atributo privado da classe, que só é acessado por meio do método `getFibonacci`.
- c) Escreva uma classe `Agenda` para receber e armazenar três nomes distintos. Os três nomes devem ser passados pelo usuário através da execução do método `setNames`, que interage com o usuário solicitando cada um dos nomes. Os três nomes devem ser armazenados em um atributo privado do tipo array de strings. Antes de inserir os nomes no array, um método para validação deve ser usado. Essa validação assegura que cada um dos nomes não tenha mais do que 10 caracteres. Caso possua, uma mensagem deve ser exibida ao usuário e o nome deve ser truncado para caber no limite máximo definido. A classe `Agenda` ainda deve oferecer um método público para que os usuários possam visualizar todos os nomes inseridos. Use os métodos `substr (string substr (size_t pos = 0, size_t n = npos) const;)` e `length (int length ())` definidas na biblioteca `string`.

## == Respostas da Lista de Exercícios

---

1)

a)

```
/* *****
Programa Principal ***** */

#include <iostream>
#include "fibonacci.h"

/* Programa do Laboratório 1:
   Fibonacci sem classes.
   Autor: Miguel Campista */

using namespace std;

int main () {
    int idx;

    cout << "Entre com o índice da série de Fibonacci: ";
    cin >> idx;

    cout << "\n\nO número eh: " << computeFibonacci (idx) << endl;

    return 0;
}

/* *****
Arquivo fibonacci-sc.h ***** */

int computeFibonacci (int);

/* *****
Arquivo fibonacc-sc.cpp ***** */

/* *****
Arquivo fibonacci.cpp ***** */

int computeFibonacci (int idx) {
    // Índice não positivos devem ser tratados na função principal

    if (idx == 0)
        return 0;
    else if (idx == 1)
        return 1;
    else
        return computeFibonacci (idx - 1) + computeFibonacci (idx - 2);
}

/* ***** */
```

b)

```
/* *****
Programa Principal ***** */

#include <iostream>
#include "fibonacci.h"

/* Programa do Laboratório 1:
   Fibonacci com classes.
   Autor: Miguel Campista */

using namespace std;

int main () {
    // Objeto da classe Fibonacci
    Fibonacci fibonacci;
    int idx;
```

```

    cout << "Entre com o índice da série de Fibonacci: ";
    cin >> idx;

    fibonacci.setFibonacci (idx);

    cout << "\n\nO número eh: " << fibonacci.getFibonacci () << endl;

    return 0;
}

/*****
/***** Arquivo fibonacci.h *****/

class Fibonacci {
public:
    void setFibonacci (int);
    int getFibonacci ();

private:
    int computeFibonacci (int);
    int resultado;
};

/*****
/***** Arquivo fibonacci.cpp *****/

#include "fibonacci.h"

void Fibonacci::setFibonacci (int idx) {
    resultado = computeFibonacci (idx);
}

int Fibonacci::getFibonacci () { return resultado; }

int Fibonacci::computeFibonacci (int idx) {

    // Índice não positivos devem ser tratados na função principal

    if (idx == 0)
        return 0;
    else if (idx == 1)
        return 1;
    else
        return computeFibonacci (idx - 1) + computeFibonacci (idx - 2);
}

/*****/

```

c)

```

/*****/
/*****/ Programa Principal *****/

#include <iostream>
#include <string>
#include "agenda.h"

/* Programa do Laboratório 1:
   Agenda para 3 nomes.
   Autor: Miguel Campista */

using namespace std;

int main () {
    // Objeto da classe Agenda
    Agenda agenda;

    agenda.setNames ();
    agenda.getNames ();

    return 0;
}

/*****/
/*****/ Arquivo agenda.h *****/

```

```

#include <iostream>
#include <string>

using namespace std;

class Agenda {
public:
    void setNames ();
    void getNames ();

private:
    string names [3];
    string checkName (string);
};

/*****
*****/
/*****/
Arquivo agenda.cpp *****/
*****/

#include "agenda.h"

void Agenda::setNames () {
    string name;

    for (int i = 0; i < 3; i++) {
        cout << "Entre com o nome " << (i + 1) << ": " << endl;
        getline (cin, name);
        names [i] = checkName (name);
    }
}

void Agenda::getNames () {
    cout << "\nOs nomes na agenda são: " << endl;

    for (int i = 0; i < 3; i++)
        cout << names [i] << endl;
}

string Agenda::checkName (string name) {
    if (name.length () > 10) {
        name = name.substr (0, 10);
        cout << "Nome truncado para: " << name << endl;
    }
    return name;
}

/*****/
*****/

```