

# exercícios cinco

1 - Implemente a seguinte função para retornar o equivalente em Fahrenheit de uma temperatura Celsius.

```
//Atividade 1
//Jamilly Ferreira Rodrigues

#include <iostream>
#include <iomanip>
using namespace std;

// Função para converter Celsius para Fahrenheit
float cel(float ius) {
    return (ius* 9.0 / 5.0) + 32.0;
}

int main() {
    cout << "Celsius para Fahrenheit" << endl;

    cout << setw(10) << "Celsius" << setw(15) << "Fahrenheit" << endl;

    for (int ius= 0; ius<= 100; ius += 10) {
        float fahrenheit = cel(ius);

        cout << setw(10) << ius << setw(15) << fixed << setprecision(1) << fahrenheit << endl;
    }

    return 0;
}
```

2 - Um número inteiro maior que 1 é considerado primo se for divisível por apenas 1 e ele próprio. Por exemplo, 2, 3, 5 e 7 são números primos, mas 4, 6, 8 e 9 não são.

```
//Atividade 2 A
//Jamilly Ferreira Rodrigues

#include <iostream>
using namespace std;

bool p(int j) {
    if (j <= 1) {
        return false;
    }

    for (int k = 2; k <= j / 2; k++) {
        if (j % k == 0) {
            return false;
        }
    }

    return true;
}

int main() {
    int d;

    cout<< "fale um valor: " ;
    cin>> d;

    if (p(d))
        cout<< "Numero primo";
    else
        cout <<"Numero nao e primo";
}
```

b) Use esta função em um código que determina e imprime todos os números primos entre 2 e 1.000.

```
//Atividade 2 b
//Jamilly Ferreira Rodrigues

#include <iostream>
using namespace std;

bool p(int j) {
    if (j <= 1) {
        return false;
    }

    for (int k = 2; k <= j / 2; k++) {
        if (j % k == 0) {
            return false;
        }
    }

    return true;
}

int main() {
    cout << "Numeros primos entre 2 e 1.000:" << endl;

    for (int i = 2; i <= 1000; i++) {
        if (p(i)) {
            cout << i << " ";
        }
    }

    cout << endl;

    return 0;
}
```

c) Inicialmente, você pode pensar que  $n / 2$  é o limite superior para o qual você deve testar se um número é primo, mas você precisa ir apenas até a raiz quadrada de  $n$ .

```
//Atividade 2 c
//Jamilly Ferreira Rodrigues

#include <iostream>
#include <cmath>
using namespace std;

bool p(int j) {
    if (j <= 1) {
        return false;
    }

    for (int k = 2; k <= sqrt(j); k++) {
        if (j % k == 0) {
            return false;
        }
    }

    return true;
}

int main(){
    int f;

    cout<< "fale um valor: ";
    cin >> f;

    if(p(f))
        cout<<" numero primo";
    else
        cout<< "nao e primo";

    return 0;
}
```

### 3 - Um número inteiro é considerado um número perfeito se a soma de seus fatores...

```
//Jamilly Ferreira
// atividade 3
#include <iostream>
using namespace std;

int main() {
    int somaDiv;

    for (int n = 1; n <= 1000 ;++n) {
        somaDiv = 0;

        for (int j = 1; j < n; ++j) {
            if (n % j == 0) {
                somaDiv += j;
            }
        }

        if (somaDiv == n && n != 1) {
            cout << "O numero " << n << " e PERFEITO." << endl;
        } else {
            cout << "O numero " << n << " nao e PERFEITO." << endl;
        }
    }

    return 0;
}
```

#### 4 - Os computadores estão desempenhando um papel crescente na educação.

```
//Jamilly Ferreira
// atividade 4

#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

int main() {
    srand(time(0));
    int n1, n2 ;
    int r;

    while (true) {

        n1 = rand() % 10;
        n2 = rand() % 10;

        cout << n1 << " vezes " << n2 << "? ";
        cin >> r;

        if (r == n1 * n2) {
            cout << "Muito bom!" << endl;
        } else {
            cout << "Nao. Por favor, tente novamente." << endl;
            continue;
        }
    }

    return 0;
}
```

5 - Escreva um código que reproduza o jogo de “adivinhar o número” da seguinte forma:

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

int main() {
    srand(time(0));

    char j = 's';
    int num, p;

    while (j == 's') {
        num = (rand() % 1000) + 1;

        cout << "adivinhar o meu numero entre 1 e 1000." << endl;
        bool adivinhou = false;

        while (!adivinhou) {
            cin >> p;

            if (p < num) {
                cout << "Muito baixo. Tente novamente." << endl;
            } else if (p > num) {
                cout << "Muito alto. Tente novamente." << endl;
            } else {
                cout << "Excelente! Você adivinhou o numero!" << endl;
                cout << "Você gostaria de jogar novamente (s ou n)? ";
                adivinhou = true;
            }
        }

        cin >> j;
    }

    return 0;
}
```

6 - (Torres de Hanói) Todo programador iniciante deve lidar com certos problemas clássicos.

```
#include <iostream>
using namespace std;

void moveDisk(int j, int a) {
    cout << j << " ? " << a << endl;
}

void moveTower(int n, int j, int a, int t) {
    if (n == 1) {
        moveDisk(j, a);
    } else {
        moveTower(n - 1, j, t, a);

        moveDisk(j, a);

        moveTower(n - 1, t, a, j);
    }
}

int main() {
    int num = 3;
    int j = 1;
    int a = 3;
    int t = 2;

    cout << "Torres de Hanoi com " << num << " discos:" << endl;

    moveTower(num, j, a, t);

    return 0;
}
```