



Prazo: -

LAB

AULA: SEG TER QUA QUI SEX HORÁRIO: 1,2 3,4

PROFESSOR: Daniel Caetano

DISCIPLINA: CCE0067 – Lógica de Programação

CURSO: _____

R.A.: _____

NOME: **GABARITO**

INSTRUÇÕES

- a) Preencha corretamente o **CURSO**, o **R.A.** e seu **NOME** e, se necessário, o **DIA DA SEMANA/HORÁRIO** da aula;
- b) **NÃO** responda as questões na parte da frente da capa, mas use o verso se desejar;
- c) Se o trabalho for composto por mais de uma folha, elas devem ser **grampeadas**, com a capa na frente;
- d) **NÃO** serão aceitos trabalhos após o prazo, fique atento;
- e) Trabalhos de laboratório devem ter um **visto do professor** na capa para serem aceitos.

QUESTÕES - AULA 14

EXERCÍCIOS QUE DEVEM SER ENTREGUES: APENAS -

14.1) Imprima a soma dos números de 1 a 300.

```
#include <iostream>
using namespace std;
main()
{
    // Declara contador e soma total
    int C, SOMA;

    // Inicialmente, a soma vale zero
    SOMA = 0;
    // Faz contador variar de 1 a 300
    for (C = 1; C <= 300; C = C + 1)
    {
        // E acrescenta cada valor do contador C em SOMA
        SOMA = SOMA + C;
    }
    // Imprime a soma total
    cout << "Soma: " << SOMA << endl;
}
```

14.2) Imprima a soma dos números divisíveis por 11 que ocorrem entre 1 a 3000.

```
#include <iostream>
using namespace std;
main()
{
    // Declara contador e soma total
    int C, SOMA;

    // Inicialmente, a soma vale zero
    SOMA = 0;
    // Faz contador variar de 1 a 300
    for (C = 1; C <= 3000; C = C + 1)
    {
        if (C%11==0) SOMA = SOMA + C;
    }
    // Imprime resultado
    cout << "Soma: " << SOMA << endl;
}
```

14.3) A multiplicação de dois valores inteiros positivos pode ser calculada como uma série de somas. Por exemplo:

$2 \cdot x$ pode ser escrito como $x+x$

$7 \cdot x$ pode ser escrito como $x+x+x+x+x+x+x$

Observe a regra e faça um programa que receba dois valores inteiros positivos, x e y , e calcule, usando estrutura de repetição, o valor de $x \cdot y$. NOTA: NÃO use o operador de multiplicação $*$!

```
#include <iostream>
using namespace std;
main() {

    // Declara variáveis inteiras
    int X, Y, SOMA, C;

    // Apresenta função do programa
    cout << "Calcula Multiplicação" << endl;

    // Recebe valores
    cout << "Digite multiplicando: ";
    cin >> X;
    cout << "Digite multiplicador: ";
    cin >> Y;

    // Soma inicial vale zero
    SOMA = 0;
    // Repete Y vezes
    for (C = 0; C < Y; C = C + 1) {
        // A soma de X
        SOMA = SOMA + X;
    }
    // Imprime resultado total da soma
    cout << SOMA << endl;
}
```

14.4) A potenciação com expoentes inteiros positivos pode ser calculada como uma série de multiplicações. Por exemplo:

x^2 pode ser escrito como $x*x$
 x^7 pode ser escrito como $x*x*x*x*x*x*x$

Observe a regra e faça um programa que receba um valor para a base (na variável x) e um valor inteiro positivo para o expoente (na variável y) e calcule, usando estrutura de repetição, o valor de xy. NOTA: NÃO use a função pow!

```
#include <iostream>
using namespace std;
main() {

    // Declara variáveis inteiras
    int X, Y, MULT, C;

    // Apresenta função do programa
    cout << "Calcula Potenciacao" << endl;

    // Recebe valores
    cout << "Digite a base: ";
    cin >> X;
    cout << "Digite o expoente: ";
    cin >> Y;

    // Soma inicial vale zero
    MULT = 1;
    // Repete Y vezes
    for (C = 0; C < Y; C = C + 1) {
        // A multiplicação por X
        MULT = MULT * X;
    }
    // Imprime resultado total das multiplicações
    cout << MULT << endl;
}
```

14.5) Imprima os fatoriais dos números 1 a 10. O fatorial (!) se calcula pela seguinte fórmula: $N! = N*(N-1)!$

Ex: 1! = 1
 2! = 2*1 = 2
 3! = 3*2*1 = 6
 4! = 4*3*2*1 = 24

...

```
#include <iostream>
using namespace std;
main()
{
    // Declara contador/multiplicador
    int C, MULT;
    // Inicialmente começa em 1
    MULT = 1;
    // Para cada valor de 1 a 10..
    for (C = 1; C <= 10; C=C+1)
    {
        // Multiplica o valor anterior pelo contador
        MULT = MULT * C;
        // E imprime esse resultado como sendo C!
        cout << C << "! = " << MULT << endl;
    }
}
```

14.6) Se quisermos fazer o fatorial de 1 a 20, o que acontece com os resultados? Altere o programa para usar variáveis do tipo "long long" ao invés de "int" e veja o que acontece com o resultado numérico. Você consegue explicar o que provavelmente muda no programa ao trocar o tipo da variável de "int" para "long long".

```
#include <iostream>
using namespace std;
main()
{
    // Declara contador/multiplicador
    long long C, MULT;
    // Inicialmente começa em 1
    MULT = 1;
    // Para cada valor de 1 a 10..
    for (C = 1; C <= 10; C=C+1)
    {
        // Multiplica o valor anterior pelo contador
        MULT = MULT * C;
        // E imprime esse resultado como sendo C!
        cout << C << "! = " << MULT << endl;
    }
}
```

Os resultados ficam corretos em um maior número de resultados. O que muda é que com o tipo de variável "long long" o computador reserva mais bits para cada número, permitindo que valores maiores possam ser armazenados nas variáveis.