



Prática 1 - INF 110 - Programação I - 2020

1 Introdução

Esta prática está supondo que o aluno utilizará o sistema operacional Linux durante as aulas. Usuários de Windows, por exemplo, podem utilizar o Cygwin (o terminal do Cygwin) como alternativa (combinada com algum editor de texto com *syntax highlighting*, como o *Sublime*).

1. Nessa prática você irá criar programas simples em C++. Para tal, abra um editor de texto (*gedit*, *leafpad*, *Sublime*, etc) e salve o arquivo (vazio, sem texto algum) com o nome *programa1.cpp*; a extensão *.cpp* indica que o arquivo contém código na linguagem C++ (C Plus Plus). O *gedit* ou *leafpad* são encontrados no menu que aparece ao clicarmos no botão inferior esquerdo da área de trabalho do Ubuntu.

Dica: salve o arquivo na pasta *documentos*, que fica dentro da pasta *aluno*.

2. Digite então o seguinte código C++:

```
//Programa1: Ola Mundo!  
//Imprime mensagem na tela  
#include <iostream>  
  
int main ()  
{  
    std::cout << "Ola Mundo! " << std::endl;  
    return 0;  
}
```

Salve o arquivo novamente.

3. Abra o aplicativo *Terminal* (dependendo da sua versão do Ubuntu o aplicativo pode ter o nome *LXTerminal*). O Terminal também é encontrado no menu que aparece ao clicarmos no botão inferior esquerdo da área de trabalho.
4. No Terminal, entre na pasta que contém o arquivo *programa1.cpp* utilizando o comando *cd <nome da pasta>*, onde *<nome da pasta>* deve ser substituído pelo nome da pasta onde o arquivo *programa1.cpp* foi salvo (ex.: *cd documentos/alunos*).

5. Digite o comando `g++ programa1.cpp`. Se o compilador `g++` não exibir nenhuma mensagem, seu programa está pronto para ser executado. O compilador `g++` terá criado um arquivo `a.out` com o código binário do seu programa. Se aparecer alguma mensagem de erro, corrija o erro no editor de texto, salve o arquivo e compile o código novamente.
6. Digite `./a.out` no Terminal para rodar o programa compilado

Dicas para os exercícios.

- Salvar o arquivo fonte antes de compilar, caso contrário a versão antiga do programa será compilada.
- Compilar sempre antes de testar o programa, caso contrário a versão antiga que será testada.
- O comando

```
g++ programa2.cpp -o programa2
```

cria um executável chamado `programa2`. Assim, pode-se executar o programa com o comando `./programa2`.

- No Terminal, use a seta para cima no teclado para exibir/executar os últimos comandos sem ter que digitá-los novamente.

2 Exercícios

1. Digite o código abaixo em um novo arquivo e teste-o com valores distintos.

```
//Programa 2: calculo da nota final de um candidato a monitoria
#include <iostream>
int main()
{
    int entrevista, curriculo, prova; // notas do candidato
    int total;                        // total de pontos

    //Leitura dos dados
    std::cout << "Escreva a nota da entrevista: ";
    std::cin >> entrevista;
    std::cout << "Escreva a nota do curriculo: ";
    std::cin >> curriculo;
    std::cout << "Escreva a nota da prova: ";
    std::cin >> prova;

    //Cálculo do resultado
    total = entrevista + curriculo + prova;

    std::cout << "Nota final do candidato: " << total;

    return 0;
}
```

- (a) Note que o resultado sai “colado” ao *prompt* de novo comando. Para resolver isso, acrescente

```
<< std::endl
```

depois de imprimir o valor total, como mostrado em sala de aula.

- (b) Coloque `//` antes do comando abaixo e verifique o que acontece.

```
std::cin >> entrevista;
```

- (c) Coloque `//` antes do comando abaixo e verifique o que acontece.

```
total = entrevista + curriculo + prova;
```

- (d) Suponha que a nota final não seja o total mas a média das três notas. Acrescente uma variável *media*, que receberá o valor total dividido por 3. Escreva o total e a média.

- (e) Para as notas 7, 8 e 8, a média deveria ser 7,6666; veremos como solucionar esse problema na próxima aula.

2. Digite o código abaixo em um novo arquivo e teste-o com valores distintos.

```
//Programa 3: caixa de restaurante
#include <iostream>
int main()
{
    int comida,bebida,sobremesa; // consumo
    int valordaconta, valorpago, troco; // outros valores

    std::cout << "Escreva o valor consumido em comida, bebida e sobremesa ";
    std::cin >> comida >> bebida >> sobremesa; //Lê os dados
    std::cout << "Escreva o valor pago pelo cliente: ";
    std::cin >> valorpago;

    valordaconta = comida + bebida + sobremesa; //Realiza a soma do valor da conta
    troco = valorpago - valordaconta;

    std::cout << "\n\nRESTAURANTE PEGUE E PAGUE\n";
    std::cout << "Consumo: " << comida << " + " << bebida << " + " << sobremesa << " = ";
    std::cout << valordaconta << "\n" ;
    std::cout << "Valor Pago: " << valorpago << "\n" ;
    std::cout << "Troco: " << troco << "\n" ;
    std::cout << "Volte sempre!\n";
    return 0;
}
```

`\n` é uma sequência de escape. O efeito é uma mudança de linha, assim como *endl*, mas pode ser colocado junto com o texto, dentro das aspas (e possui outras vantagens...).

- (a) Experimente digitar apenas um valor e pressionar a tecla `Enter`. O que acontece?

- (b) Substitua o primeiro comando `std::cin` por três, uma para cada variável, como mostrado abaixo. Há alguma diferença na execução?

```
std::cin >> comida;  
std::cin >> bebida;  
std::cin >> sobremesa;
```

- (c) Experimente entrar os dados 15, 5 e 3 para os valores consumidos e 18 para o valor pago. Qual o problema? Veremos como solucionar o problema na próxima aula.
3. Faça um programa para ler as dimensões de uma caixa (largura, altura e comprimento), e escrever seu volume.
4. Faça um programa para ler o salário base de um funcionário, o valor da hora extra, e o número de horas extras que ele trabalhou. O programa deve também calcular e imprimir o salário total do funcionário.
- Exemplo:** *se o salário base é 1000, o valor da hora extra é 50 e ele trabalhou 3 horas extras, seu salário total deverá ser 1150.*