

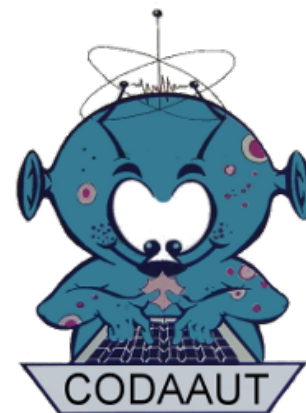
Programação



UFOP



INSTITUTO FEDERAL
MINAS GERAIS
Campus Ouro Preto



Programas Híbridos C/C++

Programas Híbridos C/C++

- A linguagem C++ foi criada a partir da linguagem C, mantendo toda a sintaxe desta última
 - Logo, C é um subconjunto da linguagem C++.
- C++ é uma linguagem orientada a objetos, com muitas vantagens sobre C, que é estruturada
 - Por exemplo a STL;
 - No entanto, C não caiu em desuso.

Programas Híbridos C/C++

- É possível criar programas híbridos C/C++
 - Por exemplo, criar um arquivo **.cpp** cujo conteúdo seja um programa em C puro.
- Graças ao C++, é possível criar programas C que:
 - Usem vetores realmente dinâmicos sem *calloc* e *malloc*;
 - Usem pilhas, listas, filas, conjuntos, multiconjuntos, mapas, multimapas e etc já codificados;
 - Tratem *strings* com mais naturalidade.
- Os programas podem ser estruturados e ainda assim utilizar objetos pré-existentes.

Programas Híbridos C/C++

- Para criar um programa híbrido, basta criar um código-fonte C++ e programar misturando C e C++
 - Compilação normal.
- Não é uma prática tão comum
 - Utilizada basicamente em competições de programação...

Esqueleto em C++

```
#include <iostream> // biblioteca para leitura C++
#include <cstdio> // biblioteca para leitura C
using namespace std;

// a função main inicia a execução do programa
int main()
{
    return 0; // indica que o programa terminou com sucesso
} // fim da função main
```

Entrada e Saída

Entrada e Saída

- Em todos os problemas que veremos, os dados são lidos da entrada padrão e escritos na saída padrão
- Nada de abrir arquivos ou chamadas de sistema.

Entrada e Saída

- Existem dois aspectos importantes sobre a leitura de dados
 - O quê deve ser lido?
 - Até quando ler?
- O primeiro diz respeito à formatação dos valores
 - número inteiro, número real, caractere, *string*, linha inteira...
- O segundo diz respeito ao término da entrada
 - Número fixo de valores, valor especial ou EOF.
- O enunciado do problema deve ser lido atentamente
 - A codificação da leitura dos dados não deve ser um processo lento.

Entrada e Saída

- Em nossos programas híbridos, utilizaremos as instruções/fluxos a seguir:
 - *scanf*: Lê dados de acordo com a formatação indicada;
 - *cin*: Lê dados de acordo com o tipo da variável;
 - *gets*: Lê uma linha excluindo o '\n', descarta o '\n' e coloca '\0' no final.

Números Inteiros

```
scanf("%d", &i);  
cin >> i;
```

- *i* deve ser uma variável do tipo *int*;
- No *scanf*:
 - %2d leria apenas um número de no máximo dois dígitos;
 - Para descartar um valor, usamos *
 - e. g., o código abaixo lê um número, ignora um caractere e lê outro número:

```
scanf("%d%*c%d", &i1, &i2);
```

Caracteres

- ❑ `scanf("%c", &c);`
 - ❑ Lê o próximo caractere.
- ❑ `scanf(" %c", &c);`
 - ❑ Pula espaços em branco e lê o caractere seguinte.
- ❑ `c` deve ser uma variável do tipo ***char***.

Caracteres

- ❑ `c = cin.get();`
 - ❑ Lê o próximo caractere.
- ❑ `cin.get(c);`
 - ❑ Lê o próximo caractere.
- ❑ `cin >> c;`
 - ❑ **Pula espaços em branco** e lê o caractere seguinte.
- ❑ **c** deve ser uma variável do tipo ***char***.

Linhas Inteiras

■ `gets(s);`

- Lê uma linha excluindo o `'\n'`, descarta o `'\n'` e coloca `'\0'` no final;
- `s` deve ser um vetor do tipo **`char`**.

■ `cin.getline(s, tam);`

- Lê uma linha excluindo o `'\n'`, descarta o `'\n'` e coloca `'\0'` no final;
- `s` deve ser um objeto da classe **`string`**.

Strings

- ❑ `scanf("%s", s);`
 - ❑ Lê uma única palavra;
 - ❑ `s` deve ser um vetor do tipo ***char***.
- ❑ `cin >> s;`
 - ❑ Lê uma única palavra;
 - ❑ `s` deve ser um objeto da classe ***string***.

Formatos *scanf*

Formato	Saída	Exemplo
<i>c</i>	Caractere	A
<i>d</i>	Inteiro decimal com sinal	392
<i>f</i>	Ponto flutuante decimal	392.65
<i>lf</i>	Ponto flutuante precisão dupla	
<i>s</i>	String	

Entrada

- Por exemplo, suponha um programa que deve ler dois números da entrada e calcular o valor absoluto de sua diferença
- Ler dados até quando?
 - Se não foi especificado um critério de parada, os dados são lidos até que acabem.

Entrada

```
int a, b, resultado;  
while(cin>>a>>b){  
    if(a>b)  
        resultado = a-b;  
    else  
        resultado = b-a;  
}
```

Saída

- Em nossos programas híbridos, recomenda-se a utilização das instruções:
 - ***printf***: Imprime dados no formato especificado (vide próximos slides);
 - ***cout***: Imprime dados no formato especificado pela variável, ou texto entre aspas duplas.

cout

- ❑ `cout<<"texto";`
 - ❑ Imprime o texto entre aspas duplas.
- ❑ `cout<<variavel;`
 - ❑ Imprime o conteúdo da variável.
- ❑ `cout<<conteudo<<endl;`
 - ❑ Imprime e depois quebra a linha (endl).

cout

- ▣ ***setw***

- ▣ Define o tamanho do campo a ser impresso.

- ▣ ***setprecision***

- ▣ Define o número de casas decimais.

- ▣ ***setiosflags***

- ▣ Define a apresentação entre ponto decimal, notação científica, etc.

- ▣ Estes manipuladores estão definidos na biblioteca ***iomanip***.

cout

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

int main () {
    double f =3.14159;

    //imprime com 5 casas decimais
    cout << setprecision(5) << f << endl;
    //imprime com 9 casas decimais
    cout << setprecision(9) << f << endl;
    return 0;
}
```

printf

- Sintaxe:

`printf("%[formato]", [variavel]);`

Em que:

- **[formato]** define o tipo da variável a ser impressa
- **[variável]** deve ser uma variável de um tipo compatível com o formato.

printf

Formato	Saída	Exemplo
<i>c</i>	Caractere	A
<i>d</i>	Inteiro decimal com sinal	392
<i>e</i>	Notação científica, usando o caractere e	3.9265e+2
<i>E</i>	Notação científica, usando o caractere E	3.9265E+2
.nformato	Número de casas decimais (<i>n</i>)	".2f"
<i>f</i>	Ponto flutuante decimal	392.65
<i>lf</i>	Ponto flutuante precisão dupla	
<i>s</i>	String	

printf

```
//imprime uma variável inteira
printf("%d\n", variavel_inteira);

//imprime um caractere e uma string
printf("%c %s\n", variavel_caractere, vetor_caracteres);

//imprime um número real com 8 casas decimais e
//depois com apenas 2 casas decimais
printf("%f %.2f\n", ponto_flutuante, ponto_flutuante);
```



Perguntas?