Fundamentos de Programação

Operadores Aritméticos e de Atribuição

Variáveis do Tipo String

Uma variável capaz de arrmazenar uma string deve ser declarada informando-se qual o número máximo de caracteres que ela poderá armazenar.

Exemplo:

```
char Nome[30]; // isto define que a variável poderá armazenar
               // uma string de até 29 caracteres.
```

Ao trabalharmos com strings deve-se incluir o arquivo de cabeçalho (diretiva) ***string.h***

Variáveis do Tipo String

```c
#include <stdio.h>

int main() {
    char nome[50]; // Declara uma string com tamanho máximo de 50 caracteres

    printf("Digite o seu nome: ");
    scanf("%s", nome); // Lê a string e a armazena na variável 'nome'

    printf("Ola, %s! Bem-vindo ao programa.\n", nome);

    return 0;
}
```

Variáveis do Tipo String

```c
#include <stdio.h>

int main() {
    char nome[50]; // Declara uma string com tamanho máximo de 50 caracteres

    printf("Digite o seu nome: ");
    fgets(nome, sizeof(nome), stdin); // Lê uma linha inteira, incluindo espaços em branco

    printf("Ola, %s! Bem-vindo ao programa.\n", nome);

    return 0;
}
```

Variáveis do Tipo String

Por que meu código não lê corretamente as entradas de minhas strings????

Em C, fflush(stdin) é uma prática comum para "limpar" o buffer de entrada antes de ler uma string após a leitura de outros tipos de dados usando funções como scanf. A razão pela qual algumas pessoas utilizam fflush(stdin) antes de ler uma string após a leitura de outros tipos de dados é que o scanf deixa caracteres de nova linha (como o Enter) no buffer de entrada, e isso pode interferir na leitura da próxima entrada de dados.

```c
char nome[50];
char nome2[50];
printf("digite seu nome:\n");
scanf("%s",nome);

printf("nome: %s\n",nome);

fflush(stdin);
printf("digite outro nome\n");

fgets(nome2,sizeof(nome2),stdin);
printf("nome2: %s\n",nome2);
return 0;
```

Faça um programa em C para calcular a fórmula de báscara!

Operadores

- Operadores são usados para realizar operações entre variáveis, constantes e expressões
- O Python divide os operadores nos seguintes grupos:
  - Aritméticos
  - Atribuição

  → Serão vistos na aula de hoje
  - Comparação
  - Lógicos
  - Identidade
  - Associação

  → Serão vistos na aula seguinte

Operadores Aritméticos

- São usados entre valores numéricos para realizar operações matemáticas comuns:

Operador	Exemplo	Comentário
=	**x = y**	**Atribui o valor de y a x**
+=	x += y	Equivale a x = x + y
-=	x -= y	Equivale a x = x – y
*=	x *= y	Equivale a x = x * y
/=	x /= y	Equivale a x = x / y
%=	x %= y	Equivale a x = x % y

Funções Matemáticas (math.h)

Função	Exemplo	Comentário
`ceil`	`ceil(x)`	Arredonda o número real para cima; ceil(3.2) é 4
`cos`	`cos(x)`	Cosseno de x (x em radianos)
`exp`	`exp(x)`	**e** elevado à potencia x
`fabs`	`fabs(x)`	Valor absoluto de x
`floor`	`floor(x)`	Arredonda o número real para baixo; floor(3.2) é 3
`log`	`log(x)`	Logaritmo natural de x
`log10`	`log10(x)`	Logaritmo decimal de x
`pow`	`pow(x, y)`	Calcula x elevado à potência y
`sin`	`sin(x)`	Seno de x
`sqrt`	`sqrt(x)`	Raiz quadrada de x
`tan`	`tan(x)`	Tangente de x

Funções Matemáticas

```c
#include <stdio.h>
#include <math.h>

int main() {
    double base = 2.0;
    double expoente = 3.0;
    double resultado = pow(base, expoente);
    printf("%.1f elevado a %.1f é %.1f\n", base, expoente, resultado);
    return 0;
}
```

Funções Matemáticas

```c
#include <stdio.h>
#include <math.h>

int main() {
    double numero = 16.0;
    double resultado = sqrt(numero);
    printf("A raiz quadrada de %.1f é %.1f\n", numero, resultado);

    numero = 100.0;
    resultado = log(numero);
    printf("O logaritmo natural de %.1f é %.1f\n", numero, resultado);

    return 0;
}
```

Funções Matemáticas

```c
#include <stdio.h>
#include <math.h>

//M_PI é uma constante de PI definida pela biblioteca math.h

int main() {
    double angulo = 45.0;
    double resultado = sin(angulo * M_PI / 180.0);
    printf("O seno de %.1f graus é %.3f\n", angulo, resultado);

    angulo = 60.0;
    resultado = cos(angulo * M_PI / 180.0);
    printf("O cosseno de %.1f graus é %.3f\n", angulo, resultado);

    return 0;
}
```

Funções Matemáticas

Mas e quais as diferenças em relação ao incremento ++i e i++ ?

i++ é conhecido como pós-incremento, enquanto ++i é chamado de pré incremento.

i++ é pós-incremento porque ele incrementa o valor de i em 1 depois que a operação é concluída.

```
int i,j;
i = 1:
j = i++;
```

Aqui o valor de j = 1, mas i = 2. Aqui o valor de i será atribuído a j primeiro e depois incrementado.

Funções Matemáticas

Aqui o valor de j = 2, mas i = 2. Aqui o valor de i será atribuído a j após a incrementação de i. Da mesma forma, ++i será executado antes de j = i

```
int i, j;
i = 1;
j = ++i;
```

Operadores de Comparação

- São usados para realizar comparação entre dois valores:

Operador	Nome	Sintaxe	Exemplo	Resultado
==	Igual	x == y	10 == 3	Falso
!=	Diferente	x != y	12.6 != 5.1	Verdadeiro
>	Maior que	x > y	3 > 3.2	Falso
<	Menor que	x < y	4 < 4	Falso
>=	Maior ou Igual	x >= y	‘b’ >= ‘a’	Verdadeiro
<=	Menor ou Igual	x <= y	‘A’ <= ‘a’	Verdadeiro

Operadores Lógicos

```c
// Operador ==
int a = 5, b = 5;
printf("a == b: %d\n", a == b);

// Operador !=
int c = 10, d = 5;
printf("c != d: %d\n", c != d);

// Operador <
int e = 10, f = 5;
printf("e < f: %d\n", e < f);
```

Operadores Lógicos

São usados para combinar expressões de comparação e lógicas:

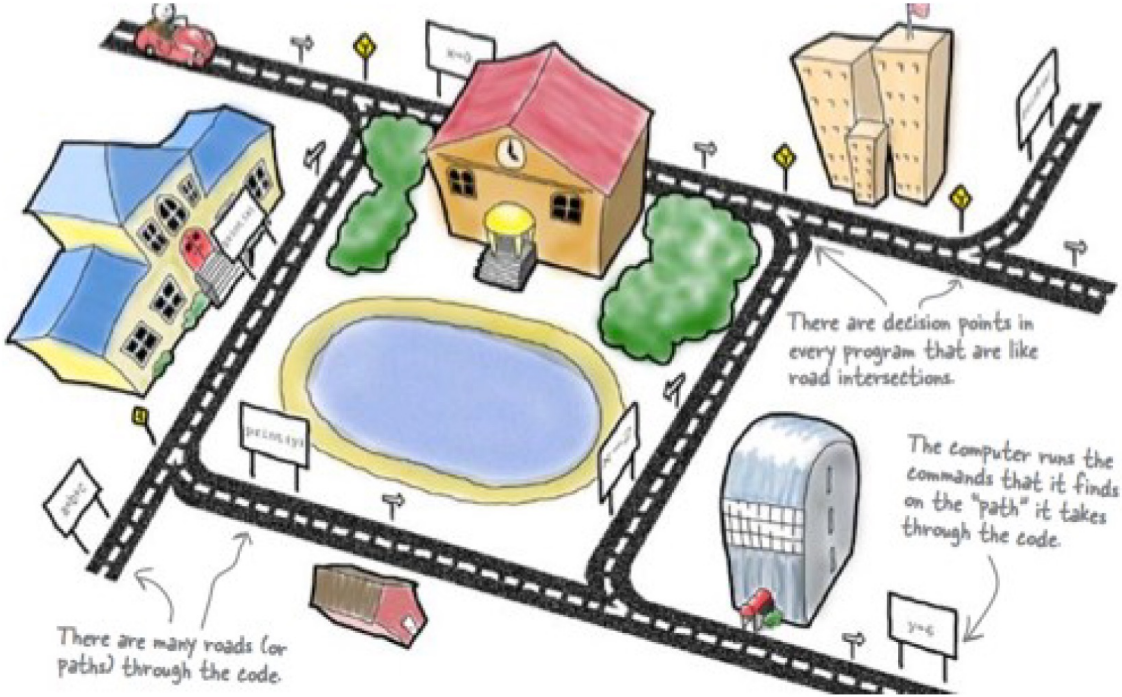
Operador	Descrição	Sintaxe
&&	Retorna verdadeiro se todos os elementos envolvidos forem verdadeiros	x && y
||	Retorna falso se todos os elementos envolvidos forem falsos	x || y
!	Retorna o inverso do elemento (Negação – not)	!x

```c
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool condicao1 = true;
    bool condicao2 = false;

    // Operação AND (&&)
    bool resultado_and = condicao1 && condicao2;
    printf("Resultado AND: %d\n", resultado_and);

    // Operação OR (||)
    bool resultado_or = condicao1 || condicao2;
    printf("Resultado OR: %d\n", resultado_or);

    return 0;
}
```

Categoria	Operadores
Parênteses	`()`
Pós-incremento	`x++`, `x--`
Pré-incremento	`++x`, `--x`
Operadores unários	`+`, `-`, `!`
Multiplicação/Divisão/Resto	`*`, `/`, `%`
Adição/Subtração	`+`, `-`
Relacional	`<`, `>`, `<=`, `>=`
Igualdade	`==`, `!=`
Lógico AND	`&&`
Lógico OR	`||`
Condicional	`?:`
Atribuição	`=`, `+=`, `-=`, `*=`, `/=`, `%=`, `<<=`, `>>=`, `&=`, `|=`, `^=`
Separador de sequência	`,`

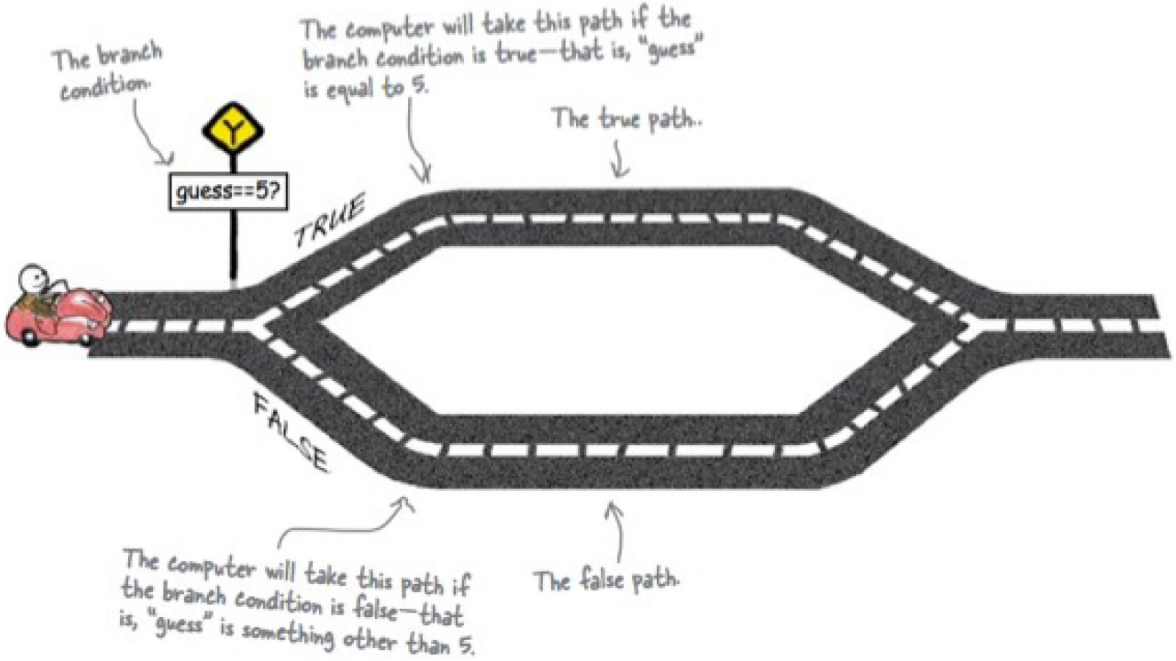
Condicionais IF - ELSE

A execução de um código é semelhante a um veículo transitando por uma cidade



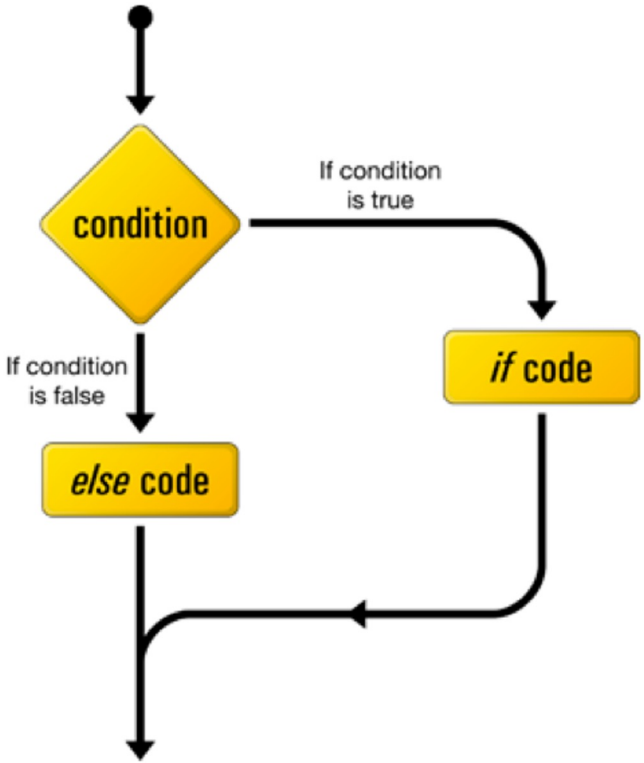
Estrutura de Seleção

Em um determinado momento, chegamos em uma bifurcação. Qual caminho devemos seguir?



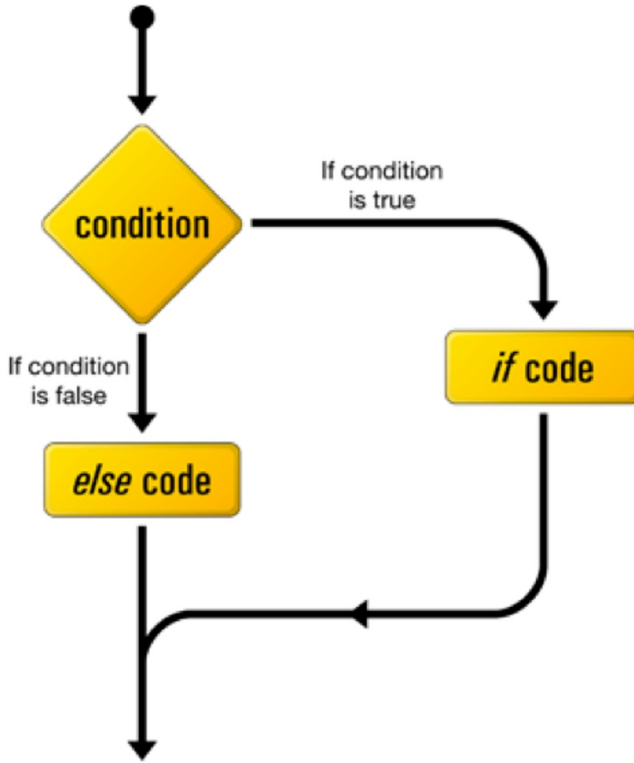
Condicionais IF - ELSE

```c
// Verifica se o número é positivo usando a estrutura if
    if (numero > 0) {
        printf("O número inserido é positivo.\n");
    }
```



Condicionais IF - ELSE

```c
// Verifica se o número é positivo, negativo ou zero usando if-else
    if (numero > 0) {
        printf("O número inserido é positivo.\n");
    } else if (numero < 0) {
        printf("O número inserido é negativo.\n");
    } else {
        printf("O número inserido é zero.\n");
    }
```



Exercício: Menu de Opções

Escreva um programa em C que exiba um menu de opções para o usuário e execute a ação correspondente à opção selecionada. Use os comandos if e else para testar. As ações podem ser apenas imprimir qual a opção o usuário escolheu. O importante é testar o condicional.

Exercício: Menu de Opções -> uma outra solução?

O COMANDO SWITCH

```c
switch (opcao) {
    case 1:
        printf("Opção 1 selecionada.\n");
        break;
    case 2:
        printf("Opção 2 selecionada.\n");
        break;
    case 3:
        printf("Opção 3 selecionada.\n");
        break;
    case 4:
        printf("Opção 4 selecionada.\n");
        break;
    default:
        printf("Opção inválida.\n");
}
```

Vamos adicionar ao código de nossa fórmula de Báscara condições IF e Else. Use as condições para testar se existem duas raízes reais, um única raiz ou não existem raízes para o polinômio.

Desafio de Programação!

Referências

- https://en.cppreference.com/w/c/language
- https://www.cprogramming.com/reference/
- https://cplusplus.com/