

---

# Laboratório de Introdução à Ciência da Computação I

## **Aula 2 – Comandos de Decisão**

Professor:

Jó Ueyama

Monitor PAE:

Fernando Alva

# Sumário

---

- Comandos
  - if
  - if-else
  - if/if-else aninhados
  - else-if
  - switch
- Operadores lógicos
- Operador ternário
- Exercícios

# Comando if

---

```
if (expressão de teste)
instrução;
```

```
if (expressão de teste)
{
Instrução_1;
Instrução_2;
...
Instrução_n;
}
```

# Comando if

---

Exemplo:

```
int main()
{
    char ch;
    ch = getche();
    if (ch == 'p')
    {
        printf("\n Voce pressionou a tecla p.\n");
    }
    system("pause" );
}
```

# Comando if-else

---

```
if (expressão de teste)
    instrução_1;
else
    instrução_2;
```

```
if (expressão de teste)
{
    instrução_1;
    Instrução_2;
    ...
    Instrução_n;
}
else
{
    instrução_1;
    instrução_2;
    ...
    Instrução_n;
}
```

# Comando if-else

---

Exemplo:

```
int main()
{
    char ch;
    ch = getche();
    if (ch == 'p')
    {
        printf("\n Voce pressionou a tecla p.\n");
    }
    else
    {
        printf("\n Voce não pressionou a tecla p.\n");
    }

    system("pause" );
}
```

# Comando if/if-else aninhados

---

```
if (expressão de teste_1)
  if (expressão de teste_2)
    instrução_1;
  else
    instrução_2;
else
  instrução_3;
```

# Exemplo de if/if-else aninhados

```
int main()
{
    char ch1, ch2;
    printf("\n Entre caractere 1:");
    ch1 = getche();
    if (ch1 == 'p')
    {
        printf("\n Entre caractere 2:");
        ch2 = getche();
        if (ch2 == 'q')
        {
            printf("\n Você digitou p e q.\n");
        }
        else
        {
            printf("\n Você digitou p e não q.\n");
        }
    }
    else
    {
        printf("\n Voce NÃO digitou p e q.\n");
    }
    system("PAUSE");
}
```



## Comando <else if>

---

```
if (expressão de teste_1)
    instrução_1;
else if (expressão de teste_2)
    instrução_2;
```

# Operadores para expressões de teste

---

## Relacionais

> maior

>= maior ou igual

< menor

<= menor ou igual

== igualdade

!= diferente

## Lógicos

&& E

|| OU

! Negação (unário)

# Exemplo com operadores lógicos

```
int main()
{
    char ch1, ch2;
    printf("\n Entre caractere 1:");
    ch1 = getche();
    printf("\n Entre caractere 2:");
    ch2 = getche();

    if (ch1 == 'p' && ch2 == 'q')
    {
        printf("\n Você digitou p e q.\n");
    }
    else if (ch1 == 'p' || ch2 == 'q')
    {
        printf("\n Você digitou p ou q.\n");
    }

    if (!(ch1 == 'p') && !(ch2 == 'q'))
    {
        printf("\n Você NÃO digitou p e nem q.\n");
    }

    system("PAUSE");
}
```

# Precedência de operadores

Operador	Tipo
! - ++ --	Lógico e Aritmético (unários)
* / %	Aritmético
+ -	Aritmético
< <= > >=	Relacional
== !=	Relacional
&&	Lógico
+= -= *= /= %=	Aritmético de atribuição

# Comando switch

---

- Comando de decisão alternativo ao <else-if>
- Considerado mais flexível e de formato mais claro, principalmente quando o programa precisa escolher “uma” entre várias alternativas de decisão

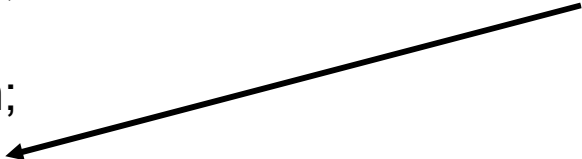
# Comando switch

---

switch (expressão constante)

```
{  
  case constante_1:  
    instrução_1;  
    ...  
    instrução_n;  
    break;  
  case constante_2:  
    instrução_1;  
    ...  
    instrução_n;  
    break;  
  default:  
    instrução_1;  
    ...  
    instrução_n;  
}
```

O que ocorre se remover o <break> ?



# Exemplo com switch

```
int main()
{
    float num1, num2;
    char op;
    printf("Digite na seguinte ordem: valor 1 \"operador\" valor 2 \n");
    scanf("%f %c %f", &num1, &op, &num2);

    switch(op)
    {
        case '+':
            printf(" = %f\n",num1 + num2);
            break;

        case '-':
            printf(" = %f\n",num1 - num2);
            break;

        default:
            printf("Operador desconhecido \n");
    }

    system("PAUSE");
}
```

# Operador condicional ternário

---

- Maneira compacta de expressar um simples instrução if-else
- Único operador ternário em C

condição ? expressão\_1 : expressão\_2

Exemplo:

```
max = (num1 > num2) ? num1 : num2;
```



# Referências

---

Ascencio AFG, Campos EAV. Fundamentos de programação de computadores. São Paulo : Pearson Prentice Hall, 2006. 385 p.

VICTORINE VIVIANE MIZRAHI, Treinamento em Linguagem C – Módulo 1 e Módulo 2, Makron Books, 1990.