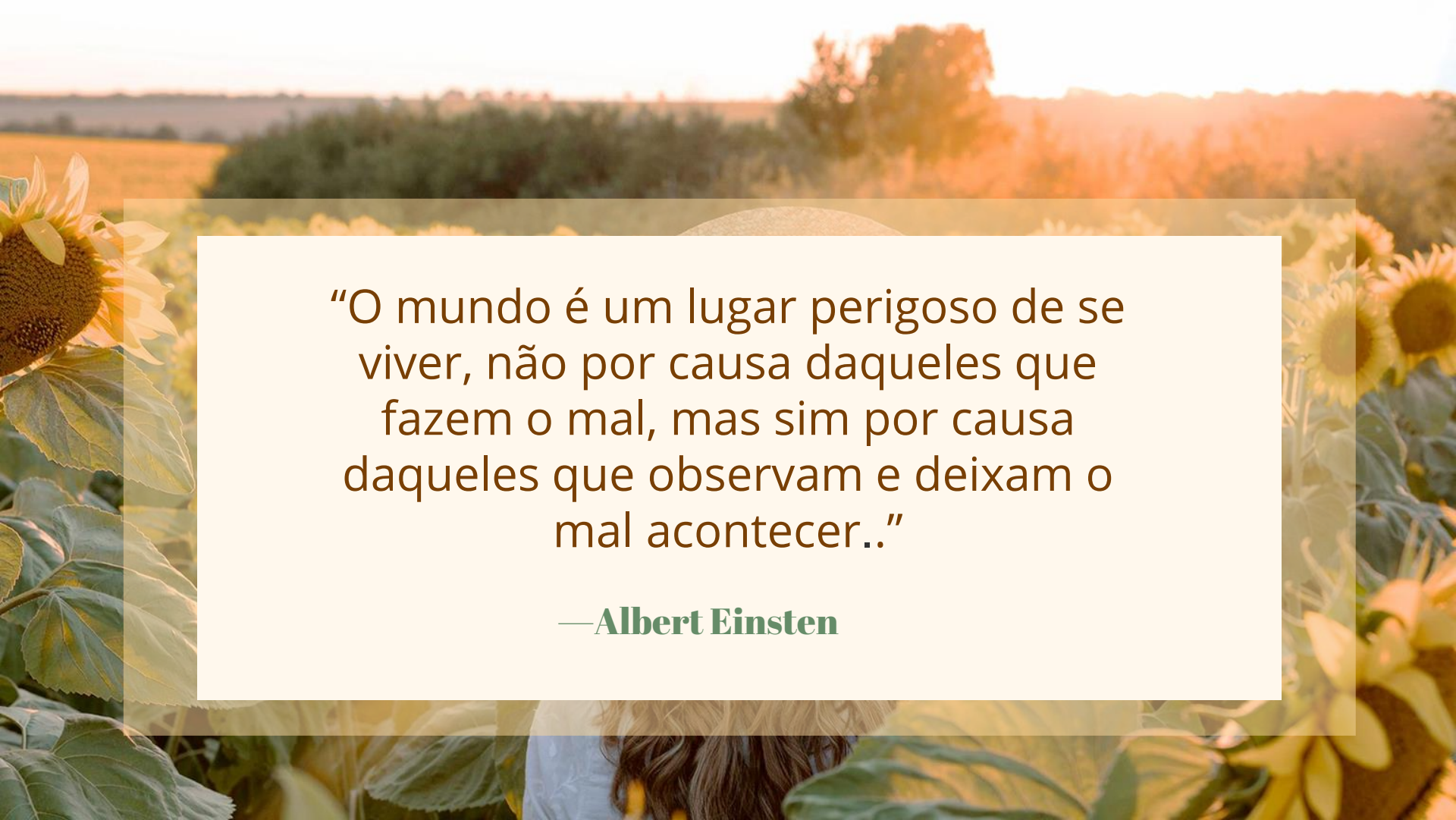




Estruturas condicionais

Aninhadas, Múltiplas Escolhas



“O mundo é um lugar perigoso de se viver, não por causa daqueles que fazem o mal, mas sim por causa daqueles que observam e deixam o mal acontecer..”

—**Albert Einsten**



01

Seleção Encadeada

Homogênea e Heterogênea

Estrutura de Seleção

Permite que durante a execução de um programa, um **comando** ou bloco seja **escolhido** para ser executado, observando uma determinada **condição**



```
14  if (sexo == 'F') {
15      pesoIdeal = (62.1f * altura) - 44.7;
16  }
17  else {
18      pesoIdeal = (72.7f * altura) - 58;
19  }
```

Seleção Encadeada

Ocorre quando um determinado **bloco** para ser executado, depender de um conjunto de **possibilidades** ou **combinações**

Seleção Encadeada Homogênea

Quando é possível identificar um padrão lógico de construção de uma estrutura encadeada

Seleção Encadeada Heterogênea

Quando não é possível identificar um padrão lógico de construção de uma estrutura encadeada





```
if (cond1) {  
    if (cond2) {  
        if (cond3) {  
            if (cond4) {  
                comandos;  
            }  
        }  
    }  
}
```

Seleção Encadeada Homogênea

```
if ((cond1) && (cond2) && (cond3) && (cond4)) {  
    comandos;  
}
```





Seleção Encadeada Heterogênea

```
1 comando0;  
2 if (condição1) {  
3     if (condição2) {  
4         comando1;  
5         comando2;  
6     }  
7 } else {  
8     if (condição3) {  
9         comando3;  
10    } else {  
11        if (condição4) {  
12            comando4;  
13            comando5;  
14        }  
15        comando6;  
16        comando7;  
17    }  
18    comando8;  
19 }  
20 comando9;
```



Exercício Exemplo



Crie um programa que **leia** três valores (a, b, c) e informe se estes podem ser ou não os comprimentos dos lados de um triângulo.

Se forem, verificar se é um triângulo **equilátero** (todos os lados iguais), **escaleno** (todos os lados diferentes) ou **isósceles** (dois lados iguais).

Para ser um triângulo, cada lado deve ser menor que a soma dos outros dois




```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main()
5 {
6     float a, b, c;
7     char mensagem[30];
8
9     printf("Digite os valores a, b, c: ");
10    scanf("%f %f %f", &a, &b, &c);
11
12    if ((a < b + c) && (b < a + c) && (c < a + b)) {
13        if ((a == b) && (b == c)) {
14            strcpy(mensagem, "Triangulo EQUILATERO");
15        }
16        else {
17            if ((a == b) || (a == c) || (b == c)){
18                strcpy(mensagem, "Triangulo ISOSCELES");
19            }
20            else {
21                strcpy(mensagem, "Triangulo ESCALENO");
22            }
23        }
24    }
25    else {
26        strcpy(mensagem, "Nao forma um triangulo");
27    }
28    printf("\n\n%s\n\n", mensagem);
29 }
```





02

Otimizando

Sequência IF - ELSE

Otimização

Suponha que uma variável X possa assumir apenas **quatro** valores, sendo eles $V1$, $V2$, $V3$ e $V4$, e que exista um comando diferente que será executado para cada valor armazenado em X

É uma situação **excludente**, se X é igual a $V3$, ele não é igual a $V1$, nem a $V2$ e nem a $V4$



Otimizando IF - ELSE

X == V1	X == V2	X == V3	X == V4	Ação
V	F	F	F	comando1
F	V	F	F	comando2
F	F	V	F	comando3
F	F	F	V	comando4

```
if (X == V1) {  
    comando1;  
}  
if (X == V2) {  
    comando2;  
}  
if (X == V3) {  
    comando3;  
}  
if (X == V4) {  
    comando4;  
}
```



Elimina testes desnecessários



X == V1	X == V2	X == V3	X == V4	Ação
V	-	-	-	comando1
F	V	-	-	comando2
F	F	V	-	comando3
F	F	F	V	comando4

```
3  if (X == V1) {
4      comando1;
5  } else {
6      if (X == V2) {
7          comando2;
8      } else {
9          if (X == V3) {
10             comando3;
11         } else {
12             if (X == V4) {
13                 comando4;
14             }
15         }
16     }
17 }
18
```



Exercício Exemplo 2



Crie um programa que **leia** quatro notas, **calcule** e imprima a **média** e uma **mensagem** de acordo com a tabela abaixo:

Média	Mensagem
Menor que 5,0	Reprovado
De 5,0 a 6,9	Recuperação
De 7,0 a 10,0	Aprovado



```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main()
5  {
6      float n1, n2, n3, n4, media;
7      char status[15];
8      printf("Nota 1: ");
9      scanf("%f", &n1);
10     printf("Nota 2: ");
11     scanf("%f", &n2);
12     printf("Nota 3: ");
13     scanf("%f", &n3);
14     printf("Nota 4: ");
15     scanf("%f", &n4);
16     media = (n1 + n2 + n3 + n4) / 4;
17     if (media < 5){
18         strcpy(status, "Reprovado");
19     } else {
20         if (media < 7) {
21             strcpy(status, "Recuperação");
22         } else {
23             strcpy(status, "Aprovado");
24         }
25     }
26     printf("\nMedia: %.1f\nStatus: %s\n\n", media, status);
27 }
```



Exercício Exemplo 3

Crie um programa que simule uma calculadora. Leia dois números inteiros e a operação desejada conforme o quadro abaixo. Imprima o resultado. Lembre-se que se a operação for divisão e o segundo número for igual a zero, deve ser impresso uma mensagem de erro.



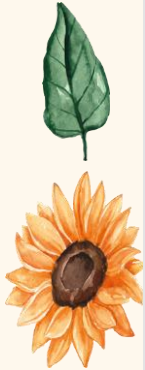
Símbolo	Operação Aritmética
+	Adição
-	Subtração
*	Multiplicação
/	Divisão


```

1 #include <stdio.h>
2 #include <string.h>
3 int main()
4 {
5     int n1, n2;
6     float resultado = 0;
7     char operacao;
8     printf("Digite o primeiro valor: ");
9     scanf("%d", &n1);
10    printf("Digite o segundo valor: ");
11    scanf("%d", &n2);
12    printf("Digite + - * /: ");
13    fflush(stdin);
14    scanf("%c", &operacao);
15    if ((operacao == '/') && (n2 == 0)) {
16        printf("\nERRO!!!");
17    } else {
18        if (operacao == '/') {
19            resultado = (float) n1 / n2;
20        } else {
21            if (operacao == '+') {
22                resultado = n1 + n2;
23            } else {
24                if (operacao == '-') {
25                    resultado = n1 - n2;
26                } else {
27                    if (operacao == '*') {
28                        resultado = n1 * n2;
29                    }
30                }
31            }
32        }
33        printf("%d %c %d = %.1f\n\n", n1, operacao, n2, resultado);
34    }
35    printf("\n\nFim de Programa\n");
36 }

```

14V	15						34
14F	17V	18				32	34
14F	17F	20V	21			32	34
14F	17F	20F	23V	24		32	34
14F	17F	20F	23F	26V	27	32	34
14F	17F	20F	23F	26F		32	34





03

Múltiplas Escolhas

Estrutura de Seleção de Múltipla Escolha

Utilizada quando existem **várias** seleções encadeadas homogêneas do tipo

if comando **else** comando

O comando **switch** é utilizado com um conjunto de valores para realizar a **seleção** do bloco de execução



```
if (X == V1) {
    comando1;
} else {
    if (X == V2) {
        comando2;
    } else {
        if (X == V3) {
            comando3;
        } else {
            if (X == V4) {
                comando4;
            } else {
                comando5;
            }
        }
    }
}
```

Seleção SWITCH-CASE

```
switch(X) {
    case V1:
        comando1;
        break;
    case V2:
        comando2;
        break;
    case V3:
        comando3;
        break;
    case V4:
        comando4;
        break;
    default:
        comando5;
}
```



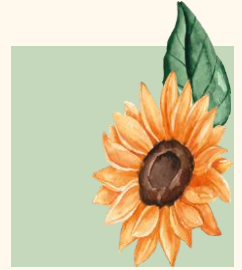
Otimizando o SWITCH-CASE



```
switch(X) {  
    case V1: case V2:  
        comando1;  
        break;  
    case V3:  
    case V4:  
    case V5:  
        comando2;  
        break;  
    case V6:  
        comando3;  
        break;  
    case V7: case V8: case V9: case V10:  
        comando4;  
        break;  
    default:  
        comando5;  
}
```



Exercício Exemplo 4



Crie um programa que leia um valor de 1 a 7 e mostre para o usuário o dia da semana correspondente. Considere domingo = 1, segunda-feira = 2 e assim por diante

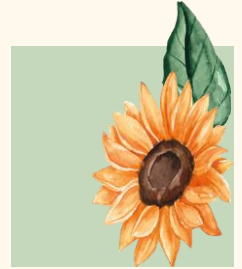




```
1 #include <stdio.h>
2 #include <string.h>
3 int main()
4 {
5     int dia;
6     char diaSemana[15];
7     printf("Digite o numero do dia da semana (entre 1 e 7): ");
8     scanf("%d", &dia);
9     switch (dia)
10    {
11        case 1:
12            strcpy(diaSemana, "domingo");
13            break;
14        case 2:
15            strcpy(diaSemana, "segunda-feira");
16            break;
17        case 3:
18            strcpy(diaSemana, "terca-feira");
19            break;
20        case 4:
21            strcpy(diaSemana, "quarta-feira");
22            break;
23        case 5:
24            strcpy(diaSemana, "quinta-feira");
25            break;
26        case 6:
27            strcpy(diaSemana, "sexta-feira");
28            break;
29        case 7:
30            strcpy(diaSemana, "sabado");
31            break;
32        default:
33            strcpy(diaSemana, "Dia Invalido!");
34    }
35    printf("\n\n%s\n\n", diaSemana);
36 }
```



Exercício Exemplo 5



Desenvolver um programa que leia o mês e o ano de uma data e exiba a quantidade de dias que aquele mês possui. Lembre-se que em anos bissextos, o mês de fevereiro possui 29 dias.

OBS: Um ano é bissexto se ele for múltiplo de 4, exceto quando ele for múltiplo de 100. Os anos múltiplos de 100 somente são bissextos quando são múltiplos de 400, usado a partir de 1752 (por exemplo 1800 não é bissexto, mas 2000 é)




```
1 #include <stdio.h>
2 int main()
3 {
4     int mes, ano;
5     int qtdDias, valida = 0;
6     printf("Mes (mm): ");
7     scanf("%d", &mes);
8     printf("Ano (aaaa): ");
9     scanf("%d", &ano);
10    switch(mes)
11    {
12        case 1: case 3: case 5: case 7: case 8: case 10: case 12:
13            qtdDias = 31;
14            break;
15        case 4: case 6: case 9: case 11:
16            qtdDias = 30;
17            break;
18        case 2:
19            if ( ( (ano % 4 == 0) && !(ano % 100 == 0) ) || (ano % 400 == 0) ) {
20                qtdDias = 29;
21            }
22            else {
23                qtdDias = 28;
24            }
25            break;
26        default:
27            qtdDias = 0;
28    }
29    if (qtdDias == 0) {
30        printf("\nMes inexistente");
31    }
32    else {
33        printf("\nO mes %d do ano %d possui %d dias\n\n", mes, ano, qtdDias);
34    }
35 }
```





```
1  #include <stdio.h>
2  #include <string.h>
3  int main()
4  {
5      int x;
6      char operacao[20];
7      printf("Opcao: ");
8      scanf("%d", &x);
9      switch (x){
10         case 1 ... 10:
11             strcpy(operacao, "Prog1");
12             break;
13         case 11 ... 20:
14             strcpy(operacao, "Prog2");
15             break;
16         case 21 ... 30:
17             strcpy(operacao, "Prog3");
18             break;
19         case 31 ... 40:
20             strcpy(operacao, "FIM");
21             break;
22         default:
23             strcpy(operacao, "Opcao Invalida");
24     }
25     printf("\n%s\n\n", operacao);
26 }
```





04

Exercícios

1 - Origem Produto

Escreva um programa que, tendo como dado de entrada o código de origem de um produto, mostre sua procedência (conforme tabela). Caso o código não seja nenhum dos especificados, o produto deve ser encarado como importado



Código de Origem	Procedência
1	Sul
2	Norte
3 ou 4	Nordeste
5, 6 ou 7	Sudeste
8 até 10	Centro-Oeste

2 – Cargo e Salário

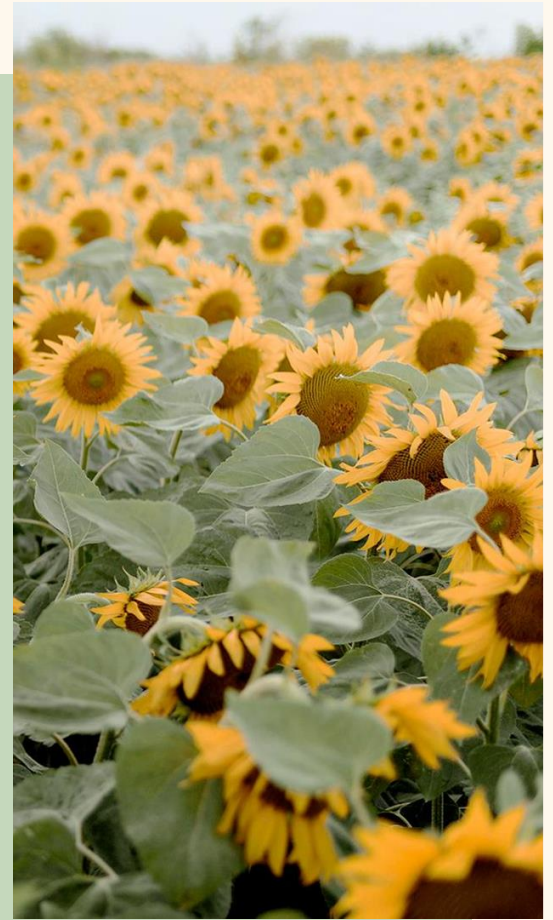
Escreva um programa que receba o código do cargo de um funcionário e o seu salário atual e imprima seu cargo, o percentual de aumento ao qual este funcionário tem direito e também o valor do novo salário, seguindo as regras do quadro

Código	Cargo	Percentual
1	Escriturário	50 %
2	Secretário	35 %
3	Caixa	20 %
4	Gerente	10 %
5	Diretor	Não tem aumento



Obrigada!

Alguma pergunta?
professora@lucilia.com.br





Créditos

- Template: Slidesgo
- Deitel: “Java – Como programar”
- Ascencio: “Fundamentos de programação”
- Wikipedia

