

# Condições

---

PDSI  
Heitor Ramos

Revendo o assunto até agora

# Reverendo o assunto até agora

```
int a;  
int b;  
a = 10;  
b = a;
```

## Revendo o assunto até agora

```
a = 2 + 2;
```

```
a = 10 / 3;
```

```
a = a + 1;
```

## Revendo o assunto até agora

```
a = 2 + 2;  
float a = (float) 10 / 3;    // Cast ou conversão de tipos  
a = a + 1;
```

# scanf

---

# Nova função

## scanf

- O oposto do printf
- Se o printf imprime na tela o scanf lê do teclado
  - Na verdade são chamados de entrada e saída padrão
  - Fora do escopo de AEDS1

# scanf

```
#include <stdio.h>
int main(void) {
    int n;
    printf("Digite um numero: \n");
    scanf("%d", &n);
    printf("O valor digitado foi
%d\n",n);
    return 0;
}
```



# scanf

```
#include <stdio.h>

int main(void) {
    int valor1, valor2, valor3;
    printf("Digite tres numeros: \n");
    scanf("%d %d %d", &valor1, &valor2, &valor3);
    printf("O valores digitados foram %d %d %d\n",
           valor1, valor2, valor3);
    return 0;
}
```

# O operador & ("address-of")

Toda variável tem um endereço de memória associado a ela. Esse endereço é o local onde essa variável é armazenada no sistema. O operador & retorna o endereço de uma determinada variável

```
#include <stdio.h>
int main(void) {
    int n = 8;
    printf("valor %d, endereco 0x%x\n", n, &n);
    return 0;
}
```

## O operador & ("address-of")

- É necessário usar o operador & no comando scanf, pois esse operador indica que o valor digitado deve ser colocado no endereço referente a uma variável.
- **Esquecer de colocar o & comercial é um erro muito comum que pode ocasionar erros de execução.**

# Usamos os mesmos formatos do printf

<b>Tipo</b>	<b>Num de bits</b>	<b>Formato</b>
-----		
<b>char</b>	<b>8</b>	<b>%c</b>
unsigned char	8	%C
signed char	8	%c
<b>int</b>	<b>16</b>	<b>%d ou %i</b>
unsigned int	16	%u
signed int	16	%i
short int	16	%hi
unsigned short int	16	%hu
signed short int	16	%hi
long int	32	%li
signed long int	32	%li
unsigned long int	32	%lu
<b>float</b>	<b>32</b>	<b>%f</b>
<b>double</b>	<b>64</b>	<b>%lf</b>
long double	80/128	%Lf

# if/else

---

# Novo problema

Ler um número usando scanf e determinar se o mesmo é par ou ímpar

## Novo problema

```
#include <stdio.h>
int main(void) {
    int n;
    printf("Digite um numero: \n");
    scanf("%d", &n);
    if (n % 2 == 0) {
        printf("O valor digitado eh par\n");
    } else {
        printf("O valor digitado eh impar\n");
    }
    return 0;
}
```

# Operador %

Retorna o resto de uma divisão

```
int a = 10;  
int quociente = a / 3;  
int resto = a % 3;
```



# Operador %

Retorna o resto de uma divisão

```
int a = 10;  
int quociente = a / 3;    // quociente vai ser igual a 3  
int resto = a % 3;        // o resto vai ser igual a 1
```

## if, else

Um comando condicional é aquele que permite decidir se um determinado **bloco de comandos** deve ou não ser executado, a partir do resultado de uma expressão **relacional** ou **lógica**.

## if, else

**If** e **else** indicam bifurcações na execução de um programa. Sabendo que cada { e } indica um bloco:

```
if (n % 2 == 0) {      // ou eu entro neste bloco
    printf("O valor digitado eh par\n");
} else {               // ou no else, nunca nos 2
    printf("O valor digitado eh impar\n");
}
```

# blocos

**If** e **else** indicam bifurcações na execução de um programa. Sabendo que cada { e } indica um bloco:

```
{      // inicio de um bloco  
...  
}      // fim de um bloco
```

# Novo problema

Ler dois números usando scanf e imprimir o maior deles

# Expressões relacionais

Expressões relacionais são aquelas que realizam uma comparação entre duas expressões e retornam

- Zero (0),
  - se o resultado é falso
- Um (1),
  - ou qualquer outro número diferente de zero, se o resultado é verdadeiro.

# Expressões relacionais

- **(expressao) == (expressao)**
  - Retorna verdadeiro quando as expressões forem iguais.
  - Exemplo: `a == b`
  - `(1 == 1) → 1`
  - `(0 == 1) → 0`
  - `(7 + 2) == (8 + 1) → 1`
- **(expressao) != (expressao)**
  - Retorna verdadeiro quando as expressões forem diferentes.
  - Exemplo: `a != b`
  - `(2 != 2) → 0`
  - `(1 != 3) → 1`
  - `(3 + 5 != 4 + 4) → 0`

# Expressões relacionais

- $(\text{expressao}) > (\text{expressao})$

- Retorna verdadeiro quando a expressão da esquerda tiver valor maior que a expressão da direita.
- Ex:  $a > b$

- $(\text{expressao}) < (\text{expressao})$

- Retorna verdadeiro quando a expressão da esquerda tiver valor menor que a expressão da direita.
- Ex:  $a < b$



# Expressões relacionais

- **(expressao) >= (expressao)**

- Retorna verdadeiro quando a expressão da esquerda tiver valor maior ou igual que a expressão da direita.
- Ex:  $a \geq b$

- **(expressao) <= (expressao)**

- Retorna verdadeiro quando a expressão da esquerda tiver valor menor ou igual que a expressão da direita.
- Ex:  $a \leq b$

# Resolvendo o problema

Ler dois números usando scanf e imprimir o maior deles

## Maior de 2

```
#include <stdio.h>
int main(void) {
    int valor1;
    int valor2;
    printf("Digite dois numeros: \n");
    scanf("%d %d", &valor1, &valor2);
    if (valor1 > valor2) {
        printf("O maior valor foi %d\n", valor1);
    } else {
        printf("O maior valor foi %d\n", valor2);
    }
    return 0;
}
```

# Álgebra Booleana

---

# Outro problema

Para se matricular no prestigioso colégio PDS, um aluno tem que satisfazer as seguintes condições:

1. Ter entre 12 (inclusivo) e 18 (exclusivo) anos  $[12, 18)$
2. Ter nota superior a 90 no ensino fundamental

Como fazer um programa que determina se um aluno pode ou não se matricular?

```
#include <stdio.h>
int main(void) {
    int idade;
    printf("Digite a idade do aluno: \n");
    scanf("%d", &idade);
    float nota_fundamental;
    printf("Digite a nota no fundamental: \n");
    scanf("%f", &nota_fundamental);
    int pode_matricular = 0;
    if (idade >= 12) {
        if (idade < 18) {
            if (nota_fundamental > 90) {
                pode_matricular = 1;
            }
        }
    }
    printf("1 se sim; 0 se não. Resposta foi %d\n",
           pode_matricular);
}
```

# Expressões Booleanas

Expressões lógicas são aquelas que realizam uma operação lógica (**ou, e, nao, etc...**) e retornam verdadeiro ou falso (como as expressões relacionais).

S1M → VERDADEIRO

NÃ0 → FALSO

## e, and, &&

( expressao ) && ( expressao ) : Retorna verdadeiro quando ambas as expressões são verdadeiras. Sua tabela verdade é:

Op1	Op2	RET
FALSO (0)	FALSO (0)	FALSO (0)
FALSO (0)	VERDADEIRO (1)	FALSO (0)
VERDADEIRO (1)	FALSO (0)	FALSO (0)
VERDADEIRO (1)	VERDADEIRO (1)	VERDADEIRO (1)



ou, or, ||

( expressao ) || ( expressao ) : Retorna verdadeiro quando pelo menos uma das expressões é verdadeiras. Sua tabela verdade é:

Op1	Op2	RET
FALSO (0)	FALSO (0)	FALSO (0)
FALSO (0)	VERDADEIRO (1)	VERDADEIRO (1)
VERDADEIRO (1)	FALSO (0)	VERDADEIRO (1)
VERDADEIRO (1)	VERDADEIRO (1)	VERDADEIRO (1)

# não, not, !

! ( expressao ): Retorna verdadeiro quando a expressão é falsa e vice-versa.  
Sua tabela verdade é:

Op1	RET
FALSO (0)	VERDADEIRO (1)
VERDADEIRO (1)	FALSO (0)

# Simplificações Úteis

- $!(a == b)$  é equivalente a  $a != b$
- $!(a != b)$  é equivalente a  $a == b$
- $!(a > b)$  é equivalente a  $a <= b$
- $!(a < b)$  é equivalente a  $a >= b$
- $!(a >= b)$  é equivalente a  $a < b$
- $!(a <= b)$  é equivalente a  $a > b$

```
#include <stdio.h>

int main(void) {
    int idade;
    printf("Digite a idade do aluno: \n");
    scanf("%d", &idade);

    float nota_fundamental;
    printf("Digite a nota no fundamental: \n");
    scanf("%f", &nota_fundamental);

    int tem_idade = (idade >= 12) && (idade < 18);
    int pode_matricular = tem_idade && (nota_fundamental > 90);

    printf("1 se sim; 0 se não. Resposta foi %d\n",
           pode_matricular);
}
```

# switch

---

# Último problema

Dado um conceito de um aluno (A até F), estabeleça como nota do aluno a média da menor nota possível dentro do conceito com a maior nota possível.

Por exemplo, se o aluno tem conceito B:

$$\text{nota} = (90 + 80) / 2$$

# Conceito

```
#include <stdio.h>
int main(void) {
    char conceito;
    printf("Digite o conceito: ");
    scanf("%c", &conceito);
    float nota = 0;
    if (conceito == 'A') {
        nota = (100 + 90) / 2;
    } else if (conceito == 'B') {
        nota = (80 + 90) / 2;
    }
    // . . . e assim por diante
}
```

# switch

Serve para simplificar expressões de if/else

```
switch (variável inteira) {  
    case valor1: comandos break;  
    case valor2: comandos break;  
    ...  
    default: comandos;  
}
```



```
switch(conceito) {  
  case 'A':  
    nota = (100 + 90) / 2;  
    break;  
  case 'B':  
    nota = (90 + 80) / 2;  
    break;  
  case 'C':  
    nota = (80 + 70) / 2;  
    break;  
  case 'D':  
    nota = (70 + 60) / 2;  
    break;  
  case 'E':  
    nota = (60 + 40) / 2;  
    break;  
  default:  
    nota = 0;  
    break;  
}
```

# Exercícios

Escreva um programa que, dado duas datas, determine qual delas ocorreu cronologicamente antes em relação a outra. Cada data é composta de 3 números inteiros, um representando o dia, outro o mês e outro o ano.

# Exercícios

Escreva um programa que recebe um operando, um operador aritmético e outro operando e calcule a operação indicada. As operações possíveis são soma(+), subtração(-), multiplicação(\*) e divisão(/).

## Exemplo

10 + 2

imprime 12 na tela.

# Exercícios

Escreva um programa que ordene três números. Tente encontrar uma versão com apenas 3 comandos **if**.