

# Programação 1

**Atribuição, operadores aritméticos,  
entrada de dados**

# Armazenando na memória

```
#include <stdio.h>
#include <stdlib.h>
```

tipo de  
variável

```
main()
{
```

Declaração  
de variáveis

```
int ano;
```

comando de atribuição

```
ano = 2018;
```

```
printf("Estamos em %d\n", ano);
```

```
system("pause");
```

```
}
```

código de formatação para inteiros

2018

Memória

ano

2018



OBS: Todas as variáveis devem ser declaradas

# Armazenando na memória

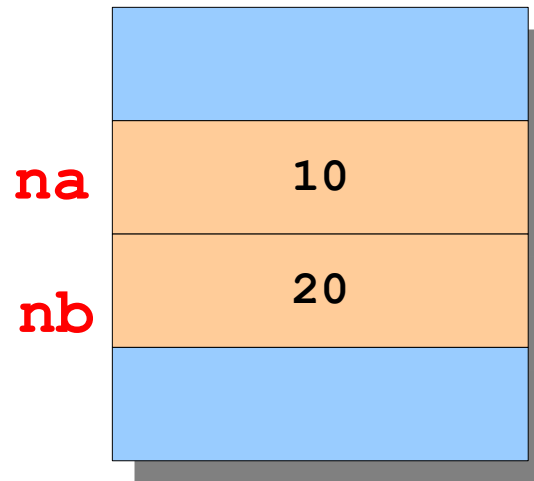
```
#include <stdio.h>
#include <stdlib.h>
```

```
main()
{
    int na,nb;
```

```
    na = 10;
    nb = 20;
    printf("a:%d    b:%d\n",na,nb) ;
```

```
    na = nb;
    printf("a:%d    b:%d\n",na,nb) ;
    system("pause") ;
}
```

Memória



O que será impresso aqui?

A light yellow rounded rectangle with a blue border. It contains two lines of text: 'a:10 b:20' and 'a:? b:?.'. An arrow points from the question box to the 'a:?' part of the second line.

a:10	b:20
a:?	b:?

# Armazenando na memória

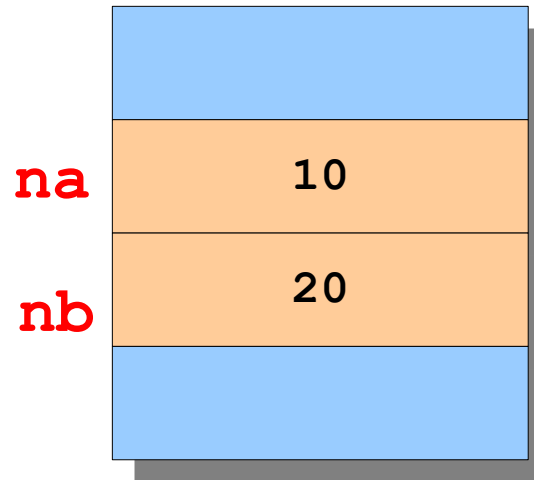
```
#include <stdio.h>
#include <stdlib.h>
```

```
main()
{
    int na,nb;
```

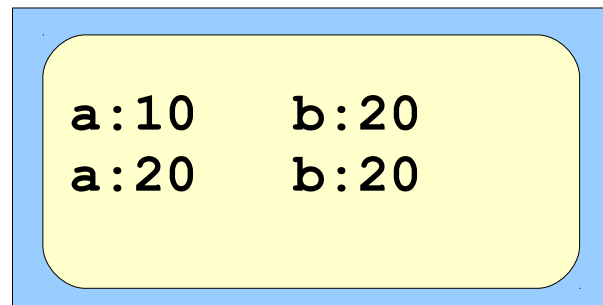
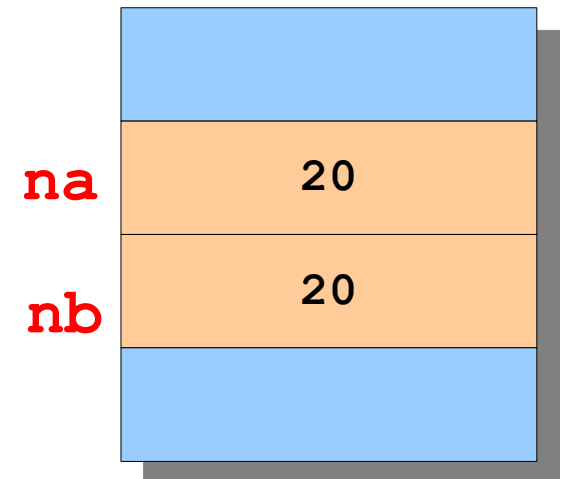
```
    na = 10;
    nb = 20;
    printf("a:%d    b:%d\n",na,nb) ;
```

```
    na = nb;
    printf("a:%d    b:%d\n",na,nb) ;
    system("pause") ;
}
```

Memória



Memória



# Calculando

```
#include <stdio.h>
#include <stdlib.h>
```

```
main()
{
  int parc1, parc2, r;
```

variável

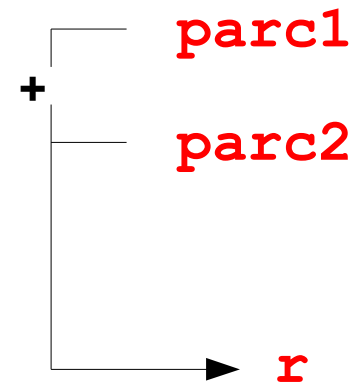
```
  parc1 = 12;
  parc2 = 15;
```

constante

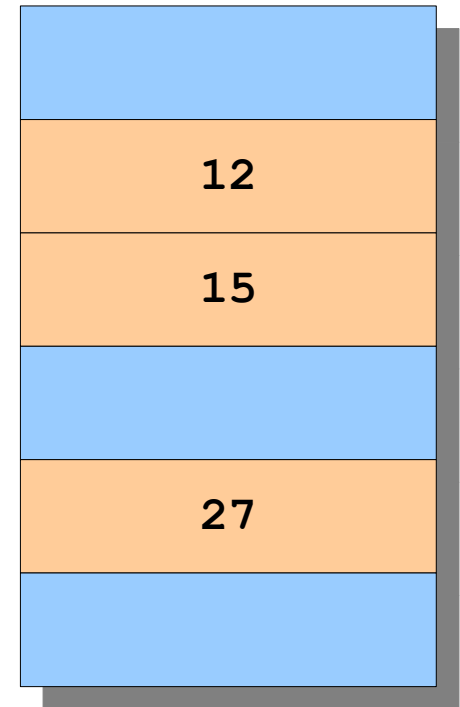
expressão

```
  r = parc1 + parc2;
```

```
  printf("soma:%d\n", r);
  system("pause");
}
```



Memória



soma: 27

# Definições

- **Constante**: É um valor fixo, invariante.
- **Variável**: É um espaço de memória que pode conter, a cada tempo, valores diferentes. Seu conteúdo é referenciado através de um nome (identificador). Devemos fornecer **nomes significativos** para as variáveis.
- **Expressões**: São representações simbólicas de operações a serem feitas sobre determinados operandos, visando a obtenção de um resultado.

# Atribuição

variável = constante;

Ex:  $a = 5;$

variável = variável;

Ex:  $b = c;$

variável = expressão;

Ex:  $d = a + 3;$

Atribuições inválidas:

$3 = a;$        $3 + b = a;$        $c - 5 = 9 + f;$

# Comando para entrada de dados

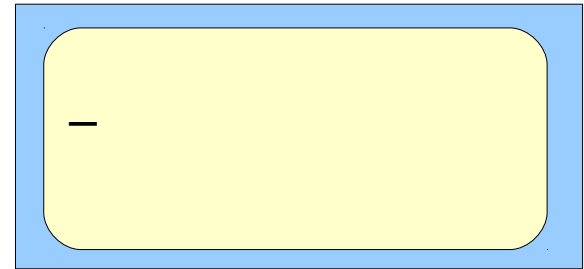
```
#include <stdio.h>
#include <stdlib.h>

main()
{
    int parc1, parc2, r;

    scanf ("%d", &parc1);
    scanf ("%d", &parc2);

    r = parc1 + parc2;

    printf ("Resultado: %d\n", r);
    system ("pause");
}
```



## scanf

Executa uma leitura do teclado e armazena o valor informado na variável especificada.



# Comando para entrada de dados

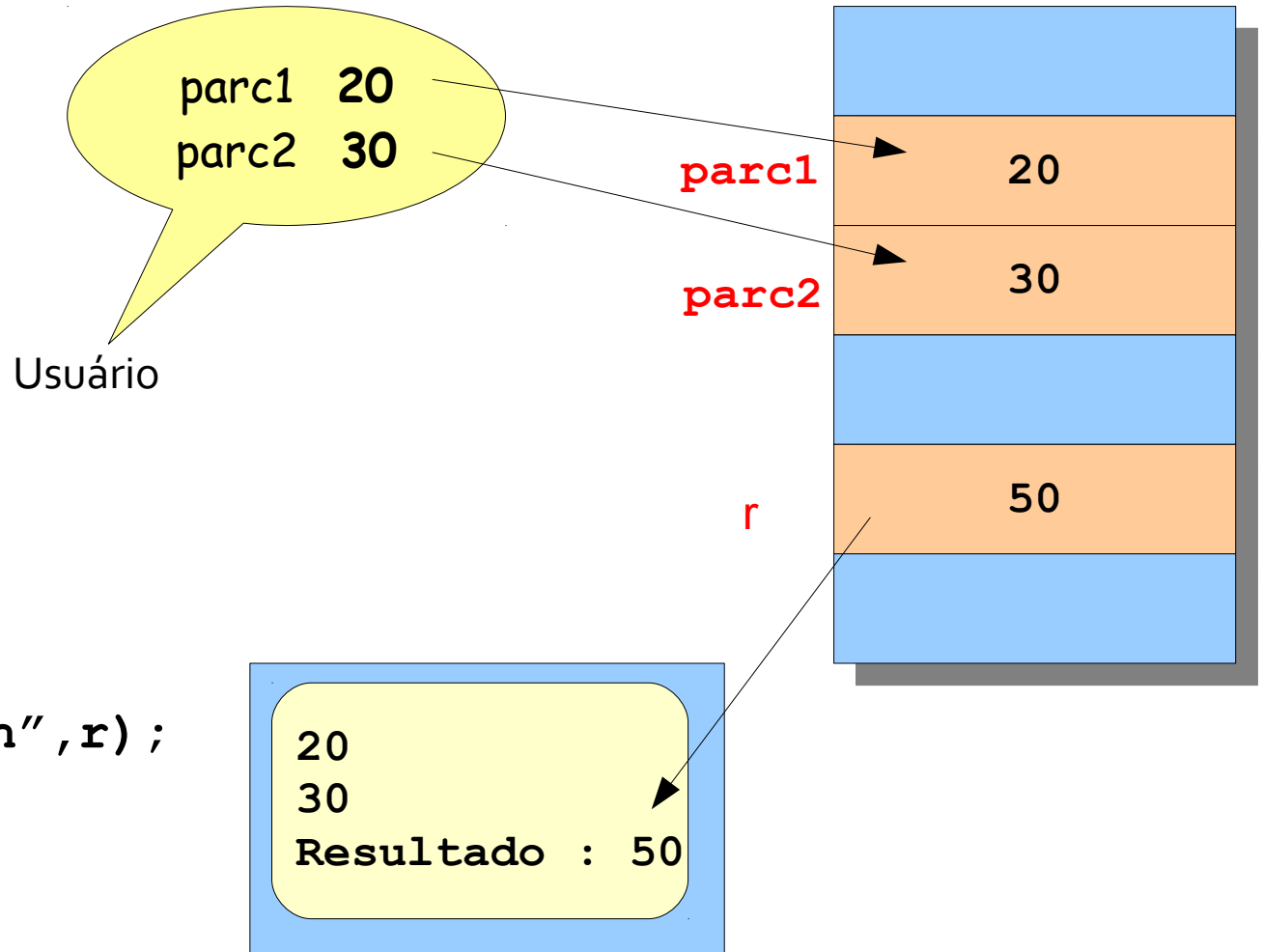
```
#include <stdio.h>
#include <stdlib.h>
```

```
main()
{
    int parc1, parc2, r;
```

```
    scanf ("%d", &parc1);
    scanf ("%d", &parc2);
```

```
    r = parc1 + parc2;
```

```
    printf ("Resultado: %d\n", r);
    system ("pause");
}
```



OBS: O valor das parcelas não estão fixas no programa.  
Ele pode ser utilizado para quaisquer dados de entrada.

# Inserindo mensagens junto às entradas de dados

```
#include <stdio.h>
#include <stdlib.h>
```

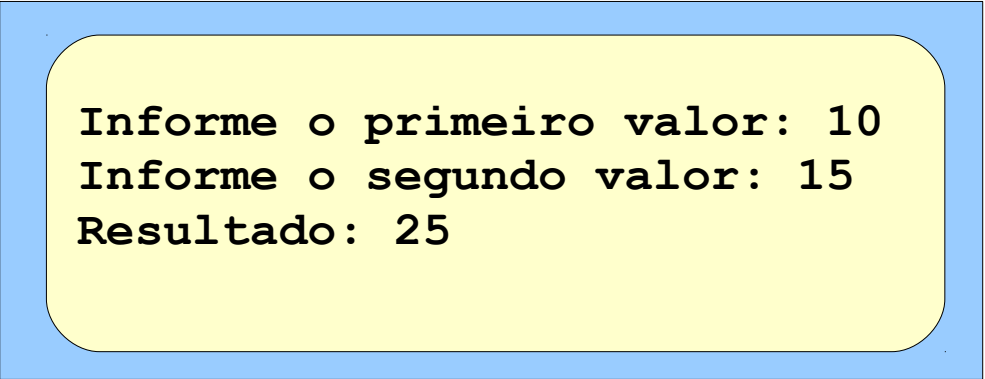
```
main()
{
    int parc1, parc2, r;
```

```
    printf("Informe o primeiro valor: ");
    scanf("%d", &parc1);
```

```
    printf("Informe o segundo valor: ");
    scanf("%d", &parc2);
```

```
    r = parc1 + parc2;
```

```
    printf("Resultado: %d\n", r);
    system("pause");
}
```



Informe o primeiro valor: 10  
Informe o segundo valor: 15  
Resultado: 25

# Permitindo a digitação de números com decimais

## float

Tipo de dados utilizado para armazenar valores que possuem casas decimais.

Código de formatação: **%f**

```
#include <stdio.h>
#include <stdlib.h>
```

```
main()
{
    float parc1, parc2, r;
```

```
    printf("Informe o primeiro valor: ");
    scanf("%f", &parc1);
```

```
    printf("Informe o segundo valor: ");
    scanf("%f", &parc2);
```

```
    r = parc1 + parc2;
```

```
    printf("Resultado: %f\n", r);
    system("pause");
}
```

# Operadores aritméticos

Operador	Operação	Prioridade
*	Multiplicação	1ª
/	Divisão	1ª
%	Resto da divisão	1ª
+	Adição	2ª
-	Subtração	2ª

## Expressões em C

(Determinar o valor atribuído às variáveis abaixo)

`a = 5 + 2 * 3;`

`b = (5 + 2) * 3;`

`c = 11 % 3;`

`d = ((5-3)*4 + 6) / 2;`

11	3
2	3

OBS: O operador % deve ser utilizado apenas com operandos inteiros (int)

# Escrevendo expressões em C

Matemática

em C

$$m = \frac{a+b}{2}$$

```
m = (a+b) / 2 ;
```

$$m = a + \frac{b}{2}$$

```
m = a + b / 2 ;
```

$$r = \left[ \frac{a}{b} - (d+e) \right] - 3$$

```
r = (a/b - (d+e)) - 3 ;
```

OBS: Para utilizar as funções matemáticas abaixo incluir o arquivo **math.h**

$$r = 5 + \sqrt{a+b}$$

```
r = 5 + sqrt(a+b) ;
```

$$x = (a+b)^2$$

```
x = pow(a+b, 2) ;
```

$$c = 2 \pi R$$

```
c = 2 * M_PI * r ;
```

# Regras para criar nomes de variáveis

- Deve começar com uma letra ou sublinhado (*underline*).
- O restante pode ser composto por letra, dígito ou sublinhado.
- Não é permitido a utilização de acentuação.
- Palavras reservadas da linguagem **não** podem ser utilizadas.

auto	default	float	register	struct	volatile
break	do	for	return	switch	while
case	double	goto	short	typedef	
char	else	if	signed	union	
const	enum	int	sizeof	unsigned	
continue	extern	long	static	void	

Exemplos de nomes válidos:

base area n1 num\_alunos ftotal iNumAlunos \_teste

Exemplos de nomes **inválidos**:

1a n\$ %aumento num-alunos área num alunos float

# Roteiro para a resolução dos problemas

- Ler o enunciado até compreender o problema.
- Identificar as saídas exigidas pelo problema.
- Identificar as entradas descritas no enunciado.
- Verificar as transformações necessárias para, dadas as entradas, produzir as saídas especificadas.
- Escrever o algoritmo.
- Testar cada passo, verificando se sua execução está conduzindo aos objetivos desejados.
- Escrever o programa.
- Testar o programa no computador.

# Problema resolvido

Escreva um programa para ler a resistência (Ohms) de 2 resistores associados em paralelo, calcular e escrever a resistência equivalente.

Exemplos de dados de entrada e suas respectivas saídas:

Exemplo 1	Exemplo 2	Exemplo 3
<b>Entrada</b>	<b>Entrada</b>	<b>Entrada</b>
200.00 (r1)	250.00 (r1)	300.00 (r1)
100.00 (r2)	120.00 (r2)	210.00 (r2)
<b>Saída</b>	<b>Saída</b>	<b>Saída</b>
66.66 (R. eq.)	81.08 (R. eq.)	123.52 (R. eq.)



# Problema resolvido

- Ler o enunciado e compreender o problema.
- Identificar as saídas exigidas pelo problema.

**Saída:** Resistência equivalente

- Identificar as entradas descritas pelo enunciado.

**Entradas:** Resistência de 2 resistores.

- Verificar as transformações necessárias para, dadas as entradas, produzir as saídas especificadas.

$$\text{Resistência equivalente} = \frac{1}{\frac{1}{R1} + \frac{1}{R2}}$$

# Programa em C

- Escrever o programa.

```
#include <stdio.h>
#include <stdlib.h>
```

```
main()
{
    float requiv,r1,r2;
```

Ler

```
    printf("Informe a resistência 1: ");
    scanf("%f",&r1);
```

```
    printf("Informe a resistência 2: ");
    scanf("%f",&r2);
```

Calcular

```
    requiv = 1 / (1/r1+1/r2);
```

Escrever

```
    printf("Resistência equivalente: %f\n",requiv);
    system("pause");
}
```