

Entrada e saída de dados
Prioridade aritmética
Operadores relacionais

Prof. Dr. Gerson Pastre de Oliveira



Saída de dados – função **printf**

- A linguagem C possui algumas funções que realizam saída de dados, sendo a função **printf** a mais completa e a mais usada
- Essa função está inserida na biblioteca **stdio.h** (*standard input and output library*) – no caso, a função **printf** envia dados para a saída padrão do dispositivo computacional (geralmente, um monitor de vídeo ou tela)

Função **printf** – sintaxes mais comuns

*printf (“<mensagem>”); // **exibe uma mensagem na tela***

Exemplo:

printf(“Aula de algoritmos e logica de programacao”);

Neste caso, a mensagem sairá na tela exatamente como foi escrita.

*Note que os parênteses e as aspas **sempre** devem ser usados nas diversas sintaxes do comando **printf***

Função **printf** – sintaxes mais comuns

```
printf ("<mensagem>%<tipo_dados>",<variável>);
```

Exemplo:

```
float sal = 1234.56;
```

```
printf("Valor do salário = %f",sal);
```

Neste caso, a mensagem sairá na tela exatamente como foi escrita até encontrar o %, pois, quando isso ocorre, deverá vir um caractere que designa um tipo de dados, que será correspondente a uma variável a ser impressa

Função **printf** – mais exemplos

Exemplo:

```
float sal1 = 1234.56, sal2 = 7890.12;
```

```
printf("Valor do salário 1 = %f, valor do salário 2 = %f",sal1, sal2);
```

Podem ser intercaladas várias mensagens e vários protótipos indicando a saída de dados por meio de uma lista de variáveis após a aspas e separadas por vírgulas – as variáveis devem ser colocadas na ordem em que se quer que apareçam na saída de dados, pois cada uma delas corresponde a um protótipo (no caso desse exemplo, a cada %f corresponde uma variável)

Função **printf** – mais exemplos

Exemplo:

```
float p = 156.10;
```

```
int cod = 100;
```

```
printf("Preço do produto cod %d = %f", cod, p);
```

Nesse caso, temos a saída de dois dados em meio à mensagem: um dado inteiro (o código do produto) e um dado de ponto flutuante (o preço do produto), nessa ordem

Principais protótipos e tipos de dados para os comandos **printf** e **scanf**

Protótipo	Tipo de dados
%d	int
%f	float
%lf	double
%c	caractere
%s	cadeia de caracteres (string)

Entrada de dados – função **scanf**

- A linguagem C possui algumas funções que realizam entrada de dados, sendo a função **scanf** a mais completa e a mais usada
- Essa função está inserida na biblioteca **stdio.h** (*standard input and output library*) – no caso, a função **scanf** permite a captura de dados desde a entrada padrão do dispositivo computacional (geralmente, um teclado)

Função **scanf** – sintaxes mais comuns

```
scanf ("%<tipo_dados>", &<variável>);
```

Exemplo:

```
float sal;
```

```
scanf ("%f", &sal);
```

*Neste caso, o controle do programa gera uma interrupção para o teclado (tratada pelo sistema operacional) e aguarda a digitação de um dado que será armazenado na variável de ponto flutuante **sal**. O **e comercial** (&) faz parte da sintaxe (menos com o protótipo %s) e significa “endereço da variável na memória”. É preciso atentar que para o fato de que o comando **scanf** não exibe mensagens na tela, sendo comum usá-lo em conjunto com o **printf**.*

Função **scanf** – sintaxes mais comuns

```
scanf ("%<tipo_dados>%<tipo_dados>", &<variável_1>, &<variável_2>);
```

Exemplo:

```
float sal; int idade; char opcao;
```

```
scanf ("%f%d%c", &sal, &idade, &opcao);
```

*O comando **scanf** pode ser usado com uma lista de entradas, que são feitas na sequência em que as variáveis aparecem no comando. Essa opção não é muito usada, porque nenhuma mensagem para o usuário é inserida entre as diversas entradas.*

Operadores aritméticos

Operador	Símbolo
Adição	+
Subtração	-
Multiplicação	*
Divisão	/
Resto de divisão	%

Operadores aritméticos

- As linguagens de programação (e a linguagem C , em particular) seguem uma ordem de prioridade aritmética para os operadores, o que significa que alguns operadores são avaliados antes dos outros em uma expressão aritmética, o que concorre para determinar o resultado correto da expressão

Prioridade aritmética

1. Multiplicação $*$, divisão $/$, resto de divisão $\%$
 2. Adição $+$, subtração $-$
- Se, em uma mesma operação de atribuição, aparecerem operadores de mesma prioridade, são resolvidos antes os que estiverem mais à esquerda

Prioridade aritmética

```
#include<stdio.h>
main()
{
    int a = 5, b = 10, c = 15, d, e;

    d = a + b * 2 / 4 - 1;
    printf("Valor de d = %d",d);

    e = c % 3 - d * b / 2 + a;
    printf("\nValor de e = %d",e);

    a = a - 2;
    b = b + 2;
    c = c / 3;

    int f = a + b / 6 % 2 - 1;
    printf("\nValor de f = %d",f);
}
```

```
Valor de d = 9
Valor de e = -40
Valor de f = 2
```

```
-----
Process exited after 0.5745 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

Prioridade aritmética

Parênteses mudam a prioridade aritmética: as operações dentro de parênteses são realizadas antes

```
#include<stdio.h>
main()
{
    int a = 5, b = 10, c = 15, d, e;

    d = (a + b) * 2 / (4 - 1);
    printf("Valor de d = %d",d);

    e = c % (d - 3) * b / 2 + a;
    printf("\nValor de e = %d",e);

    a = a - 2;
    b = b + 2;
    c = c / 3;

    int f = (a + b) * 6 - (4 + 1);
    printf("\nValor de f = %d",f);
}
```

```
Valor de d = 10
Valor de e = 10
Valor de f = 85
```

```
-----
Process exited after 0.2127 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Prioridade aritmética

```
#include<stdio.h>
main()
{
    int a = 5, b = 10, c = 15;

    a = (a * (c - b) + 7) / 2;

    printf("\nValor de a = %d",a);
}
```

Parênteses mudam a prioridade aritmética:
as operações dentro de parênteses são realizadas antes

```
Valor de a = 16
-----
Process exited after 0.2052 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```


Troca de variáveis

- Dadas duas variáveis, **x** e **y**, com valores digitados pelo usuário, inverter seus valores (o valor de **x** vai para **y** e o valor de **y** vai para **x**)

```
#include<stdio.h>
main()
{
    int x, y;
    printf("\nDigite x = ");
    scanf("%d",&x);
    printf("\nDigite y = ");
    scanf("%d",&y);
    x = y;
    y = x;
    printf("\nSaida de x = %d",x);
    printf("\nSaida de y = %d",y);
}
```

Digite x = 4

Digite y = 5


Saida de x = 5

Saida de y = 5

Process exited after 2.283 seconds with return value 0

Press any key to continue. . .

```
#include<stdio.h>
main()
{
    int x, y, aux;
    printf("\nDigite x = ");
    scanf("%d",&x);
    printf("\nDigite y = ");
    scanf("%d",&y);
    aux = x;
    x = y;
    y = aux;
    printf("\nSaida de x = %d",x);
    printf("\nSaida de y = %d",y);
}
```



```

Digite x = 5
Digite y = 4
Saida de x = 5
Saida de y = 4
-----
Process exited after 2.54 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

Operadores relacionais

Operador	Símbolo	“Contrário”
Maior que	>	<=
Menor que	<	>=
Maior ou igual a	>=	<
Menor ou igual a	<=	>
Igual	==	!=
Diferente	!=	==

Operadores relacionais

- O resultado de uma expressão envolvendo operadores relacionais é um valor lógico (*verdadeiro* ou *falso*) – isso é muito importante!
- Por exemplo, a expressão $5 > 3$ tem o valor lógico *verdadeiro*
- Já a expressão $5 < 3$ tem o valor lógico *falso*
- O valor lógico *falso* pode ser representado por 0 (zero) e o valor lógico *verdadeiro* pode ser representado por 1 (um)
- Chamamos uma expressão relacional de **teste lógico** ou **condição**

Operadores relacionais

```
#include<stdio.h>
main()
{
    printf("\n5>3   : %d",5>3);
    printf("\n5<3   : %d",5<3);
    printf("\n3==3  : %d",3==3);
    printf("\n3!=3   : %d",3!=3);
    printf("\n7>=5   : %d",7>=5);
    printf("\n7<=5   : %d",7<=5);
}
```

```
5>3   : 1
5<3   : 0
3==3  : 1
3!=3   : 0
7>=5   : 1
7<=5   : 0
```

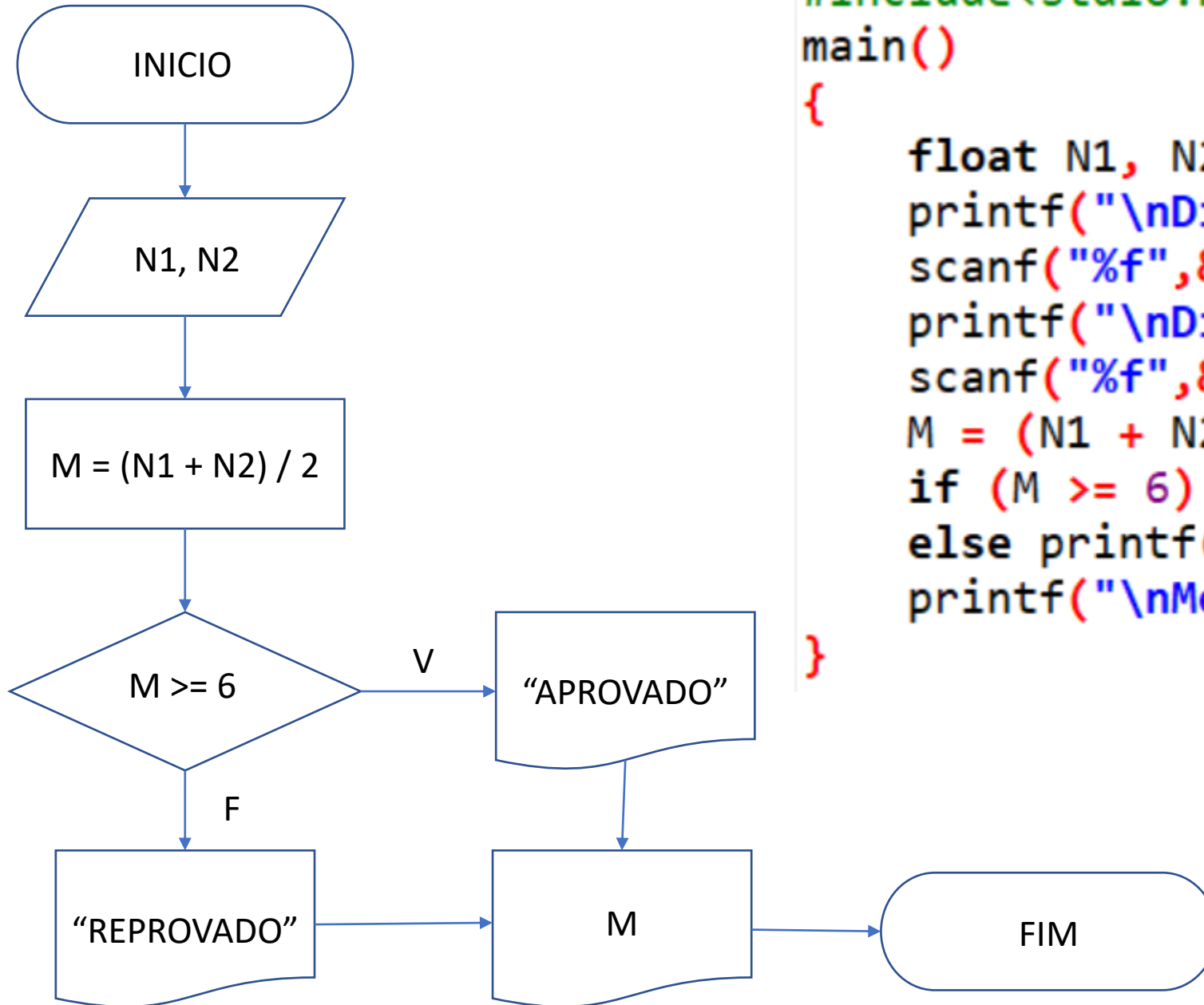
```
-----
Process exited after 0.1904 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Outro tipo de problema

- Em uma escola, um aluno faz duas provas e precisa ter a média aritmética calculada em relação às duas notas obtidas. Caso esse aluno obtenha média maior ou igual a 6, estará aprovado; caso contrário, estará reprovado. Construa um algoritmo que permita a entrada das duas notas, o cálculo da média, a exibição da média e a exibição do resultado (aprovado ou reprovado)

Outro tipo de problema

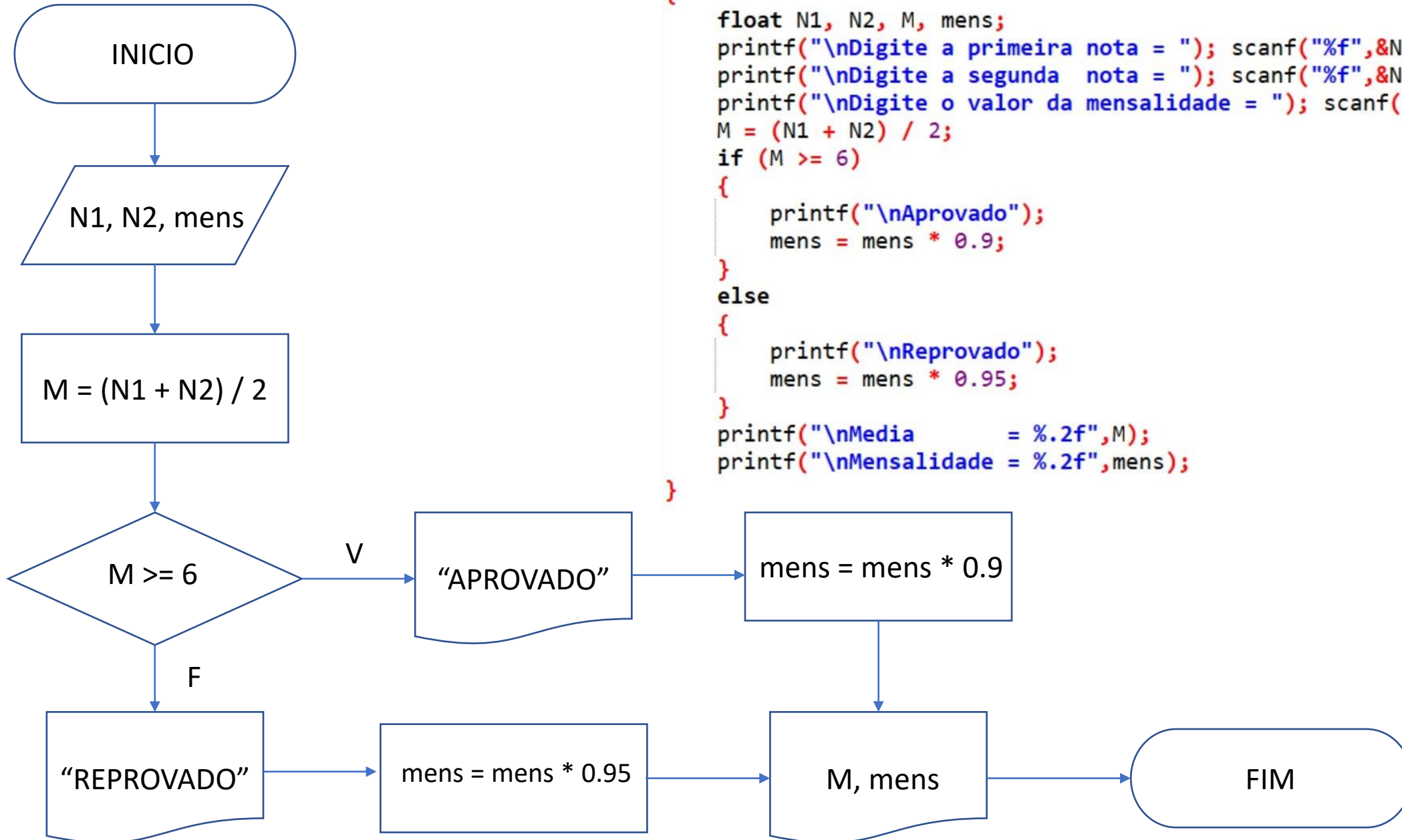
- Aprovado \rightarrow média ≥ 6
- Reprovado \rightarrow média < 6 (o contrário)...



```
#include<stdio.h>
main()
{
    float N1, N2, M, mens;
    printf("\nDigite a primeira nota = ");
    scanf("%f",&N1);
    printf("\nDigite a segunda nota = ");
    scanf("%f",&N2);
    M = (N1 + N2) / 2;
    if (M >= 6) printf("\nAprovado");
    else printf("\nReprovado");
    printf("\nMedia = %.2f",M);
}
```

Outro tipo de problema... de novo

- Agora, suponha que, além de mostrar se o aluno foi aprovado e reprovado, também deverá ser mostrado o valor da mensalidade a ser paga: no caso de aprovação, o aluno deverá ter um desconto de 10% sobre o valor informado pelo usuário; caso contrário, terá desconto de 5%.



```
#include<stdio.h>
main()
{
    float N1, N2, M, mens;
    printf("\nDigite a primeira nota = "); scanf("%f",&N1);
    printf("\nDigite a segunda nota = "); scanf("%f",&N2);
    printf("\nDigite o valor da mensalidade = "); scanf("%f",&mens);
    M = (N1 + N2) / 2;
    if (M >= 6)
    {
        printf("\nAprovado");
        mens = mens * 0.9;
    }
    else
    {
        printf("\nReprovado");
        mens = mens * 0.95;
    }
    printf("\nMedia      = %.2f",M);
    printf("\nMensalidade = %.2f",mens);
}
```

Exercícios

1. Um funcionário recebe um salário mensal em sua empresa. Caso esse salário seja maior que 5000, ele deverá receber um aumento de 12%; caso contrário, o aumento será de 15%. Mostre o salário com aumento.
2. Um produto tem um desconto sobre seu preço, considerando o dia do mês em que foi comprado: se for na primeira quinzena, o desconto é de 8%; se for na segunda quinzena, o desconto é de 6%. Mostre o preço do produto com desconto.
3. Calcule e mostre as raízes reais de uma função polinomial de segundo grau.