

Subalgoritmos com passagem de parâmetros

Subalgoritmos



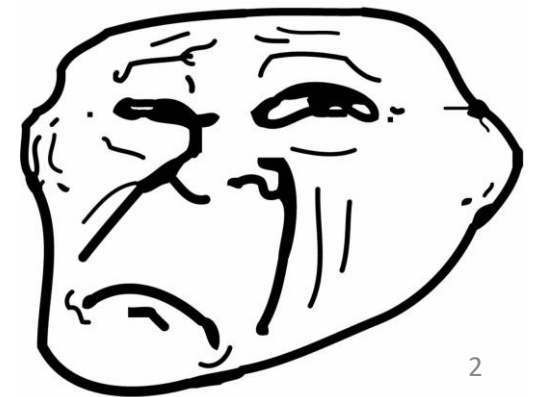
PROBLEMA: Escreva um programa para exibir a seguinte tela. Cada linha de asteriscos deve ser impressa com uma chamada à função `linhaDeAsteriscos()`.

```
*****
```

```
IFSUL
```

```
*****
```

Como permitir que a função `linhaDeAsteriscos` exiba uma quantidade **qualquer** de asteriscos?





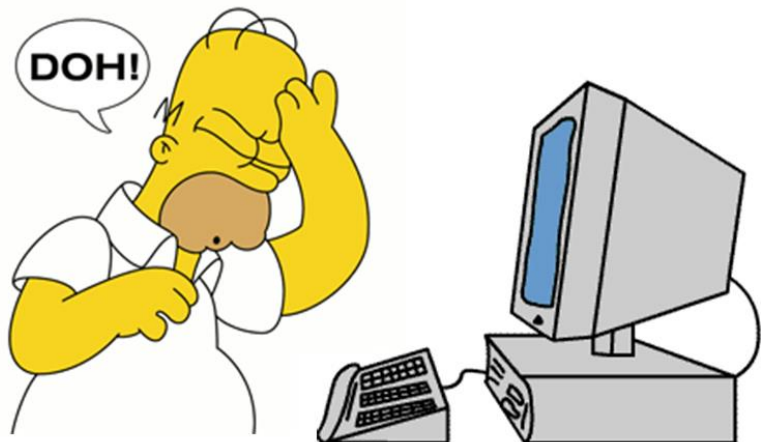
Esse programa não compila. Por que?

Declaramos o a variável **n** no algoritmo principal, e estamos tentando acessá-la no subalgoritmo.

```
#include <stdio.h>
void linhaDeAsteriscos(void);
main(){
    int n;
    n=5;
    linhaDeAsteriscos();
    printf("\nIFSUL\n");
    n=22;
    linhaDeAsteriscos();
}
```

```
void linhaDeAsteriscos(void){
    int i;

    for (i=1; i<=n; i++)
        printf("*");
}
```





Esse programa compila, mas
não funciona... Por que?

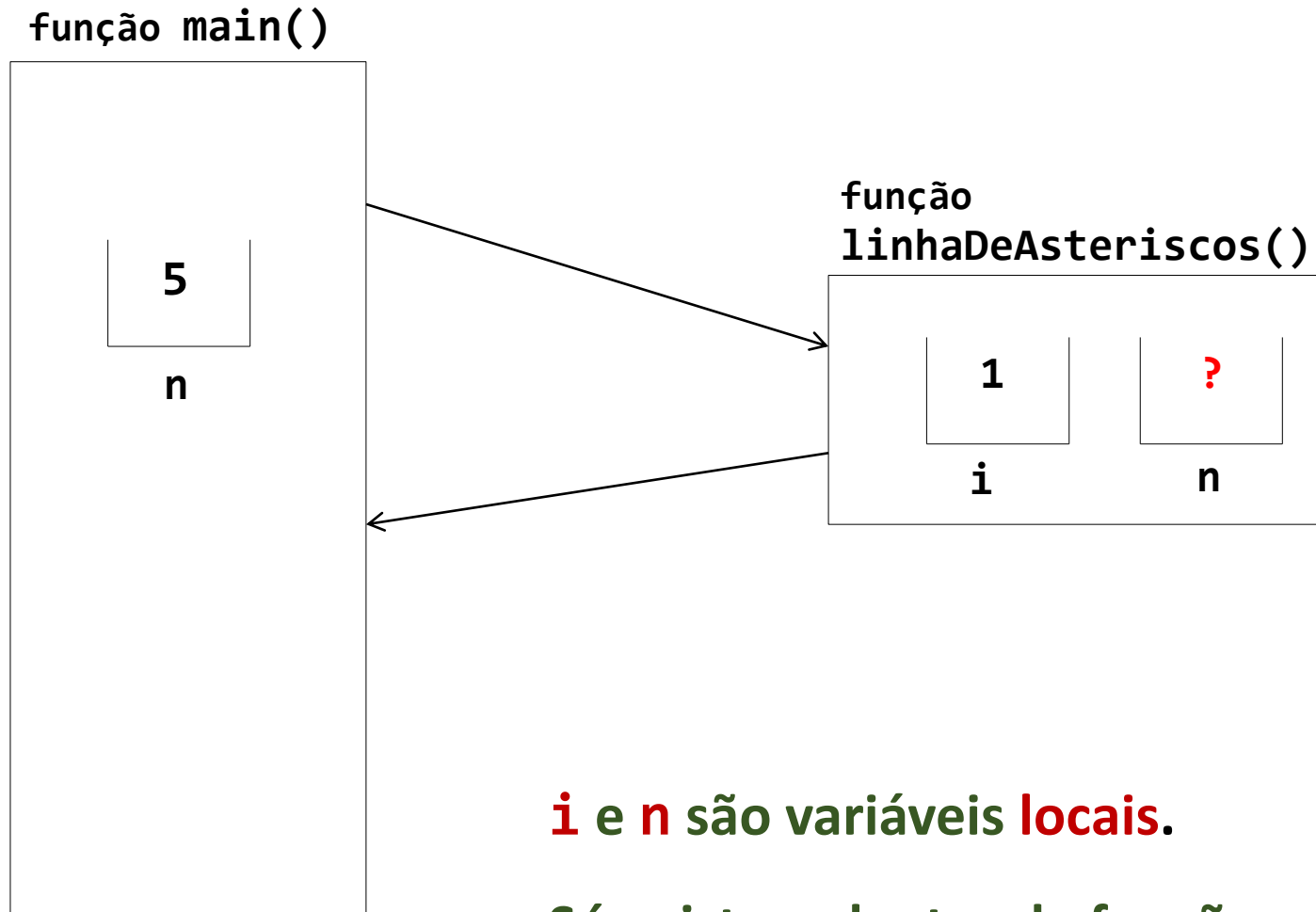
Estamos usando uma variável **n** no
escopo local da função **main()** e outra
variável **n** no escopo do subalgoritmo.

```
#include <stdio.h>
void linhaDeAsteriscos(void);
main(){
    int n;

    n=5;
    linhaDeAsteriscos();
    printf("\nIFSUL\n");
    n=22;
    linhaDeAsteriscos();
}
```

```
void linhaDeAsteriscos(void){
    int i, n;

    for (i=1; i<=n; i++)
        printf("*");
}
```



i e **n** são variáveis **locais**.

Só existem dentro da função onde foram declaradas.



Como quebrar o isolamento de variáveis?

Funções com Passagem de parâmetros

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

```
void linhaDeAsteriscos(int n);
```

Declarando parâmetros de entrada.

```
main(){
```

```
    linhaDeAsteriscos(5);
```

```
    printf("\nIFSUL\n");
```

```
    linhaDeAsteriscos(22);
```

```
}
```

```
void linhaDeAsteriscos(int n){
```

```
    int i;
```

```
    for (i=1; i<=n; i++)
```

```
        printf("*");
```

```
}
```

Argumento

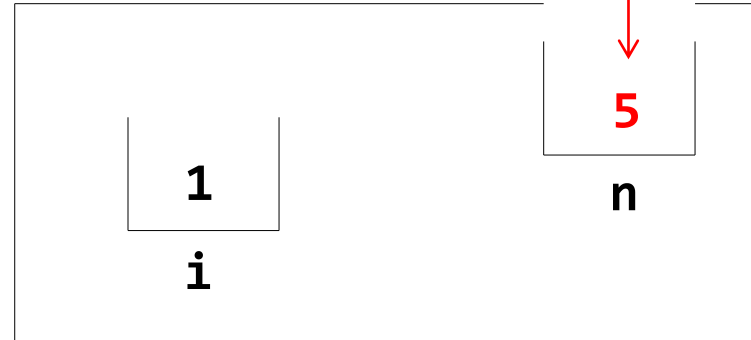
Parâmetro



Passagem de parâmetros

```
main(){  
...  
  linhaDeAsteriscos(5);  
...  
}
```

linhaDeAsteriscos(int n)



O argumento **5** é passado para o parâmetro **n** declarado na função **linhaDeAsteriscos**.

A variável **n** continua existindo apenas **na função onde ela foi declarada**.

Passagem de parâmetros



Escreva um programa para exibir a seguinte tela. Cada linha de asteriscos deve ser impressa com uma chamada à função **linhaDeAsteriscos()**.

```
*  
  
**  
  
***  
  
****  
  
*****
```




Passagem de parâmetros

Uma variável pode ser utilizada como argumento. O **valor** da variável **a** é copiado para a variável **n**.

A variável **a** continua existindo apenas na função onde ela foi declarada.

```
#include <stdio.h>
void linhaDeAsteriscos(int n);
main(){
    int a;
    for (a=1; a<=5; a++){
        linhaDeAsteriscos(a);
        printf("\n");
    }
}

void linhaDeAsteriscos(int n){
    int i;
    for (i=1; i<=n; i++)
        printf("*");
}
```

Argumento

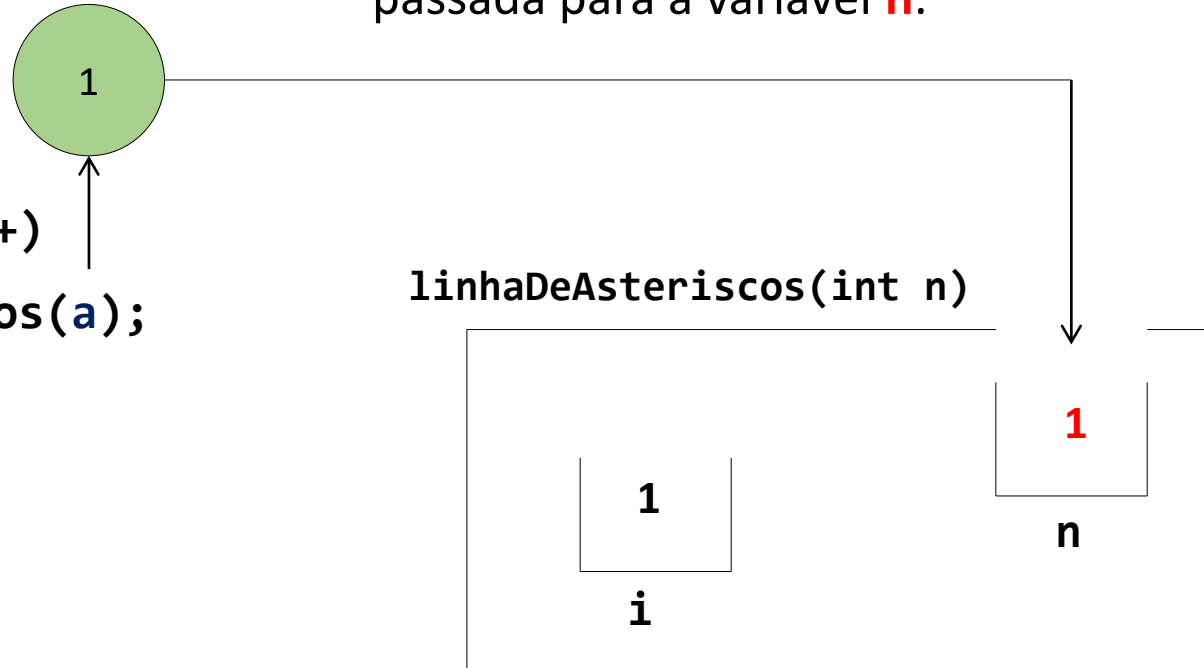
Parâmetro



Subalgoritmos

```
main(){  
    int a;  
  
    for (a=1; a<=5; a++)  
        linhaDeAsteriscos(a);  
  
    ...
```

Uma **cópia** da variável **a** é passada para a variável **n**.



Uma **expressão** também pode ser usada como argumento.

```
...  
linhaDeAsteriscos(a+4);  
...
```



```
#include <stdio.h>
void alteraNumero(int n);

main(){
    int num;
    num = 30;
    printf("Número (main) antes: %d\n", num);
    alteraNumero(num);
    printf("Número (main) depois: %d\n", num);
}

void alteraNumero(int n){
    n = 40;
    printf("Número (função): %d\n", n);
}
```

Número (main) antes: 30

Número (função): 40

Número (main) depois: 30



```
#include <stdio.h>

void alteraNumero(int num);

main(){
    int num;
    num = 30;
    printf("Número (main) antes: %d\n", num);
    alteraNumero(num);
    printf("Número (main) depois: %d\n", num);
}

void alteraNumero(int num){
    num = 40;
    printf("Número (função): %d\n", num);
}
```

Número (main) antes: 30

Número (função): 40

Número (main) depois: 30

Parâmetros



Parâmetros formais

São as **variáveis declaradas no cabeçalho das funções**. Dentro de uma função trabalha-se com estas variáveis da mesma maneira como se trabalha com variáveis locais.

Parâmetros reais (argumentos)

São aqueles que **substituem os parâmetros formais** quando ocorre a **chamada de uma função (argumentos)**.



=====
IFSUL

```
#include <stdio.h>
void linhaDeCaracteres(int n, char ch);
```

```
main(){
    linhaDeCaracteres(5, '=');
    printf("\nIFSUL\n");
    linhaDeCaracteres(15, '-');
}
```

Parâmetros reais

```
void linhaDeCaracteres(int n, char ch){
    int i;
    for (i=1; i<=n; i++)
        printf("%c",ch);
}
```

Parâmetros formais

A **ordem dos argumentos**
determina a **relação entre os**
parâmetros reais e formais.



Indica que a função não possui parâmetros

```
#include <stdio.h>

void linhaDezAsteriscos(void);
void exibeSoma(float a, float b);

main(){
    float x,y;

    printf("Informe um valor:");
    scanf("%f",&x);
    printf("Informe outro:");
    scanf("%f",&y);

    linhaDezAsteriscos();
    printf("\n");
    exibeSoma(x,y);
    linhaDeAsteriscos();
}
```

```
void linhaDezAsteriscos(void){
    int i;
    for (i=1; i<=10; i++)
        printf("*");
}
```

Os tipos devem ser declarados de forma independente mesmo para parâmetros de mesmo tipo

```
void exibeSoma(float a, float b){
    float soma;
    soma = a+b;
    printf("Soma: %f\n",soma);
}
```

Subalgoritmos com passagem de parâmetros