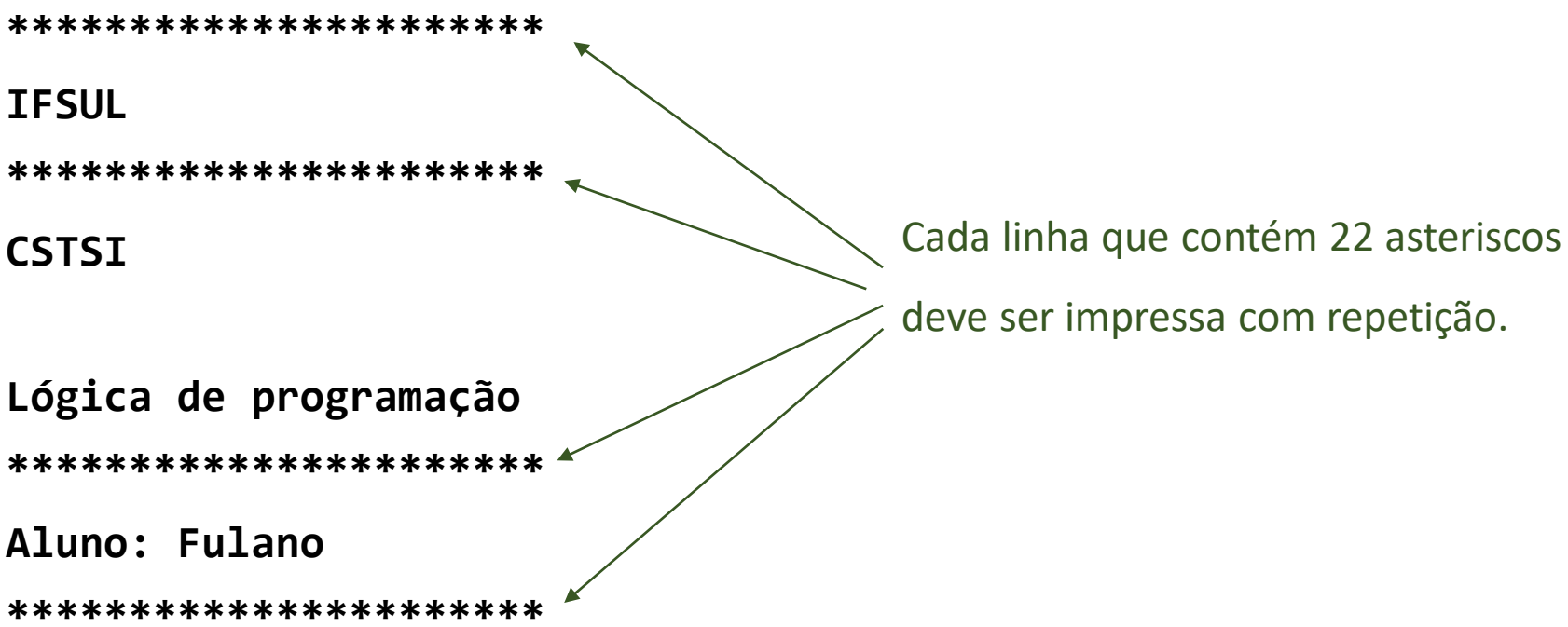
# Subalgoritmos

## Aula 20

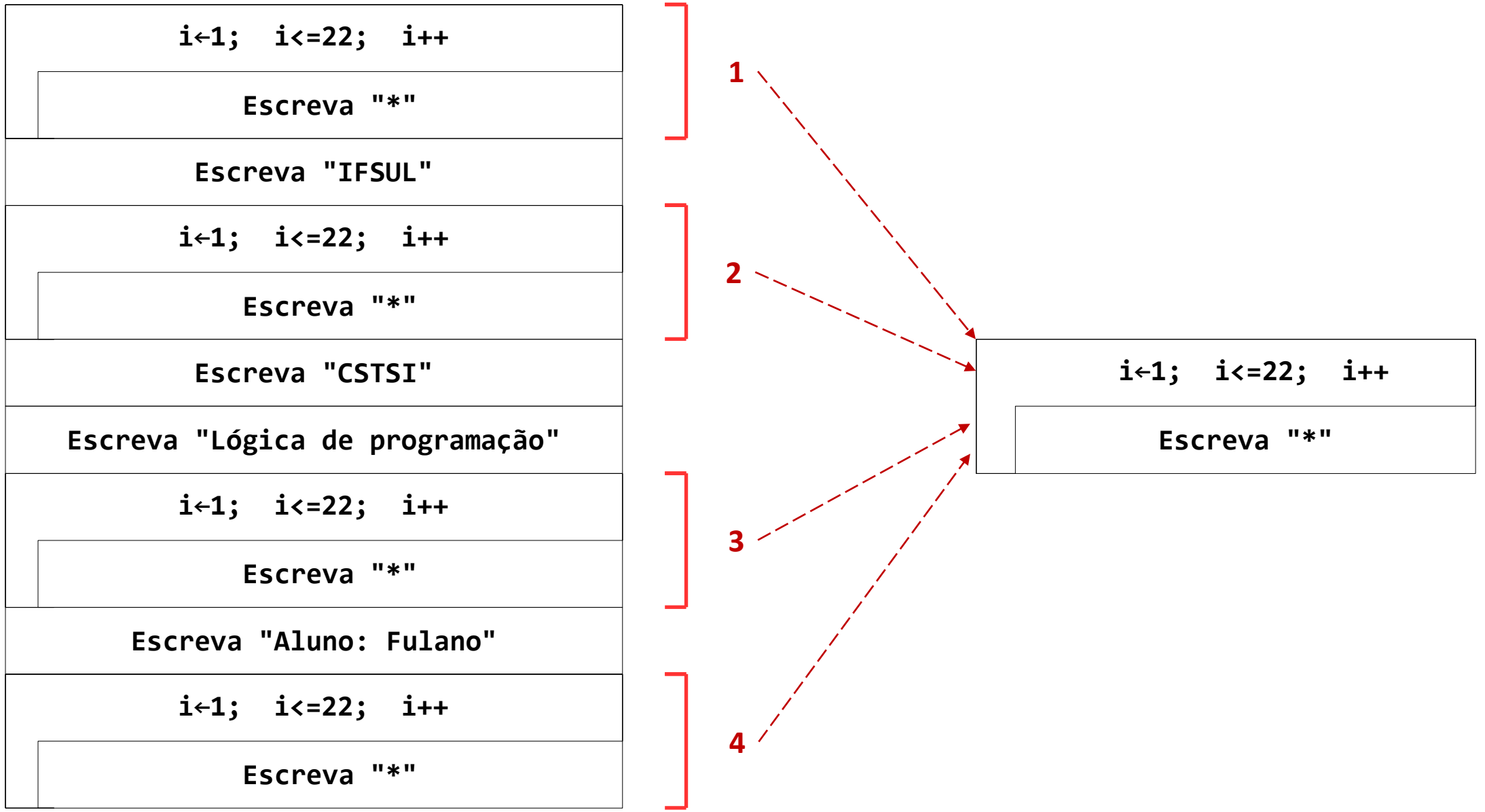
INSTITUTO FEDERAL
Sul-rio-grandense
Câmpus Pelotas

# Subalgoritmos

**PROBLEMA:** Escreva um algoritmo para exibir a seguinte tela.



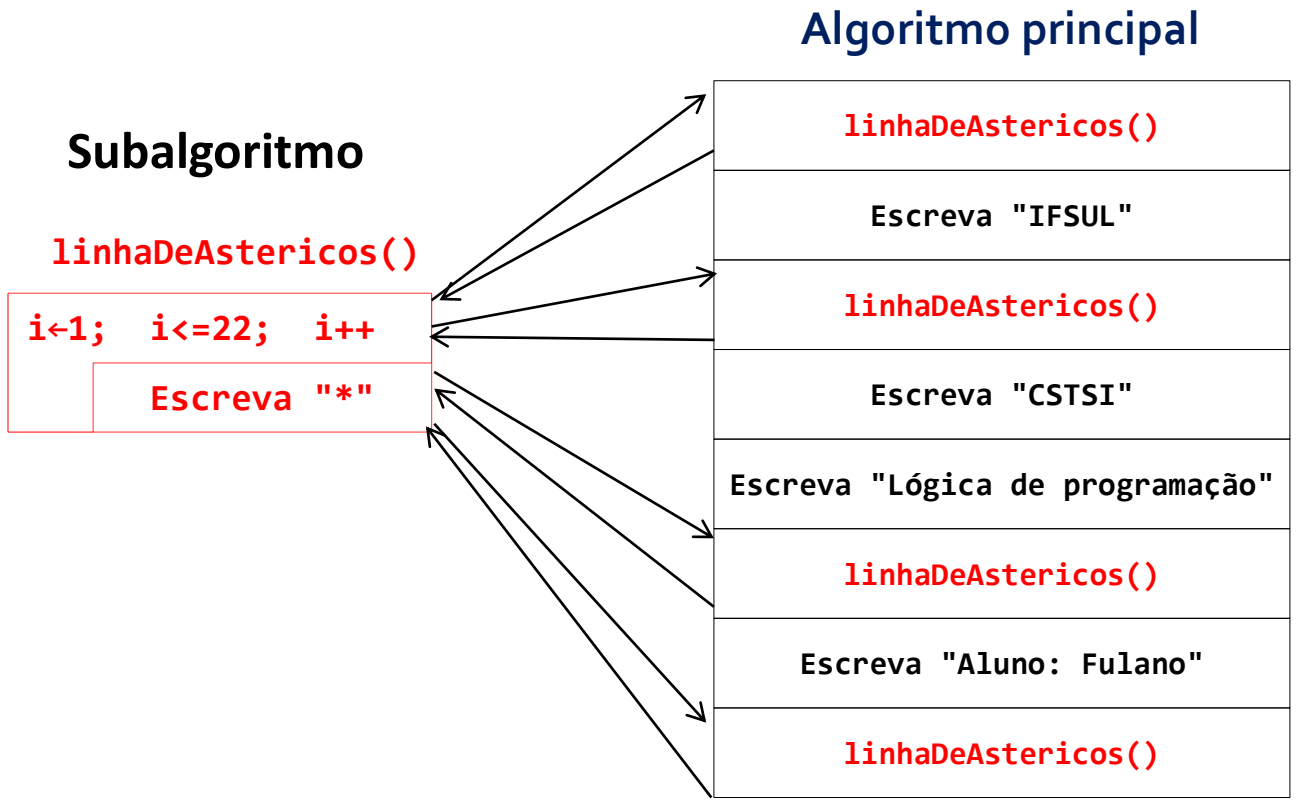
# Solução 1


i←1; i<=22; i++
Escreva "*"
Escreva "IFSUL"
i←1; i<=22; i++
Escreva "*"
Escreva "CSTSI"
Escreva "Lógica de programação"
i←1; i<=22; i++
Escreva "*"
Escreva "Aluno: Fulano"
i←1; i<=22; i++
Escreva "*"
1
2
3
4
i←1; i<=22; i++
Escreva "*"

# Subalgoritmos

É um trecho de algoritmo, com uma **função bem definida** (o mais **independente** possível do restante do algoritmo) que é **chamado dentro de um algoritmo principal**.

```
i←1;  i<=22;  i++
    Escreva "*"
Escreva "IFSUL"
i←1;  i<=22;  i++
    Escreva "*"
Escreva "CSTSI"
Escreva "Lógica de programação"
i←1;  i<=22;  i++
    Escreva "*"
Escreva "Aluno: Fulano"
i←1;  i<=22;  i++
    Escreva "*"
```



# Vantagens dos Subalgoritmos


- ✓ **Evita** que um trecho de comandos necessário em vários locais tenha que ser **escrito repetidamente**;
- ✓ **Divide e estrutura** um algoritmo em partes logicamente coerentes;
- ✓ Aumenta a **legibilidade** do algoritmo;
- ✓ Facilita a **detecção de erros**;
- ✓ Permite a **reutilização de software**; e
- ✓ **Divide** um problema grande em vários menores.

# Em “C” são chamados de funções

```
#include <stdio.h>

void linhaDeAsteriscos(void);

main(){
   linhaDeAsteriscos();
   printf("\nIFSUL\n");
   linhaDeAsteriscos();
   printf("\nCSTSI\n\n");
   printf("Lógica de programação\n");
   linhaDeAsteriscos();
   printf("\nAluno: Fulano\n");
   linhaDeAsteriscos();
}
```

Protótipo da função

Programa principal (função principal)

Chamadas da função

Retorno da função

Nome da função

Parâmetros da função

Cabeçalho da função

```
void linhaDeAsteriscos(void){
int i;
for (i=1; i<=22; i++)
  printf("*");
}
```

Variável local

Definição da função (corpo)

**PROBLEMA:** Escreva um programa para ler um inteiro Q e imprimir Q linhas de 22 asteriscos.

## Variáveis locais:

- São visíveis apenas no local onde são declaradas.
- São criadas quando a execução da função inicia e destruídas quando termina.

```
#include <stdio.h>
void linhaDeAsteriscos(void);

main(){
  int i, q;
  printf("Informe Q:");
  scanf("%d", &q);
  for (i=1; i<=q; i++){
    linhaDeAsteriscos();
    printf("\n");
  }
}

void linhaDeAsteriscos(void){
int i;

for (i=1; i<=22; i++)
  printf("*");
}
```

Variáveis locais

Variável local

# Variáveis locais

```
#include <stdio.h>
void exibeNumero(void);
main(){
  int num;
  num = 30;
  exibeNumero();
  printf("Número (main):%d\n", num);
}

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

Esse programa não compila.

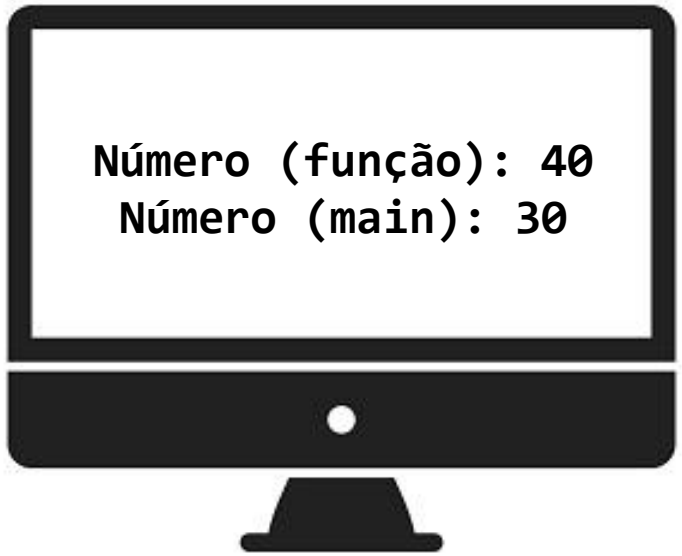
Por que??

**Variáveis locais estão isoladas** dentro da função onde foram declaradas.

# Variáveis locais

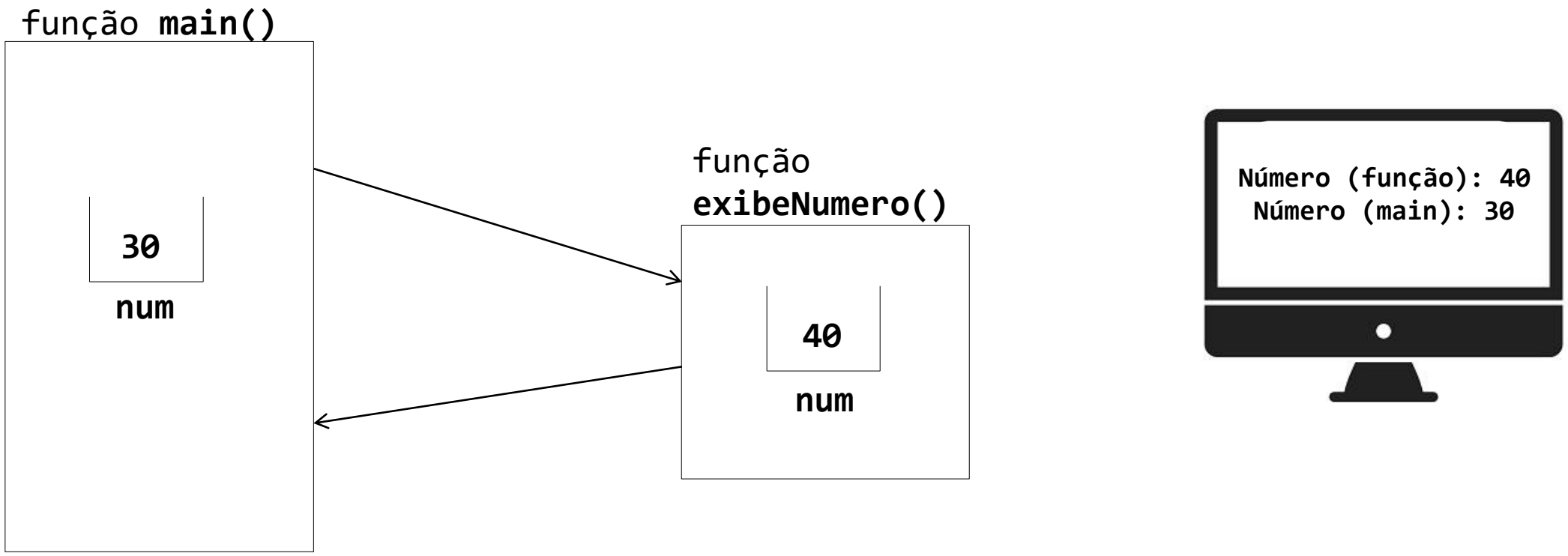
```c
#include <stdio.h>
void exibeNumero(void);
main(){
  int num;
  num = 30;
  exibeNumero();
  printf("Número (main):%d\n", num);
}

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



# Variáveis locais

"São duas variáveis (num) diferentes com o mesmo nome"



# Problema

**Escreva um programa para exibir a tela abaixo.**

O programa deve implementar uma função chamada **`retAsteriscos()`** que exibe na tela um retângulo com 3 linhas e 22 colunas de asteriscos.

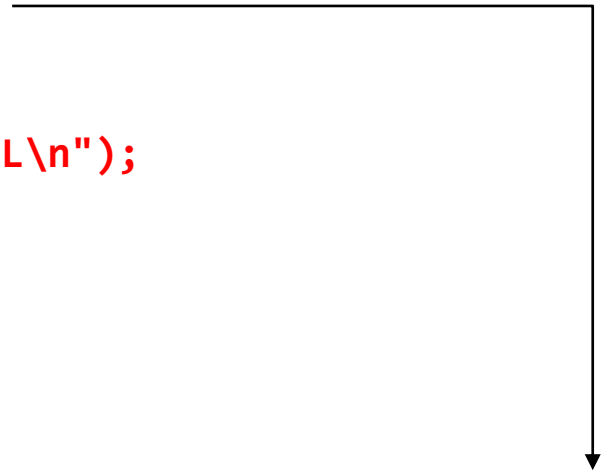
```
**********************
**********************
**********************
          IFSUL
**********************
**********************
**********************
```

# Uma função chamando outra função

```
#include <stdio.h>
void linhaDeAsteriscos(void);
void retAsteriscos(void);

main(){
  retAsteriscos();

  printf("\n     IFSUL\n");
  retAsteriscos();
}
```



```
void linhaDeAsteriscos(){
int i;
for (i=1; i<=22; i++)
  printf("*");
}
```

```
void retAsteriscos(){
  int i;
  for (i=1; i<=3; i++){
    linhaDeAsteriscos();

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

# Uma função chamando outra função

```
#include <stdio.h>
void linhaDeAsteriscos(void);
void retAsteriscos(void);

main(){
  retAsteriscos();

  printf("\n     IFSUL\n");
  retAsteriscos();
}
```

```
void linhaDeAsteriscos(){
int i;
for (i=1; i<=22; i++)
  printf("*");
}
```

```
void retAsteriscos(){
  int i;
  for (i=1; i<=3; i++){
    linhaDeAsteriscos();

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

# Uma função chamando outra função

```
#include <stdio.h>
void linhaDeAsteriscos(void);
void retAsteriscos(void);

main(){
  retAsteriscos();

  printf("\n     IFSUL\n");
  retAsteriscos();
}
```

```
void linhaDeAsteriscos(){
int i;
for (i=1; i<=22; i++)
  printf("*");
}
```

```
void retAsteriscos(){
  int i;
  for (i=1; i<=3; i++){
    linhaDeAsteriscos();

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

# Uma função chamando outra função

```
#include <stdio.h>
void linhaDeAsteriscos(void);
void retAsteriscos(void);

main(){
  retAsteriscos();

  printf("\n     IFSUL\n");
  retAsteriscos();
}
```

```
void linhaDeAsteriscos(){
int i;
for (i=1; i<=22; i++)
  printf("*");
}
```

```
void retAsteriscos(){
  int i;
  for (i=1; i<=3; i++){
    linhaDeAsteriscos();

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

# Uma função chamando outra função

```
#include <stdio.h>
void linhaDeAsteriscos(void);
void retAsteriscos(void);

main(){
  retAsteriscos();

  printf("\n     IFSUL\n");
  retAsteriscos();
}
```

```
void linhaDeAsteriscos(){
int i;
for (i=1; i<=22; i++)
  printf("*");
}
```

```
void retAsteriscos(){
  int i;
  for (i=1; i<=3; i++){
    linhaDeAsteriscos();

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

# Subalgoritmos

## Aula 20

INSTITUTO FEDERAL
Sul-rio-grandense
Câmpus Pelotas