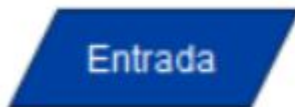
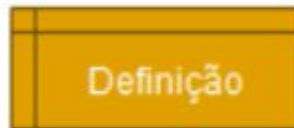


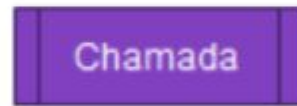
Entrada / Saída



Variáveis



Controle



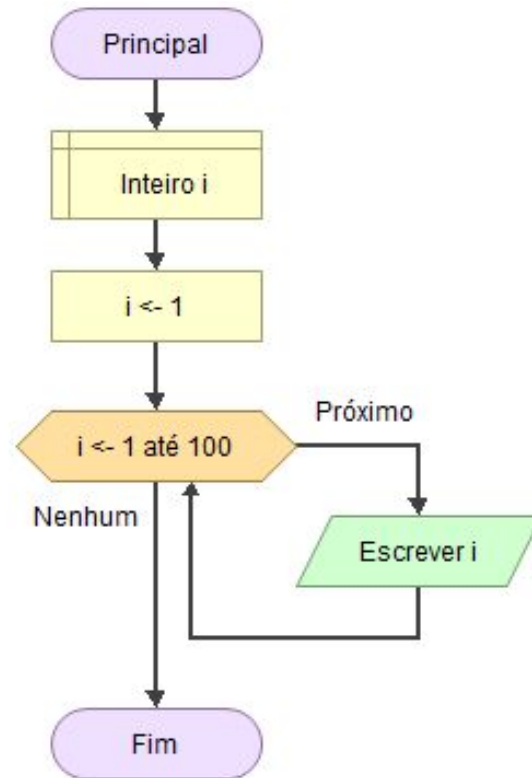
Repetição



Tipos de Loops (iterativos e Interativos)

Nos tipos de laço for, você geralmente define tudo logo no início: a inicialização, a condição de parada e qual atualização a fazer.

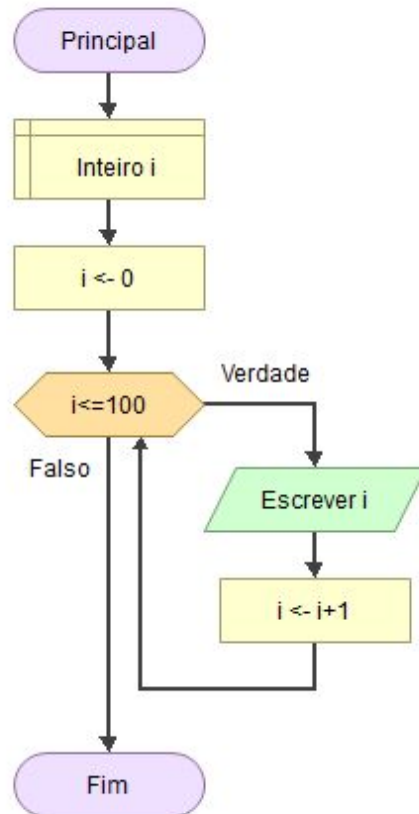
```
for (int i = 0; i < 100; i = i + 1)
    printf("%d\n", i);
}
```



Tipos de Loops

O laço while normalmente é mais simples, pois você define somente a condição de parada. Isso significa a inicialização e atualização deve ser feita em algum outro lugar, como você pode ver na imagem a seguir.

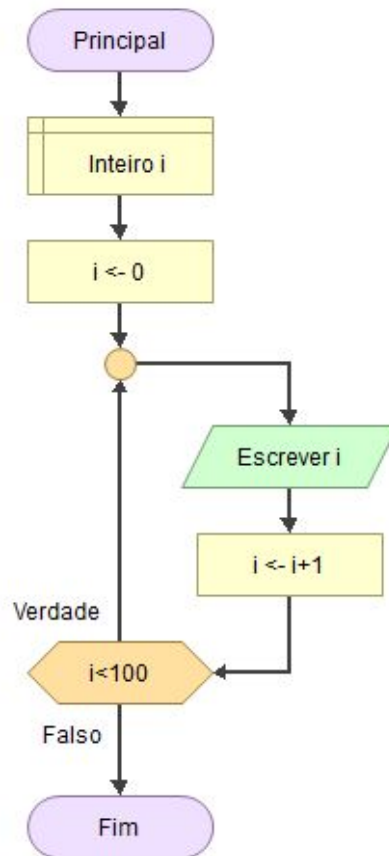
```
int i = 0;
while (i < 100){
    printf ("%d\n", i);
    i = i + 1;
}
```



Tipos de Loops

Os loops do-while (faça-equanto) se assemelham muito ao loop while, a diferença é que ele vai primeiro fazer uma vez o que está dentro do laço antes de testar a condição de parada.

```
int i = 0;
do{
    printf ("%d\n", i);
    i = i + 1;
}while (i < 100);
```

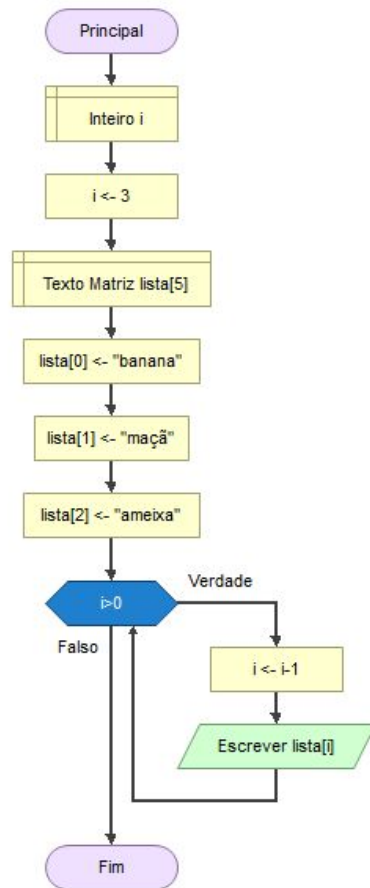


Tipos de Loops

Os loops for each (para cada) são normalmente usados quando temos alguma estrutura de dados como uma lista ou uma matriz já definidas e queremos que o laço execute uma vez para cada item dessa lista.

```
lista = ['banana', 'maçã', 'amei'
```

```
for item in lista:
    print (item)
```





Enunciado:

Escreva um programa em Python que solicite ao usuário quantos números ele deseja somar. Em seguida, o programa deve solicitar cada um desses números e ao final imprimir a soma total.



Enunciado:

Crie um programa que gera e imprime uma lista dos quadrados dos números de 1 a 10.



Enunciado:

Faça um programa que realiza uma contagem regressiva de 10 até 1 e, então, imprime "Lançamento!".



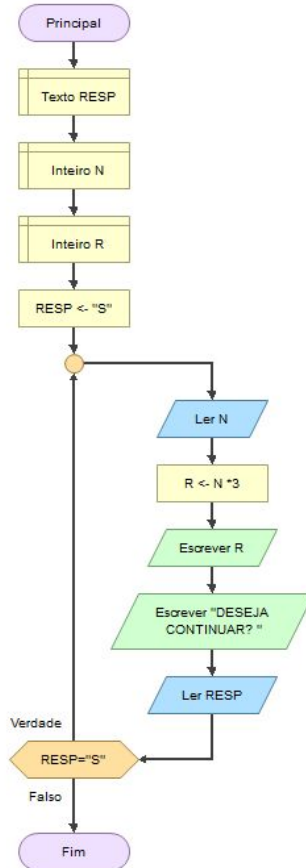
Enunciado:

Escreva um programa que pede ao usuário para inserir um número e verifica se o número é primo ou não. Um número primo é um número que só é divisível por 1 e por ele mesmo.

programa pós teste verdadeiro

```
var  
N, R : inteiro  
RESP : caractere
```

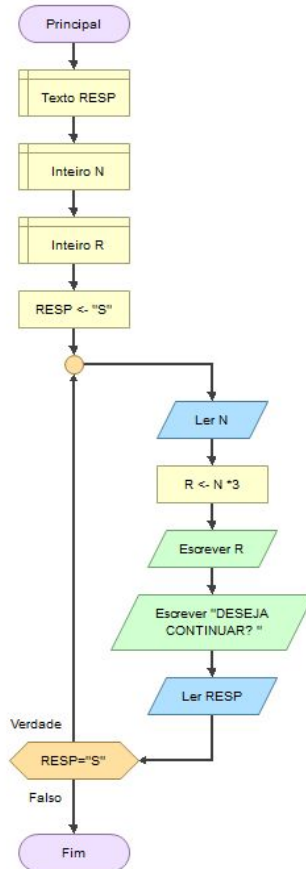
```
inicio  
  RESP <- "S"  
  execute  
    leia N  
     $r \leftarrow N * 3$   
    escreva R  
    escreva "deseja continuar"  
    leia RESP  
  enquanto_for (RESP = "S")  
fim
```



programa pós teste verdadeiro

```
var  
N, R : inteiro  
RESP : caractere
```

```
inicio  
  RESP <- "S"  
  execute  
    leia N  
     $r \leftarrow N * 3$   
    escreva R  
    escreva "deseja continuar"  
    leia RESP  
  enquanto_for (RESP = "S")  
fim
```



Bibliografia

AGUILAR, Luis Joyanes. Fundamentos de Programação: 3. ed. BOOKMAN,2008.

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. Algoritmos: Lógica e Desenvolvimento de Computadores: São Paulo: ÉRICA,2009.