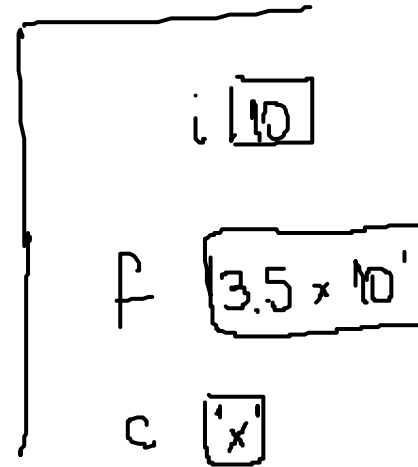


variáveis

int i = 10;

float f = 3.5;

char c = 'x';

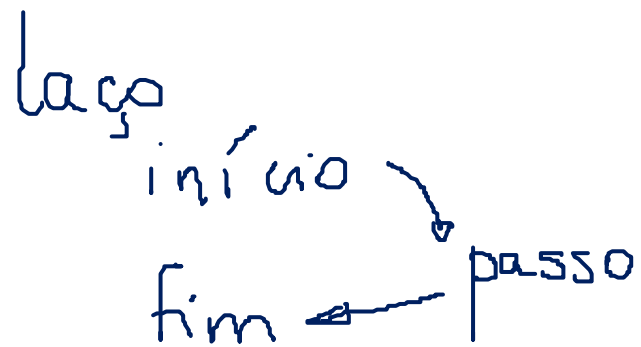


IO { print f
scan f
+ - * / %
ent
proc
saída

decisão
if () { }
else { }

repetição
while
do while

for



for { teste de início
 } bom p/ laço contado

for (início; teste; passo) {
 ==
 ==
}

→ condição inicial;
while (teste) {
 ==
 ==
 passo;
}

condição inicial; ← optional
do {
 ==
 ==
 passo (1 vez)
} while (teste);

```

int i = 1;
while (i <= 5) {
    printf ("%d", i);
    i++;
}

```

mem

kla

~~i = 1 2 3 4 5 6~~

1 2 3 4 5

int i;

```

for (i = 1; i <= 5; i++) {
    printf ("%d", i);
}

```

mem

~~i = 1 2 3 4 5 6~~

kla

1 2 3 4 5

do {

==

} while ();

for (início; teste; passo)

1) int i = 1;
for (; i <= 5; i++)

2) for (; ;) {
 printf("x");
}

→ loop infinito

3) while (1) {
 =

4) for (i = 1; i <= 5; ...) {
 printf("%d", i++);
}

5) for (i = 1, j = 10; i <= 5 & & j >= 7; i++, j--)
 printf("x");

6) for (scanf("%d", &i); i <= 0; scanf("%d", &i));

$$a = 7$$

$$b = 12$$

7 8 9 10 11 12

$$a = 13$$

$$b = 9$$

9 10 11 12 13

```
#include <stdio.h>
int main(){
    int a, b;
    printf("Digite os valores A e B: \n");
    scanf("%d%d", &a, &b);
    if(a < b){
        for( ;a <= b; a++){
            printf("%d\n", a);
        }
    }
    else{
        for( ;b <= a; b++){
            printf("%d\n", b);
        }
    }
    return 0;
}
```

```
int a, b, i;
    if (a < b) {
        for( i = a; i <= b; i++){
            printf("%d ", i);
        }
    }else{
        for( i = b; i <= a; i++){
            printf("%d ", i);
        }
    }
```

a, b

if (a % b == 0)

b é divisor de a

for

ler um inteiro positivo
(validação com do-while)

exibir os divisores, usando for

12

1 2 3 4 6 12

13

1 13

20

1 2 4 5 10 20

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

```
int main(){
    int num, i;
    do{
        printf("Digite um inteiro positivo: ");
        scanf("%d", &num);
    } while(num <= 0);
    for (i = 1; i <= num; i++){
        if(num % i == 0){
            printf("\n%d", i);
        }
    }
    return 0;
}
```

```
i = 1;
while (i <= num) {
    if
    i++;
}
```

num = 8

8	%	1	?	0	1
8	%	2	?	0	2
8	%	3	?	0	X
8	%	4	?	0	4
		5			X
		6			
		7			
8	%	8	?	0	8

num = 8

for (i = 1; i ≤ num; i++)

verdadeira
= ?

==
!=

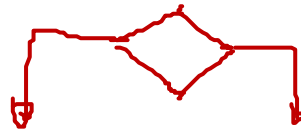
<=
>=

for (i = 1; i == num; i++)

falso

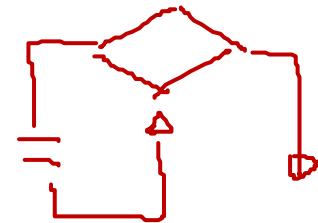
num = 8

i = 1



i = ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ 8

if
while \Leftrightarrow for



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

```
int main () {  
    int num, i;  
    i = 1;  
    do {  
        printf ("Digite um numero inteiro positivo: ");  
        scanf ("%d",&num);  
    }while (num <= 0);
```

```
    do {  
        if (num % i == 0){  
            printf ("%d  ", i);  
        }  
        i++;  
    }while (i <= num);
```

```
    return 0;
```

```
}
```

metade

num = 500

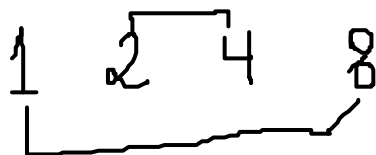
1 | 2 4 5 10 . . . 250 | 500
x

1000 000

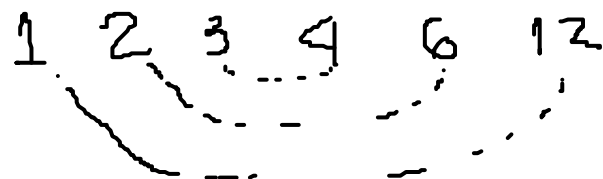
500 000 x 1000000

while (i <= num/2)
metade = num/2.

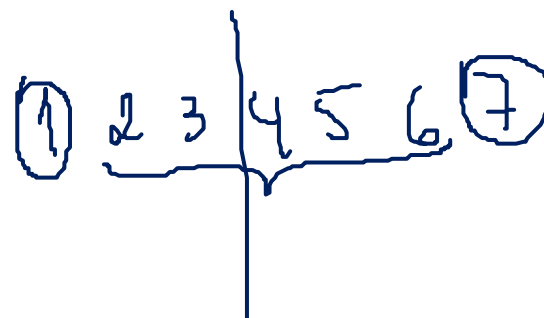
8



12



for $i = 2; i \leq \text{metade}$
 500.000



2

473
 2 3 4 237

1000 000
2 500.000
 ↗

20

$$\frac{20}{2} = 10$$

3

4

5

6

7

8

9

10

$$\text{num} = 4$$

divisores :

1 2 4

↗
metade

$$+\sqrt{4} = 2$$

for (~~0~~);

divisores :

13

13 / 2 :

6

$$\sqrt{13} \approx 3$$

↗

$$\begin{array}{r} 479 \overline{) 239} \\ 0 \end{array}$$

479

$$479 / 2 = \underline{\underline{239}}$$

$$\sqrt{479} \approx \underline{\underline{23}}$$

$$\lim_{n \rightarrow \infty} n \sqrt{n} \ll n^2$$

if (cond v) {
 ///
}