

5 valores

```
int n, ma, i;
```

```
scanf("%d", &n);
```

```
ma = n;
```

```
for (i = 1; i <= 4; i++) {
```

```
    scanf("%d", &n);
```

```
    if (n > ma) {
```

```
        ma = n;
```

```
    }
```

```
}
```

ma = 10

i = 1 2 3 4 5

com valores

6	4	10	8	9
0	1	2	3	4

```
int v[5], ma, i;
```

```
for (i = 0; i < 5; i++) {  
    scanf("%d", &v[i]);  
}
```

```
ma = v[0];
```

```
for (i = 1; i <= 4; i++) {
```

```
    if (v[i] > ma) {
```

```
        ma = v[i];
```

```
    }
```

```
}
```

dados (informação)
memória

variáveis { int
 - float
 - char

vetores _____

ponteiros

processamento
processador

i/o

decisão if - else

repetição 

modularização

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

```
int main(){  
    int v[10], ma = 0, i;  
    for(i = 0; i < 10; i++){  
        printf("\nDigite um valor inteiro: ");  
        scanf("%d", &v[i]);  
        if(v[i] % 2 == 0){  
            if(v[i] > ma){  
                ma = v[i];  
            }  
        }  
    }  
    printf("\nMaior: %d", ma);  
    return 0;  
}
```

-2 ~~-7~~ -4 -1 -8

ma = 0 X

INT_MAX

INT_MIN

ma = INT_MIN;

```

int main (){
    int valor[10], i, mv;

    for(i = 0; i < 10; i++){
        printf("digite 10 valores do vetor \n");
        scanf("%d", &valor[i]);

    }
    mv = valor[0] % 2 == 0;
    for(i = 0; i < 10; i++){
        if(valor[i] > mv){
            mv = valor[i];
        }
    }
    printf("maior valor par eh %d", mv);

    return 0;
}

```



valor[0] % 2 == 0

perguntando se o
valor[0] é par?

se verdade $mv = 1$
senão $mv = 0$

```

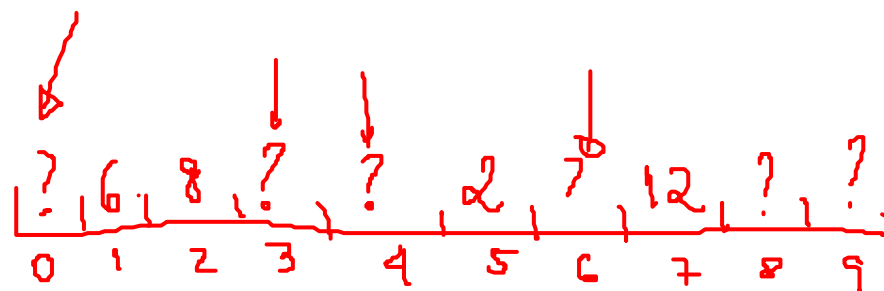
#include <stdio.h>
int main(){
    int vetor[10], numero, maior_par, i;

    printf("digite 10 valores inteiros:\n");

    for(i = 0; i < 10; i++) {
        scanf("%d", &numero);
        if(numero % 2 == 0){
            vetor[i] = numero;
        }
    }
    maior_par = vetor[0];
    for(i = 1; i <= 10; i++) {
        if(vetor[i] > maior_par){
            maior_par = vetor[i];
        }
    }
    printf("Maior par = %d", maior_par);

    return 0;
}

```



numero	i
3	0
6	1
8	2
7	3
11	4
2	5
3	6
12	7
13	8
	9

```
int main (){
```

procedural
→ em sequência

```
int i, A[10];
```

```
int ma = A[0] - → ma = .e?
```

```
for(i = 0; i < 10; i++){  
    printf("\nDigite um valor inteiro: ");  
    scanf("%d", &A[i]); ✗  
    if (ma < A[i] && A[i]%2==0){  
        ma = A[i];  
    }  
}
```

```
printf("Maior = %d\n", ma);
```



extremos separados/e

```
scanf ("%ld", &r[0]);  
ma = r[0];  
for (i = 1; i < 10; i++) {  
    scanf ("%ld", &r[i]);  
    if (r[i] > ma) {  
        ma = r[i];  
    }  
}
```

if (ma < A[i] && A[i]%2==0) {

F

~~X~~

V	e	V
V	V	F
F	F	V
F	F	F

if (A[i]%2==0 && ma < A[i])

↑

V	ou	V
V	V	F
F	V	V
F	F	F

if () ||

V

~~X~~

7 8 3 11 10
 —————>

~~i~~ ~~i~~ ~~i~~ ~~i~~ i →
 1 3 5 7 2 4 6 8 7 1
 1 1 1 1

1 3 5 7 9 11 13 15 17 19
 0 1 2 3 4 5 6 7 8 9

10
 i

```
ma = v[0]
if (v[0] % 2 == 0)
    ma = v[0];
else
    ma = v[1];
```

~~i = 0;~~
~~while (v[i] % 2 != 0) {~~
 ~~i++;~~
~~}~~

```
while (i < 10 && v[i] % 2 != 0) {
    i++;
}
```


1	3	5	7	2	4	6	8	7	1
0	1	2	3	4					

$i = 4$

```
else {
    printf("Nao tem pares!\n");
}
```

```
if (i < 10) {
    ma = v[i];
```

```
    for (j = i + 1; j < 10; j++) {
        if (v[j] > ma (v[j] % 2 == 0 && v[j] > ma)) {
            ma = v[j];
        }
```

```
    }
    printf(".....", ma);
}
```

ler 20 valores inteiros e armazená-los num vetor
separar os pares num vetor e os ímpares noutro
mostrar adequadamente os 3 vetores

0 2

1 4

2 5

3 6

4 7

13

21

4

2 4 6
0 1 2

5 7
0 1