

Erros

A questão do desenvolvimento não é simplesmente você acertar, na verdade a maioria das vezes você erra! Até as aulas anteriores nós ensinamos como os comandos, operações funcionam e como forçar um erro, o que fazer?

Constantes

Vamos tentar modificar uma constante, como o próprio nome diz é uma **constante**, ou seja, não pode ser modificada e o que me retornaria caso eu tentasse modificar?

```
#include <stdio.h>

const int VALOR = 9;

func main() {
    VALOR = VALOR + 1; // 10 ????
    printf("%d", VALOR);
}
```

Compilando o código:

```
$ gcc constante.c
constante.c:6:12: error: cannot assign to variable 'VALOR' w
    VALOR = 1; // 10 ????
    ~~~~~ ^
constante.c:3:12: note: variable 'VALOR' declared const here
    const int VALOR = 9;
    ~~~~~^~~~~~
1 error generated.
```

Variáveis

O que acontece se atribuirmos a uma variável mais do que ela consegue suportar em termos de tamanho?

```
#include <stdio.h>

int main() {
    int i = 1389289037891274897238947823748923894;
    printf("%d", i);
    return 0;
}
```

```
$ gcc variavel.c
variavel.c:4:11: error: integer literal is too large to be represented
    int i = 1389289037891274897238947823748923894;
            ^
variavel.c:4:11: warning: implicit conversion from 'unsigned long' to 'int'
    int i = 1389289037891274897238947823748923894;
            ~~~~~^~~~~~
1 warning and 1 error generated.
```

Variáveis 2

```
#include <stdio.h>

int main() {
    int i = 1;
}
```

```
char l = 'm';  
printf("%s", i + l);  
return 0;  
}
```

```
$ gcc variavel2.c  
variavel2.c:6:16: warning: format specifies type 'char *' but  
printf("%s", i + l);  
      ~~~      ^~~~~  
      %d  
1 warning generated.
```