05

Uso do long long

Resolvemos nesse curso um problema que calculava o fatorial de um número com a seguinte função recursiva:

```
int fatorial(int n){
    if(n > 0)
        return n*fatorial(n-1);
    else
        return 1;
}
```

Porém, fazendo alguns testes, você percebeu que essa função também sofria o problema de overflow para determinados valores de `n`, para consertar isso você implementou a função `fatorial_long`:

```
long long fatorial_long(int n){
    if(n > 0)
        return n*fatorial_long(n-1);
    else
        return 1;
}
```

Lembre-se que as variáveis do tipo `int` conseguem armazenar valores no intervalo de -2147483648 (-2^{31}) até 2147483647 ($2^{31}-1$), para facilitar, decorei esses valores como apenas $-2\ 10^9$ e $2\ 10^9$.

Por outro lado, as variáveis do tipo `long long`, que utilizam o dobro da memória que um `int` usa, consegue armazenar valores entre -9223372036854775808 (-2^{63}) e 9223372036854775807 ($2^{63}-1$), para facilitar, também decorei esses valores como sendo $9\ 10^{18}$ e $-9\ 10^{18}$.

Sobre essas funções o que podemos afirmar:

Selecione uma alternativa

A A função `fatorial` funciona corretamente apenas para valores de `n <= 32`, enquanto a função `fatorial_long` funciona apenas para `n <= 64`

B A função `fatorial` funciona corretamente apenas para valores de `n <= 15`, enquanto a função `fatorial_long` funciona apenas para `n <= 22`

C A função `fatorial` funciona corretamente apenas para valores de `n <= 12`, enquanto a função `fatorial_long` funciona apenas para `n <= 20`

D A função `fatorial` funciona corretamente apenas para valores de `n <= 10`, enquanto a função `fatorial_long` funciona apenas para `n <= 15`.