Matt Parker's Puzzle #12

You are creating a marching band. Your marching band must march in rows, but each row must have an equal number of performers in it.
You want your marching band to be able to march in exactly 64 different formations.

Puzzle for Submission: What is the fewest number of performers you require for your marching band to have 64 marching options? (Only whole positive numbers will be accepted)

This question is equivalent to the total number of divisors.
Example: n = 4 there are 3 formations

```
****   **      *
       **      *
               *
               *
```

```
—- Haskell Solution
-- Test if a number is a perfect square
perfect_square :: Integral a => a -> Bool
perfect_square n = sq * sq == n
    where sq = floor $ sqrt $ (fromIntegral n::Double)

-- Compute number of divisors ignoring if n is perfect square
num_of_div_raw :: (Integral a2, Num a1) => a2 -> a1
num_of_div_raw n = sum[2|c<-[1..floor $ sqrt $ (fromIntegral
                    n::Double)],n`rem`c == 0]

-- Compute corrected number of divisors
num_of_div :: (Integral a, Num p) => a -> p
num_of_div n = if perfect_square n then num_of_div_raw n - 1
               else num_of_div_raw n

-- answer -> 7560
-- (0.39 secs, 142,831,288 bytes)
answer = head[n|n<-[1..10000], num_of_div n == 64]

-- main -> 7560 formations
-- (0.40 secs, 142,831,416 bytes)
main :: IO ()
main = do
    (putStrLn.show) answer
```