

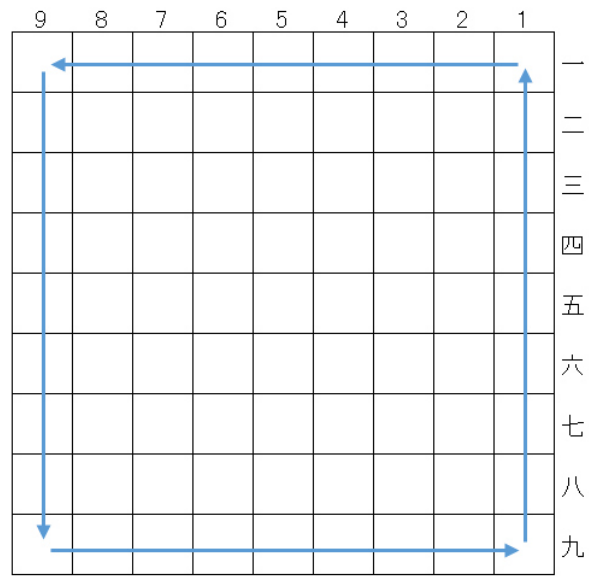
増井技術士事務所    今週のアルゴリズムの@masuideoさんの問題に挑戦中！

今週のお題：まわり将棋に挑戦！



あなたは1人でまわり将棋([Wikipedia](#))をしています。

まわり将棋では、あるカド(1一の位置)からスタートして、将棋盤の周囲のマスを移動します。



移動する「マスの数」を決めるために、サイコロの代わりに4枚の「金将」を使います。

将棋盤の上で4枚の「金将」を同時に振り、それぞれの向きで、移動するマスの数が決まります。

「金将」の向きによって、計算の元になる値は以下の通りです。

- ・裏向き：0
  - ・表向き：1
  - ・横向き：5
  - ・上向き：10
  - ・下向き：20
- (※それぞれの向きについては、[Wikipedia](#)を参照してください。)

上記を4枚の金将それぞれについて求め、その「和」が「動くマスの数」になります。

(ただし、裏向きが4枚揃った場合は、8となります。)

例えば、以下のようなマスの数だけ動きます。

金将の向き	動くマスの数
表・裏・裏・裏	1
表・表・裏・裏	2
横・表・表・裏	7
横・表・表・表	8
裏・裏・裏・裏	8
上・横・表・裏	16
下・上・横・表	36

なお、振った金将が1枚でも将棋盤から落ちたとき、または金将が重なった場合には無効(0マス)となります。

n 回振ったとき、進める駒がいずれかのカド(1一、1九、9一、9九)にいるようなコマの動きが何通りあるかを求めます。

標準入力から n が与えられたとき、標準出力にその動き方の数を出力してください。

例えば、n=1 のとき、いずれかのカドにいるために動くマスの数は以下の6通りがありますので、6を出力します。

0マス, 8マス, 16マス, 32マス, 40マス, 80マス

(「8」になるような金将の表現は2通りありますが、いずれも同じ位置に動くため1つとカウントします。)

このようにコマの動きに着目するため、例えば n=2 のとき、「1回目に3マス、2回目に5マス」動いたときと、

「1回目に5マス、2回目に3マス」動いた場合は別々とカウントします。

※ n は最大で7とします。

【入出力サンプル】

標準入力

1

標準出力

6

## 【解答方法】

■挑戦言語は下記のプログラム言語選択で選択可能なものであれば何でもOKです！

- 1)自分の書いたプログラム言語を選択
- 2)解答欄にソースコードを記入
- 3)送信前に「提出前に確認」ボタンをクリック（構文エラーがないかどうかチェックできます）
- 4)「解答コードは正常に実行されました」というメッセージを確認の上、「解答を送信」ボタンで解答してください。

■この問題にはテストケースが5つ用意されています。すべてに通れば正解です！

## 【採点について】

- ・採点は「ideone」を使ってプログラムを実行し、標準入力および標準出力のテストケースと照合して正誤を判定します
- ・各言語の標準入力と標準出力は[こちら](#)を参考にしてください
- ・標準入力の最終行の改行はあり／なし両方に対応してください

※なおCodeIQで使用しているideoneは企業版のため、webで公開されているコンシューマー版[ideone](#)とは

対応言語・バージョン・挙動が異なる場合があります。

企業版ideoneの対応バージョンは、「提出前チェック」の結果とともに表示されます。



## あなたの解答

```
import itertools
import time

# 実行時間を計測
def calc_time(n, sum):
    start = time.time()

    print(mawari_shogi(n, sum))

    elapsed_time = time.time() - start
    print(("elapsed_time: {0}".format(elapsed_time) + "[sec]"))

# n回振る時のパターン数を事前に準備しておく
def create_result(n):
    result = {}
    for hosu in range(0, 32):
        result[hosu] = mawari_shogi(n - 1, hosu)
    return result

def mawari_shogi(n, sum):
    result = 0

    if (4 < n):
        for hosu in deme_pattern:
            result += mawari_shogi(n - 1, sum + hosu)
    elif (4 == n):
        for hosu in deme_pattern:
            result += result_table[(sum + hosu) % 32]
    elif (3 == n) or (2 == n):
        for hosu in deme_pattern:
            result += mawari_shogi(n - 1, sum + hosu)
    elif (1 == n):
```



[ページ上部へ](#) ↗



公式Facebookページ



公式Twitterアカウント

### CodeIQとは

CodeIQ（コードアイキュー）とは、自分の实力を知りたいITエンジニア向けの、実務スキル評価サービスです。

### CodeIQご利用にあたって

[利用規約](#)  
[プライバシーポリシー](#)  
[お問い合わせ](#)  
[エンジニア採用ご担当者様へ](#)

### 関連サイト

[ATND](#)  
[CODE VS](#)  
[サンカク](#)  
[カクシン](#)

 CodeIQ

 リクルートグループサイトへ CodeIQ MAGAZINE

 CodeIQ JOBS

 Read CodeIQ on RSS

(C) Recruit Career Co.,Ltd.

[プレスリリース](#)

[CodeIQ](#)

[CodeIQ Magazine](#)

[CodeIQ JOBS](#)

[リクナビNEXT](#)

[リクナビNEXT ITキャリア](#)

[リクナビNEXT Tech総研](#)