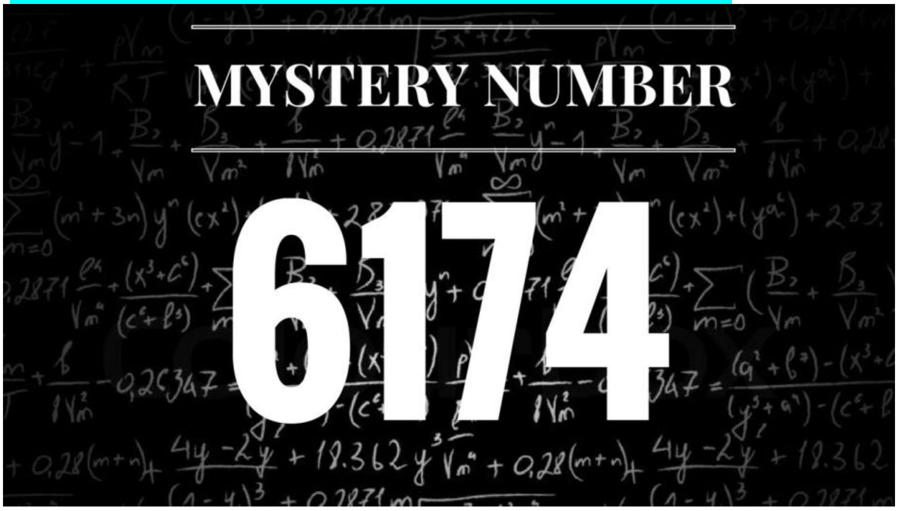
KARPREKAR’S CONSTANT



What's So Special About the Magic Number 6174?

The number '6174' looks like any other ordinary number but what if I told you this is a magic number? Would you believe me?

6174 is known as kaprekar’s constant as it was invented by **Indian mathematician** DR Kaprekar in 1949.

Even mathematicians, till date, aren't sure how to explain this magic number. Wondering, what's so unique about the number? This number is known for the rule as following:

1. Take any four-digit number, using at least two different digits

(leading zeros are allowed).

2. Arrange the digits in descending and then in ascending order

to get two four-digit numbers, adding leading zeros if

necessary.

3. Subtract the smaller number from the bigger number.

4. Go back to step 2 and repeat.

It was only in 1970 that people started recognizing Kaprekar's work after America's bestselling author, Martin Gardner wrote an article on Kaprekar in a magazine titled

'Scientific America'. Today, Kaprekar's contribution in the field of mathematics is known to the entire world.

For example, let's

6357 take the number 5432.

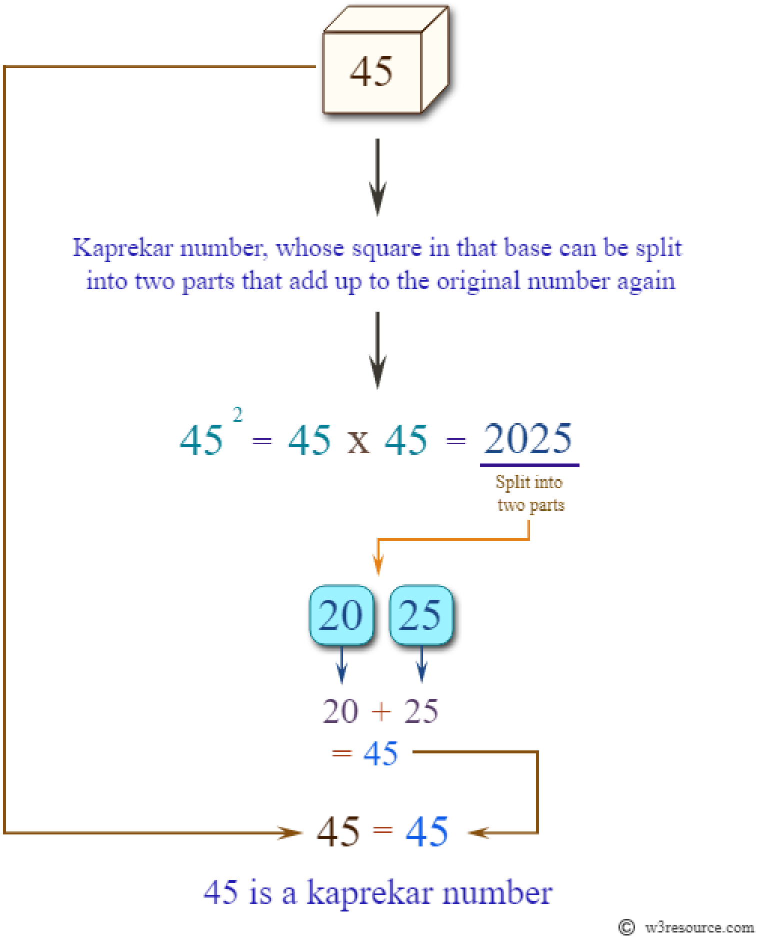
5432 – 2345 = 3087

8730 – 0378 = 8352

8532 – 2358 = 6174

7641 – 1467 = 6174

HOW TO FIND WHETHER A GIVEN NUMBER IS KAPREKAR NUMBER OR NOT?


45
Kaprekar number, whose square in that base can be split into two parts that add up to the original number again
45 2 = 45 x 45 = 2025
Split into two parts
20
25
20 + 25
= 45
45 = 45
45 is a kaprekar number
© w3resource.com

CODE IN ‘C’ LANGUAGE TO FIND WHETHER GIVE NUMBER IS KAPREKAR OR NOT?

```
#include <stdio.h>

#include <stdbool.h>

#include <math.h>

bool kaprekar(int n)
```

```
{
    if (n == 1)
       return true;
    int sqr_n = n * n;
    int ctr_digits = 0;
    while (sqr_n)
    {
        ctr_digits++;
        sqr_n /= 10;
    }
    sqr_n = n*n;
    for (int r_digits=1; r_digits<ctr_digits; r_digits++)
    {
         int eq_parts = pow(10, r_digits);

         if (eq_parts == n)
            continue;
         int sum = sqr_n/eq_parts + sqr_n % eq_parts;
         if (sum == n)
           return true;
    }
    return false;
}
int main()
{
int k;
```

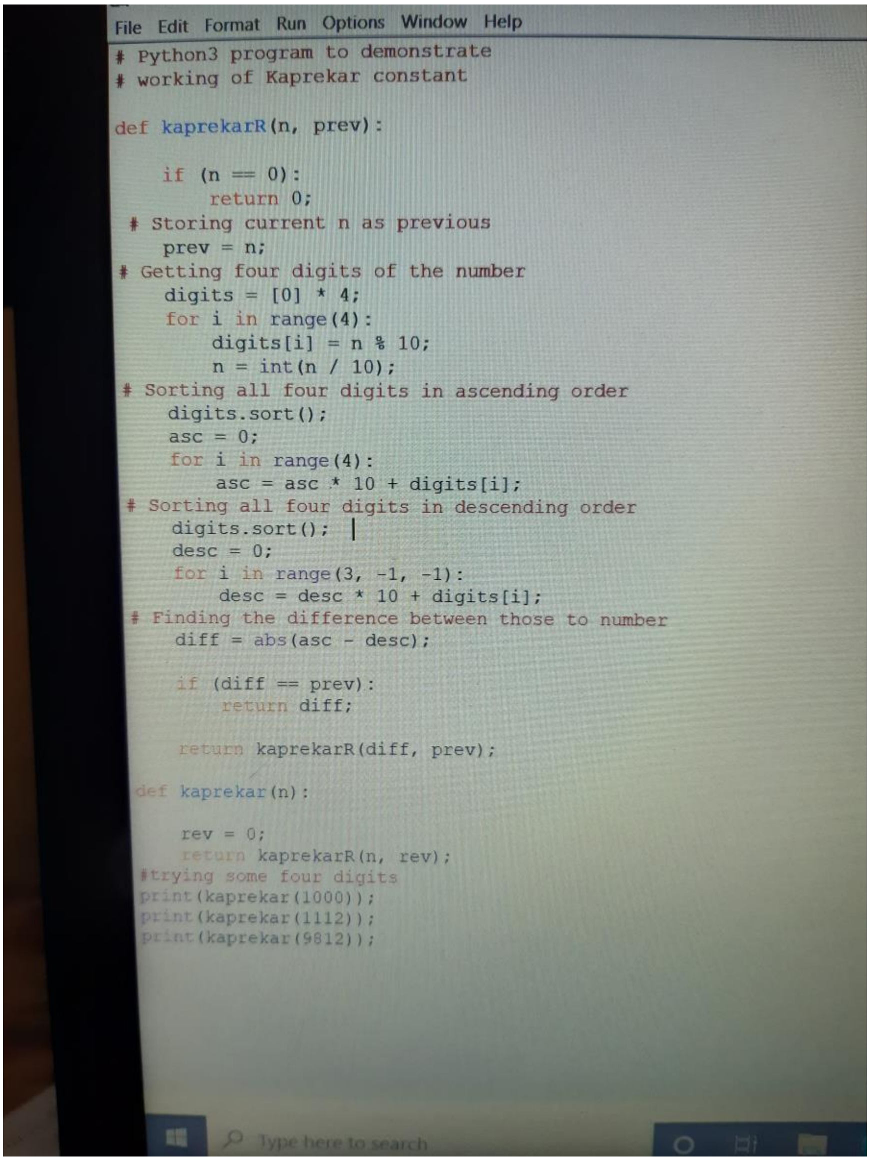
```
  printf("\n\n Check whether a given number is a Kaprekar number: \n");
  printf(" ---------------------------------------------------\n");
  printf(" Input a number: ");
  scanf("%d",&k);
        if (kaprekar(k)==true)
        {
         printf("%d is a Kaprekar number. \n",k);
        }
        else
        {
         printf("%d is not a Kaprekar number. \n",k);
        }
}
```

OUTPUT:

```
Input a number: 45
45 is a Kaprekar number.
```

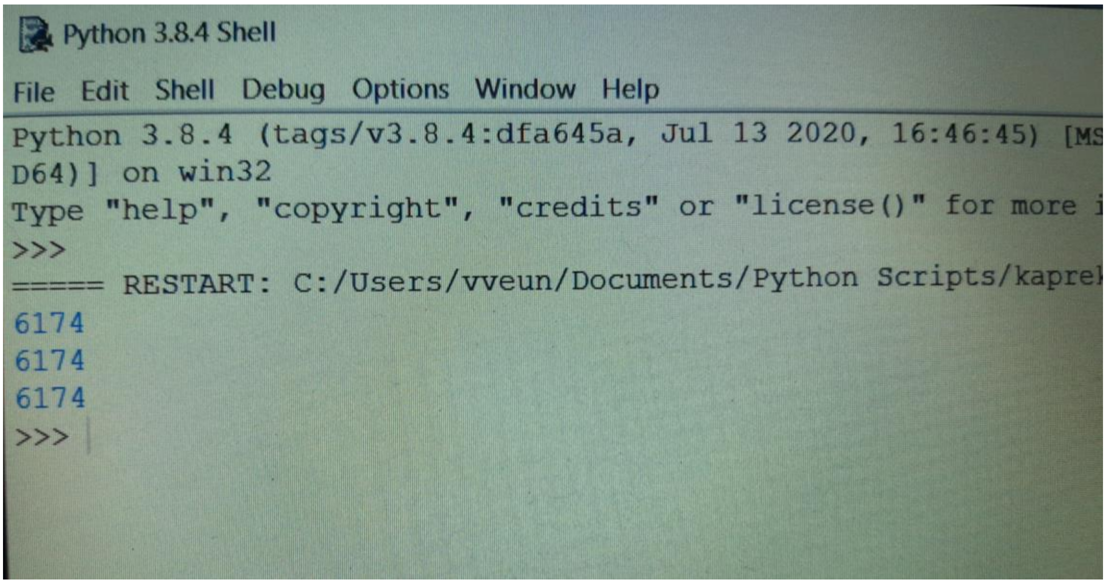
MAIN SOLUTION

DEMONSTRATING THE WORKING OF KAPREKAR’S NUMBER IN PYTHON:



```python
# Python3 program to demonstrate
# working of Kaprekar constant

def kaprekarR(n, prev):

    if (n == 0):
        return 0;
# Storing current n as previous
    prev = n;
# Getting four digits of the number
    digits = [0] * 4;
    for i in range(4):
        digits[i] = n % 10;
        n = int(n / 10);
# Sorting all four digits in ascending order
    digits.sort();
    asc = 0;
    for i in range(4):
        asc = asc * 10 + digits[i];
# Sorting all four digits in descending order
    digits.sort();
    desc = 0;
    for i in range(3, -1, -1):
        desc = desc * 10 + digits[i];
# Finding the difference between those to number
    diff = abs(asc - desc);

    if (diff == prev):
        return diff;

    return kaprekarR(diff, prev);

def kaprekar(n):

    rev = 0;
    return kaprekarR(n, rev);
#trying some four digits
print(kaprekar(1000));
print(kaprekar(1112));
print(kaprekar(9812));
```

OUTPUT



```
Python 3.8.4 (tags/v3.8.4:dfa645a, Jul 13 2020, 16:46:45) [MS
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more i
>>>
===== RESTART: C:/Users/vveun/Documents/Python Scripts/kaprek
6174
6174
6174
>>>
```

Code:

```
# Python3 program to demonstrate
# working of Kaprekar constant

def kaprekarR(n, prev):

    if (n == 0):
        return 0;
# Storing current n as previous
    prev = n;
# Getting four digits of the number
    digits = [0] * 4;
    for i in range(4):
        digits[i] = n % 10;
        n = int(n / 10);
# Sorting all four digits in ascending order
    digits.sort();
    asc = 0;
    for i in range(4):
        asc = asc * 10 + digits[i];
# Sorting all four digits in descending order
    digits.sort();
    desc = 0;
    for i in range(3, -1, -1):
        desc = desc * 10 + digits[i];
# Finding the difference between those to number
    diff = abs(asc - desc);

    if (diff == prev):
        return diff;

    return kaprekarR(diff, prev);

def kaprekar(n):

    rev = 0;
    return kaprekarR(n, rev);
#trying some four digits
print(kaprekar(1000));
print(kaprekar(1112));
print(kaprekar(9812));
```

Output

6174

6174

6174