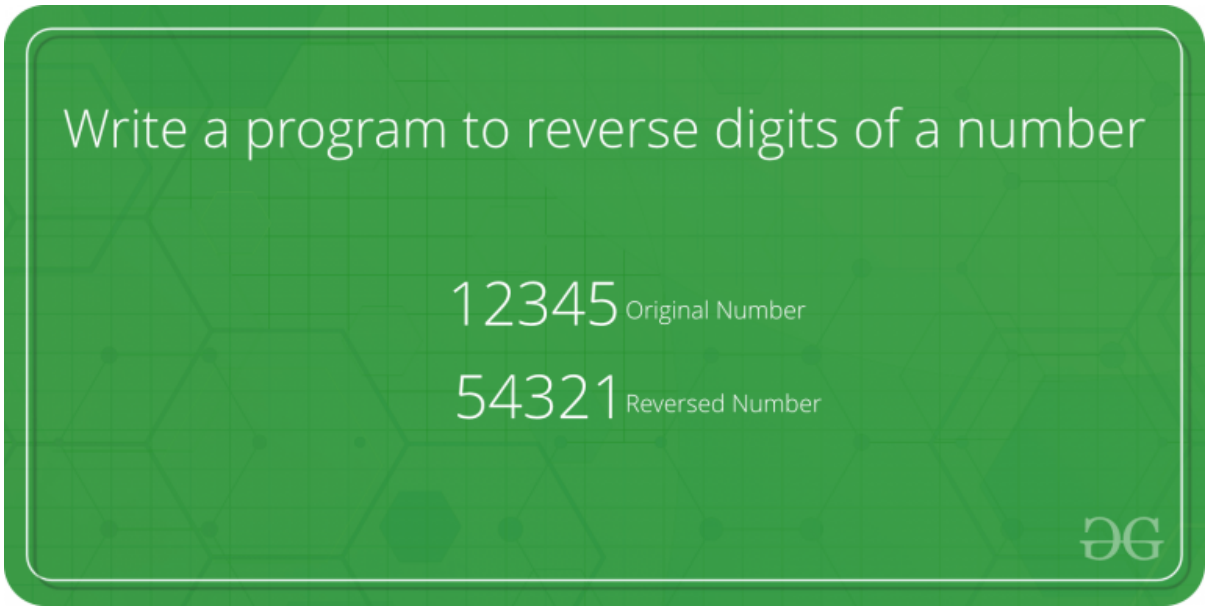
Write a program to reverse digits of a number

Last Updated: 20-04-2020

Write a program to reverse digits of an integer.



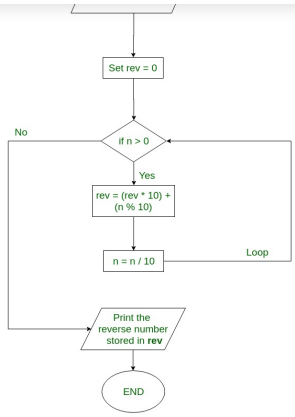
Examples :

```
Input : num = 12345
Output : 54321

Input : num = 876
Output : 678
```

Recommended: Please solve it on "*PRACTICE* " first, before moving on to the solution.

Flowchart:



Flowchart to find reverse of a number

ITERATIVE WAY

Algorithm:

```
Input:  num
(1) Initialize rev_num = 0
(2) Loop while num > 0
     (a) Multiply rev_num by 10 and add remainder of num
          divide by 10 to rev_num
               rev_num = rev_num*10 + num%10;
     (b) Divide num by 10
(3) Return rev_num
```

Example:

num = 4562

num = num/10 = 456

rev_num = rev_num *10 + num%10 = 20 + 6 = 26

num = num/10 = 45

rev_num = rev_num *10 + num%10 = 260 + 5 = 265

num = num/10 = 4

rev_num = rev_num *10 + num%10 = 265 + 4 = 2654

num = num/10 = 0

Program:

C++

```
#include <bits/stdc++.h>

using namespace std;
/* Iterative function to reverse digits of num*/
int reversDigits(int num)
{
    int rev_num = 0;
    while(num > 0)
    {
        rev_num = rev_num*10 + num%10;
        num = num/10;
    }
    return rev_num;
}

/*Driver program to test reversDigits*/
int main()
{
    int num = 4562;
    cout << "Reverse of no. is "
         << reversDigits(num);
    getchar();
    return 0;
}

// This code is contributed
// by Akanksha Rai(Abby_akku)
```

C

```
int reversDigits(int num)
{
    int rev_num = 0;
    while(num > 0)
    {
        rev_num = rev_num*10 + num%10;
        num = num/10;
    }
    return rev_num;
}

/*Driver program to test reversDigits*/
int main()
{
    int num = 4562;
    printf("Reverse of no. is %d", reversDigits(num));

    getchar();
    return 0;
}
```

Java

```
// Java program to reverse a number

class GFG
{
    /* Iterative function to reverse
    digits of num*/
    static int reversDigits(int num)
    {
        int rev_num = 0;
        while(num > 0)
        {
            rev_num = rev_num * 10 + num % 10;
            num = num / 10;
        }
        return rev_num;
    }

    // Driver code
    public static void main (String[] args)
    {
        int num = 4562;
        System.out.println("Reverse of no. is "
                           + reversDigits(num));
    }
```

Python

```python
# Python program to reverse a number

n = 4562;
rev = 0

while(n > 0):
    a = n % 10
    rev = rev * 10 + a
    n = n // 10

print(rev)

# This code is contributed by Shariq Raza
```

C#

```csharp
// C# program to reverse a number
using System;

class GFG
{
    // Iterative function to
    // reverse digits of num
    static int reversDigits(int num)
    {
        int rev_num = 0;
        while(num > 0)
        {
            rev_num = rev_num * 10 + num % 10;
            num = num / 10;
        }
        return rev_num;
    }

    // Driver code
    public static void Main()
    {
        int num = 4562;
        Console.Write("Reverse of no. is "
                      + reversDigits(num));
    }
```

PHP

```php
<?php
// Iterative function to
// reverse digits of num
function reversDigits($num)
{
    $rev_num = 0;
    while($num > 1)
    {
        $rev_num = $rev_num * 10 +
                        $num % 10;
        $num = (int)$num / 10;
    }
    return $rev_num;
}

// Driver Code
$num = 4562;
echo "Reverse of no. is ",
       reversDigits($num);

// This code is contributed by aj_36
?>
```

Time Complexity: O(Log(n)) where n is the input number.
Output:

```
2654
```

RECURSIVE WAY

Thanks to Raj for adding this to the original post.

C++

```cpp
// C++ program to reverse digits of a number
#include <bits/stdc++.h>
using namespace std;
/* Recursive function to reverse digits of num*/
int reversDigits(int num)
{
```

```
    reversDigits(num/10);
    rev_num += (num%10)*base_pos;
    base_pos *= 10;
}
return rev_num;
}

// Driver Code
int main()
{
    int num = 4562;
    cout << "Reverse of no. is "
         << reversDigits(num);

    return 0;
}

// This code is contributed
// by Akanksha Rai(Abby_akku)
```

C

```
// C program to reverse digits of a number
#include <stdio.h>;

/* Recursive function to reverse digits of num*/
int reversDigits(int num)
{
  static int rev_num = 0;
  static int base_pos = 1;
  if(num > 0)
  {
    reversDigits(num/10);
    rev_num  += (num%10)*base_pos;
    base_pos *= 10;
  }
  return rev_num;
}

/*Driver program to test reversDigits*/
int main()
{
    int num = 4562;
    printf("Reverse of no. is %d", reversDigits(num));

    getchar();
```

Java

```
// Java program to reverse digits of a number

// Recursive function to
// reverse digits of num
class GFG
{
static int rev_num = 0;
static int base_pos = 1;
static int reversDigits(int num)
{
    if(num > 0)
    {
        reversDigits(num / 10);
        rev_num += (num % 10) * base_pos;
        base_pos *= 10;
    }
return rev_num;
}

// Driver Code
public static void main(String[] args)
{
    int num = 4562;
    System.out.println(reversDigits(num));
}
}

// This code is contributed by mits
```

Python3

```
# Python 3 program to reverse digits
# of a number
rev_num = 0
base_pos = 1

# Recursive function to reverse
# digits of num
def reversDigits(num):
    global rev_num
    global base_pos
    if(num > 0):
```

```
# Driver Code
num = 4562
print("Reverse of no. is ",
         reversDigits(num))

# This code is contributed by Rajput-Ji
```

C#

```
// C# program to reverse digits of a number

// Recursive function to
// reverse digits of num
using System;
class GFG
{
static int rev_num = 0;
static int base_pos = 1;
static int reversDigits(int num)
{
    if(num > 0)
    {
        reversDigits(num / 10);
        rev_num += (num % 10) * base_pos;
        base_pos *= 10;
    }
return rev_num;
}

// Driver Code
public static void Main()
{
    int num = 4562;
    Console.WriteLine(reversDigits(num));
}
}

// This code is contributed
// by inder_verma
```

PHP

```php
/* Recursive function to
reverse digits of num*/
function reversDigits($num)
{
    global $rev_num;
    global $base_pos;
    if($num > 0)
    {
        reversDigits((int)($num / 10));
        $rev_num += ($num % 10) *
                    $base_pos;
        $base_pos *= 10;
    }
    return $rev_num;
}

// Driver Code
$num = 4562;
echo "Reverse of no. is ",
       reversDigits($num);

// This code is contributed by ajit
?>
```

Output:

```
Reverse of no. is 2654
```

Time Complexity: O(Log(n)) where n is the input number.

Reverse digits of an integer with overflow handled

Note that above above program doesn't consider leading zeroes. For example, for 100 program will print 1. If you want to print 001 then see this comment from Maheshwar.

Try extensions of above functions that should also work for floating point numbers.

Attention reader! Don't stop learning now. Get hold of all the important DSA concepts with the **DSA Self Paced Course** at a student-friendly price and become industry ready.

Recommended Posts:

Write an Efficient C Program to Reverse Bits of a Number

Write a program to reverse an array or string

C Program to reverse the digits of a number using recursion

Numbers of Length N having digits A and B and whose sum of digits contain only digits A and B

Minimum digits to be removed to make either all digits or alternating digits same

Find N numbers such that a number and its reverse are divisible by sum of its digits

Find smallest number with given number of digits and sum of digits

Find the Largest number with given number of digits and sum of digits

Number of digits in the nth number made of given four digits

Count of integers in a range which have even number of odd digits and odd number of even digits

Find smallest number with given number of digits and sum of digits under given constraints

Number formed by deleting digits such that sum of the digits becomes even and the number odd

Reverse digits of an integer with overflow handled

Smallest number with given sum of digits and sum of square of digits

Minimum number of digits to be removed so that no two consecutive digits are same

Check whether product of digits at even places is divisible by sum of digits at odd place of a number

Count of numbers between range having only non-zero digits whose sum of digits is N and number is divisible by M

Maximize the given number by replacing a segment of digits with the alternate digits given

Find the average of k digits from the beginning and l digits from the end of the given number

Check if the sum of digits of number is divisible by all of its digits

Improved By : jit_t, Mithun Kumar, inderDuMCA, Akanksha_Rai, Rajput-Ji, more

Practice Tags : MakeMyTrip MAQ Software Mathematical Recursion Modular Arithmetic

11

1.7

To-do Done

Based on **49** vote(s)

Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

GeeksforGeeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Practice

Courses

Company-wise

Topic-wise

How to begin?

Contribute

Write an Article

Write Interview Experience

Internships

Videos