Add two bit strings

Given two bit sequences as strings, write a function to return the addition of the two sequences. Bit strings can be of different lengths also. For example, if string 1 is “1100011″ and second string 2 is “10″, then the function should return “1100101″.

Since sizes of two strings may be different, we first make the size of smaller string equal to that of bigger string by adding leading 0s. After making sizes same, we one by one add bits from rightmost bit to leftmost bit. In every iteration, we need to sum 3 bits: 2 bits of 2 given strings and carry. The sum bit will be 1 if, either all of the 3 bits are set or one of them is set. So we can do XOR of all bits to find the sum bit. How to find carry – carry will be 1 if any of the two bits is set. So we can find carry by taking OR of all pairs. Following is step by step algorithm.

1. Make them equal sized by adding 0s at the begining of smaller string.
2. Perform bit addition
…..Boolean expression for adding 3 bits a, b, c
…..Sum = a XOR b XOR c
…..Carry = (a AND b) OR (b AND c) OR (c AND a)

Following is C++ implementation of the above algorithm.

```
#include <iostream>
using namespace std;

//adds the two bit strings and return the result
string addBitStrings( string first, string second );

// Helper method: given two unequal sized bit strings, converts them t
// same length by aadding leading 0s in the smaller string. Returns th
// the new length
int makeEqualLength(string &str1, string &str2)
```


```
{
    int len1 = str1.size();
    int len2 = str2.size();
    if (len1 < len2)
    {
        for (int i = 0 ; i < len2 - len1 ; i++)
            str1 = '0' + str1;
        return len2;
    }
    else if (len1 > len2)
    {
        for (int i = 0 ; i < len1 - len2 ; i++)
            str2 = '0' + str2;
    }
    return len1; // If len1 >= len2
}
```

```
// The main function that adds two bit sequences and returns the addit
string addBitStrings( string first, string second )
{
    string result;  // To store the sum bits

    // make the lengths same before adding
    int length = makeEqualLength(first, second);

    int carry = 0;  // Initialize carry

    // Add all bits one by one
    for (int i = length-1 ; i >= 0 ; i--)
    {
        int firstBit = first.at(i) - '0';
        int secondBit = second.at(i) - '0';

        // boolean expression for sum of 3 bits
        int sum = (firstBit ^ secondBit ^ carry)+'0';

        result = (char)sum + result;

        // boolean expression for 3-bit addition
        carry = (firstBit & secondBit) | (secondBit & carry) | (firstB
    }

    // if overflow, then add a leading 1
    if (carry)
        result = '1' + result;

    return result;
```


Popular Posts

```
}

// Driver program to test above functions
int main()
{
    string str1 = "1100011";
    string str2 = "10";

    cout << "Sum is " << addBitStrings(str1, str2);
    return 0;
}
```

Output:

```
Sum is 1100101
```

This article is compiled by **Ravi Chandra Enaganti**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Related Tpoics:

- Check if a number is multiple of 9 using bitwise operators
- How to swap two numbers without using a temporary variable?

- Divide and Conquer | Set 4 (Karatsuba algorithm for fast multiplication)
- Find position of the only set bit
- Swap all odd and even bits
- Write your own strcmp that ignores cases
- Binary representation of a given number
- Find the element that appears once

Writing code in comment? Please use **ideone.com** and share the link here.

13 Comments GeeksforGeeks

Sort by Newest

CODER • 2 months ago

Here is a code which is simple:

```
#include<stdio.h>

void showbit(int n)

{

printf("\n\nThe binary value of %d is ",n);

int s=sizeof(int)*8;

for(int i=s-1;i>=0;i--)
```


Recent Comments

Abhi You live US or India?

Google (Mountain View) interview · 31 minutes ago

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 1 hour ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 1 hour ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 1 hour ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 1 hour ago

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 2 hours ago

```
{

int m=1<<i; printf("%d",(n&m)=""">>i);

if(i%4==0)

printf(" ");
```

see more

Reply • Share ›

Karan Chugh • 8 months ago

Can anybody guide how to add efficiently two bit strings of length 100000 effici

Reply • Share ›

greek • 10 months ago

```
class add_bit_strings{
public static int add_strings(String str1,String str2){
int a = 0;
int b = 0;

int i;

for(i=0;i<str1.length();i++){
a<<=1;

if(str1.charAt(i) == &#0391&#039)
a+=1;
}
for(i=0;i<str2.length();i++){
b<<=1;

if(str2.charAt(i) == &#0391&#039)
b+=1;
```

```
}
```

see more

1 ▲ | ▼ • Reply • Share ›

abhishek08aug • a year ago

Here is a code without doing padding:

```
 #include<stdio.h>
#include<stdlib.h>
#include<string.h>

#define max(a,b) \
   ({ __typeof__ (a) _a = (a); \
       __typeof__ (b) _b = (b); \
     _a > _b ? _a : _b; })

#define min(a,b) \
   ({ __typeof__ (a) _a = (a); \
       __typeof__ (b) _b = (b); \
     _a < _b ? _a : _b; })

int get_length(char * str) {
  int length = 0;
```

see more

▲ | ▼ • Reply • Share ›

yaoliu • a year ago

There is a solution with iterator

```
 string addBitStr(string str1, string str2)
```

```
        string result;
        string::iterator it1 = str1.end() - 1, it2 = str2.end() - 1, i
        int carry = 0;
        for(int i = min(str1.size(), str2.size()); i > 0; i--, it1--,
        {
                int bit1 = *it1 - '0', bit2 = *it2 - '0';
                int sum = bit1 ^ bit2 ^ carry;
                carry = (bit1 & bit2) | (bit1 & carry) | (bit2 & carry
                result = (char)(sum + '0') + result;
        }
        if(it1 + 1 == str1.begin())
                it = it2;
        else
                it = it1;
```

see more

yaoliu ➚ yaoliu • a year ago

Sorry, there is a bug, the last 7th line should be:

carry = bit & carry

yuvaraj • a year ago

```
 #include <stdio.h>
#include <stdlib.h>
#include <string.h>

char string1[] = "1100011";
char string2[] = "10";
```

```
int makeEqualLength(char *str1, char *str2)
{
        int len1,len2,i;
        char *str_priv;

        len1 = strlen(str1);
        len2 = strlen(str2);

        if(len1 > len2)
        {
```

see more

• Reply • Share ›

Jitendra Khushwaha • a year ago

common yar...provide some new/tricky problems. this is a very basic question two string in a given base/radix.

• Reply • Share ›

Alexey Korovin • a year ago

It will restrict us to use only numbers that fit in integer type(~10 symbols words

• Reply • Share ›

Jitendra Khushwaha • a year ago

or we can use..
carry=(a& b) + (carry&(a ^ b)).

• Reply • Share ›

Ravi Chandra • a year ago

@sagar Sethi