# 67、二进制求和（Add Binary）

## 题目：

给你两个二进制字符串，返回它们的和（用二进制表示）。

输入为 **非空** 字符串且只包含数字 1 和 0。

**示例 1:**

输入: a = "11", b = "1"  
输出: "100"

**示例 2:**

输入: a = "1010", b = "1011"  
输出: "10101"

**提示：**

* 每个字符串仅由字符 '0' 或 '1' 组成。
* 1 <= a.length, b.length <= 10^4
* 字符串如果不是 "0" ，就都不含前导零。

## 解答：

法一：学习StringBuilder

|  |
| --- |
| public static string AddBinary(string a, string b)  {  int i = a.Length - 1;  int j = b.Length - 1;  //进位  int carry = 0;  StringBuilder sb = new StringBuilder();  while (i >= 0 || j >= 0)  {  int sum = carry;  if (i >= 0)  {  sum += a[i] - '0';  }  if (j >= 0)  {  sum += b[j] - '0';  }  //学习StringBuilder的Insert，在最前面插入  sb.Insert(0, sum % 2);  carry = sum / 2;  i--;  j--;  }  if (carry != 0)  {  sb.Insert(0,carry);  }  return sb.ToString();  } |

法二：暴力

|  |
| --- |
| //暴力列出所有的情况  public static string AddBinary1(string a, string b)  {  if (a.Length < b.Length)  {  string temp = a;  a = b;  b = temp;  }  char flag = '0';  string sum = "";  int alen = a.Length - 1;  //短的先找  for (int i = b.Length - 1; i >= 0; i--)  {  //flag == '0'  if (a[alen] == '0' && b[i] == '0' && flag == '0')  {  sum = '0' + sum;  }  else if (a[alen] == '0' && b[i] == '1' && flag == '0')  {  sum = '1' + sum;  }  else if (a[alen] == '1' && b[i] == '0' && flag == '0')  {  sum = '1' + sum;  }  else if (a[alen] == '1' && b[i] == '1' && flag == '0')  {  sum = '0' + sum;  flag = '1';  }  //flag == '1'  else if (a[alen] == '0' && b[i] == '0' && flag == '1')  {  sum = '1' + sum;  flag = '0';  }  else if (a[alen] == '0' && b[i] == '1' && flag == '1')  {  sum = '0' + sum;  }  else if (a[alen] == '1' && b[i] == '0' && flag == '1')  {  sum = '0' + sum;  }  else if (a[alen] == '1' && b[i] == '1' && flag == '1')  {  sum = '1' + sum;  flag = '1';  }  alen--;  }  //短的找完找长的剩下的  for (int i = alen; i >= 0; i--)  {  if (a[i] == '0' && flag == '0')  {  sum = '0' + sum;  }  else if (a[i] == '0' && flag == '1')  {  sum = '1' + sum;  flag = '0';  }  else if (a[i] == '1' && flag == '0')  {  sum = '1' + sum;  }  else if (a[i] == '1' && flag == '1')  {  sum = '0' + sum;  }  }  if (flag == '1')  {  sum = '1' + sum;  }  return sum;  } |