给定两个以字符串形式表示的非负整数 num1 和 num2，返回 num1 和 num2 的乘积，它们的乘积也表示为字符串形式。

**示例 1:**

**输入:** num1 = "2", num2 = "3"

**输出:** "6"

**示例 2:**

**输入:** num1 = "123", num2 = "456"

**输出:** "56088"

**说明：**

num1 和 num2 的长度小于110。

num1 和 num2 只包含数字 0-9。

num1 和 num2 均不以零开头，除非是数字 0 本身。

**不能使用任何标准库的大数类型（比如 BigInteger）**或**直接将输入转换为整数来处理**。

string multiply(string num1, string num2) {

if (num1 == "0" || num2 == "0") return "0";

int len1 = num1.length();

int len2 = num2.length();

int \*cache = new int[len1 + len2];

memset(cache, 0, (len1 + len2) \* sizeof(int));

for (int j = len2 - 1; j >= 0; j--)

{

for (int i = len1 - 1; i >= 0; i--)

{

cache[i + j + 1] += (num1[i] - '0') \* (num2[j] - '0');

}

}

int carry = 0;

for (int i = len1 + len2 - 1; i >= 0; i--)

{

int n1 = (cache[i] + carry) % 10;

int n2 = (cache[i] + carry) / 10;

cache[i] = n1;

carry = n2;

}

int start = 0;

for (int i = 0; i < len1 + len2; i++)

{

if (cache[i] != 0)

{

start = i;

break;

}

}

string result;

for (int i = start; i < len1 + len2; i++)

{

result.push\_back(cache[i] + '0');

}

delete[] cache;

return result;

}