

编程题_README

1.题目简介

给定一个整数数组，玩家 1 和玩家 2 轮流从数组中取数字。两个玩家开始得分都是 0，每一回合，玩家从数组的一端(头或尾)取出一个数字加入自己得分中，取出的数字将从数组中移除，当数组为空时游戏结束。假设玩家 1 为先手，设计一个函数，若玩家 1 能够必胜(即游戏过程中无论玩家 2 选头还是尾都必输)，则返回 true，否则返回 false。玩家 1 与玩家 2 同分认定为玩家 1 获胜。通过递归方式求解。

2.我的思路

(1)最后我们是要比较两个玩家最终得分的大小，所以我们可以比较两个玩家得分的差值就行。我们站在玩家 1 的视角，最后用玩家 1 的分数减去玩家 2 的分数，大于等于 0 则玩家 1 获胜，反之玩家 2 获胜。

(2)每个回合玩家取数字要么从头要么从尾取，故而剩下来的数组是原来数组的一个子数组。故而子数组又第一个元素下标和最后一个元素下标，分别设为 start 和 end。当 start=end 时，子数组只剩下最后一个数，玩家只能选择这个数；当 start<end 时，可以选择子数组中的 numbers[start]或 numbers[end]。选完之后再又对方玩家选择，所以这是一个递归的过程。

(3)当每次选取数字的时候，玩家肯定会选择那个人自己分数最大化的数字，所以在每次选择的时候，要比较 start 和 end 处的数字，看选择哪一个能让自己分数最大。

3.主要函数说明

(1)Get_maxdifferent(vector<int>& numbers, int start, int end)函数

这个函数的作用是计算玩家 1 和玩家 2 得分差值的最大值。注意这里选择 start 位置的得分差值为 numbers[start]-Get_maxdifferent(numbers,start+1,end)，因为差值需要减去下一步对面玩家所选择的分数，之后递归下去算对面玩家下一步的分数；同理，选择 end 位置的得分差值为 numbers[end]-Get_maxdifferent(numbers,Start,end-1)。因为差值需要减去下一步对面玩家所选择的分数，之后递归下去算对面玩家下一步的分数.而且注意这里每一步的返回值是从左边选择和从右边选

择的最大值，因为每一步玩家都会选择最有利于自己的一步。

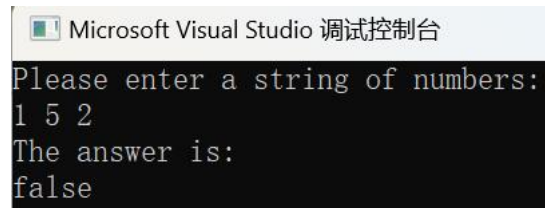
(2)Get_result(vector<int>& numbers)函数

这个函数的作用是判断最后的结果。有上面的 Get_maxdifferent(vector<int>& numbers, int start, int end)函数可以求得玩家 1 和玩家 2 得分差值的最大值(玩家 1 得分-玩家 2 得分)，如果这个差值大于 0 就是玩家 1 获胜，返回 true；反之则表示玩家 2 获胜，返回 false。

4.三个测试用例

用例 1: 1 5 2

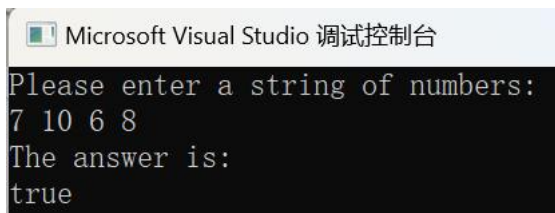
输出: false



```
Microsoft Visual Studio 调试控制台
Please enter a string of numbers:
1 5 2
The answer is:
false
```

用例 2: 7 10 6 8

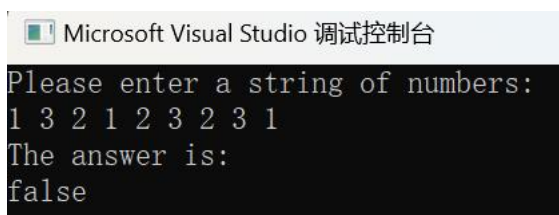
输出: true



```
Microsoft Visual Studio 调试控制台
Please enter a string of numbers:
7 10 6 8
The answer is:
true
```

用例 3: 1 3 2 1 2 3 2 3 1

输出: false



```
Microsoft Visual Studio 调试控制台
Please enter a string of numbers:
1 3 2 1 2 3 2 3 1
The answer is:
false
```