

截圖

The screenshot shows the LeetCode interface during a December LeetCode Challenge. On the left, a table displays submission history with columns for Time Submitted, Status, Runtime, Memory, and Language. The status of the submissions alternates between 'Accepted' and 'Wrong Answer'. On the right, a code editor shows a C++ solution for the 'Is Palindrome' problem. The code uses a while loop to extract digits into an array and a for loop to compare digits from both ends of the array.

Time Submitted	Status	Runtime	Memory	Language
12/16/2020 00:33	Accepted	16 ms	5.9 MB	c
12/16/2020 00:33	Accepted	20 ms	6 MB	c
12/16/2020 00:33	Wrong Answer	N/A	N/A	c
12/16/2020 00:32	Wrong Answer	N/A	N/A	c
12/16/2020 00:31	Wrong Answer	N/A	N/A	c
12/16/2020 00:31	Wrong Answer	N/A	N/A	c
12/16/2020 00:27	Wrong Answer	N/A	N/A	c
12/16/2020 00:24	Wrong Answer	N/A	N/A	c

```
1 bool isPalindrome(int x){
2     if(x<0) return false;
3     int a=0,cnt=0,x_arr[32];
4     while(x>0){
5         x_arr[cnt]=x%10;
6         x/=10;
7         cnt++;
8     }
9     for(int i=0;i<cnt/2;i++){
10         if(x_arr[i]!=x_arr[cnt-i-1]){
11             return false;
12         }
13     }
14     return true;
15 }
```

Source Code

```
1 bool isPalindrome(int x){
2     if(x<0) return false;
3     int cnt=0,x_arr[32];
4     while(x>0){
5         x_arr[cnt]=x%10;
6         x/=10;
7         cnt++;
8     }
9     for(int i=0;i<cnt/2;i++){
10         if(x_arr[i]!=x_arr[cnt-i-1]){
11             return false;
12         }
13     }
14     return true;
15 }
```

解釋

本題要求判斷一個數字是不是回文數，我的想法是將數字轉換成陣列的各個位數(line4~8)，比較index第0個跟最後一個是不是一樣，第1個跟倒數第1個是不是一樣，以此類推，只要有一個不一樣就return false，全部一樣就return true(line9~14)，但是遇到負數，比如說-101，反過來就是101-，負號不可能在後面，也就是說負數永遠不可能是回文數，因此在開頭先把他判掉(line 2)。