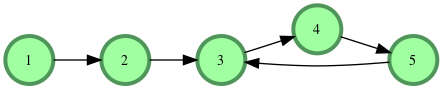
**Alternating Characters**

A linked list is said to contain a *cycle* if any node is visited more than once while traversing the list. For example, in the following graph there is a cycle formed when node 5 points back to node 3.



**Function Description**

Complete the function *has\_cycle* in the editor below. It must return a boolean *true* if the graph contains a cycle, or *false*.

has\_cycle has the following parameter(s):

* head: a pointer to a *Node* object that points to the head of a linked list.

**Note:** If the list is empty, head will be *null*.

**Input Format**

There is no input for this challenge. A random linked list is generated at runtime and passed to your function.

**Constraints**

* 0<=list, size <=100

**Output Format**

If the list contains a cycle, your function must return *true*. If the list *does not* contain a cycle, it must return *false*. The binary integer corresponding to the boolean value returned by your function is printed to stdout by our hidden code checker.

**Sample Input**

The following linked lists are passed as arguments to your function:

image

image

**Sample Output**

0

1

**Explanation**

1. The first list has no cycle, so we return *false* and the hidden code checker prints 0 to stdout.
2. The second list has a cycle, so we return *true* and the hidden code checker prints 1 to stdout.

using System.CodeDom.Compiler;

using System.Collections.Generic;

using System.Collections;

using System.ComponentModel;

using System.Diagnostics.CodeAnalysis;

using System.Globalization;

using System.IO;

using System.Linq;

using System.Reflection;

using System.Runtime.Serialization;

using System.Text.RegularExpressions;

using System.Text;

using System;

class Solution {

// Complete the alternatingCharacters function below.

static int alternatingCharacters(string s) {

int delnum=0;

char a=s[0];

for(int i=1;i<s.Length;i++){

if(a==s[i]){

delnum++;

}

else{a=s[i];}

}

return delnum;

}

static void Main(string[] args) {

TextWriter textWriter = new StreamWriter(@System.Environment.GetEnvironmentVariable("OUTPUT\_PATH"), true);

int q = Convert.ToInt32(Console.ReadLine());

for (int qItr = 0; qItr < q; qItr++) {

string s = Console.ReadLine();

int result = alternatingCharacters(s);

textWriter.WriteLine(result);

}

textWriter.Flush();

textWriter.Close();

}

}

**Congratulations**

You solved this challenge. Would you like to challenge your friends?

[Next Challenge](https://www.hackerrank.com/challenges/sherlock-and-valid-string?h_l=interview&playlist_slugs%5B%5D=interview-preparation-kit&playlist_slugs%5B%5D=strings&h_r=next-challenge&h_v=zen)

* **Test case 0**
* **Test case 1**
* **Test case 2**
* **Test case 3**
* **Test case 4**
* **Test case 5**
* **Test case 6**
* **Test case 7**
* **Test case 8**
* **Test case 9**
* **Test case 10**
* **Test case 11**
* **Test case 12**
* **Test case 13**
* **Test case 14**

Compiler Message

**Success**

Input (stdin)

Download

* **5**
* **AAAA**
* **BBBBB**
* **ABABABAB**
* **BABABA**
* **AAABBB**