

Have the function `ArrayChallenge(num)` take the `num` parameter being passed and perform the following steps. First take all the single digits of the input number (which will always be a positive integer greater than 1) and add each of them into a list. Then take the input number and multiply it by any one of its own integers, then take this new number and append each of the digits onto the original list. Continue this process until an adjacent pair of the same number appears in the list. Your program should return the least number of multiplications it took to find an adjacent pair of duplicate numbers.

For example: if `num` is 134 then first append each of the integers into a list: [1, 3, 4]. Now if we take 134 and multiply it by 3 (which is one of its own integers), we get 402. Now if we append each of these new integers to the list, we get: [1, 3, 4, 4, 0, 2]. We found an adjacent pair of duplicate numbers, namely 4 and 4. So for this input your program should return 1 because it only took 1 multiplication to find this pair.

Another example: if `num` is 46 then we append these integers onto a list: [4, 6]. If we multiply 46 by 6, we get 276, and appending these integers onto the list we now have: [4, 6, 2, 7, 6]. Then if we take this new number, 276, and multiply it by 2 we get 552. Appending these integers onto the list we get: [4, 6, 2, 7, 6, 5, 5, 2]. Your program should therefore return 2 because it took 2 multiplications to find a pair of adjacent duplicate numbers (5 and 5 in this case).

Examples

Input: 8

Output: 3

Input: 198

Output: 2

Browse Resources

Search for any help or documentation you might need for this problem. For example: array indexing, Ruby hash tables, etc.

Ans-

```
function PairSearching(num) {  
  let numArray = [num];  
  let count = 0;  
  let flag = false;  
  let searching = (arr) => {
```

```

let res = false;

let numsArr = [];

arr.forEach(val1 => {

let numList = String(val1).split("");

numList.forEach(val2 => {

let product = val1 * Number(val2);

let bigNumList =

numList.concat(String(product).split(""));

if (repeatNum(bigNumList)) {

res = true;

} else {

numsArr.push(product);

}

});

});

return res || numsArr;

}

let repeatNum = (numArr) => {

for (let i = 0, len = numArr.length; i < len - 1; i++) {

if (numArr[i] === numArr[i + 1]){

return true;

}

}

return false;

}

while (!flag) {

count++;

if (searching(numArray) === true) {

return count;

} else {

numArray = searching(numArray);

```

```
}
```

```
}
```

```
}
```

```
// keep this function call here
```

```
PairSearching(readline());
```