

Đã bắt đầu vào lúc	Thứ năm, 14 Tháng mười hai 2023, 1:19 PM
Tình trạng	Đã hoàn thành
Hoàn thành vào lúc	Thứ năm, 14 Tháng mười hai 2023, 2:37 PM
Thời gian thực hiện	1 giờ 17 phút
Điểm	3,00/3,00
Điểm	10,00 của 10,00 (100%)

Câu hỏi 1

Chính xác

Điểm 1,00 của 1,00

Implement function

```
int foldShift(long long key, int addressSize);
int rotation(long long key, int addressSize);
```

to hashing key using Fold shift or Rotation algorithm.

Review Fold shift:

The **folding method** for constructing hash functions begins by dividing the item into equal-size pieces (the last piece may not be of equal size). These pieces are then added together to give the resulting hash value.

For example:

Test	Result
cout << rotation(600101, 2);	26

Answer: (penalty regime: 0 %)

Reset answer

```
1 int foldShift(long long key, int addressSize)
2 {
3     long long tmp = key;
4     vector<int> arr(100);
5     int n = 0;
6     while(tmp != 0) {
7         arr[n] = tmp % 10;
8         tmp = tmp/10; ++n;
9     }
10    long long sum = 0;
11    while(n > 0) {
12        long long s = 0;
13        for(int i = 0; i < addressSize; ++i) {
14            if(n - i - 1 >= 0) s = s*10 + arr[n-i-1];
15        }
16        n -= addressSize; sum += s;
17    }
18    long long t = 1;
19    for(int i = 0; i < addressSize; ++i) t *= 10;
20    return sum % t;
21 }
22
23 int rotation(long long key, int addressSize)
24 {
25     long long tmp = key;
26     vector<int> arr(100);
27     int n = 0;
28     while(tmp != 0) {
29         arr[n] = tmp % 10;
30         tmp = tmp/10; ++n;
31     }
32     tmp = arr[0];
33     for(int i = 0; i < n-1; ++i) arr[i] = arr[i+1];
34     arr[n - 1] = tmp;
35     long long sum = 0;
36     while(n > 0) {
37         long long s = 0;
38         for(int i = 0; i < addressSize; ++i) {
39             if(n - i - 1 >= 0) s = s*10 + arr[n-i-1];
40         }
41         n -= addressSize; sum += s;
42     }
```

```
43     long long t = 1;
44     for(int i = 0; i < addressSize; ++i) t *= 10;
45     return sum % t;
46 }
```

	Test	Expected	Got	
✓	cout << rotation(600101, 2);	26	26	✓

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 2

Chính xác

Điểm 1,00 của 1,00

Implement three following hashing function:

```
long int midSquare(long int seed);
long int moduloDivision(long int seed, long int mod);
long int digitExtraction(long int seed, int* extractDigits, int size);
```

Note that:

In midSquare function: we eliminate 2 last digits and get the 4 next digits.

In digitExtraction: extractDigits is a sorted array from smallest to largest index of digit in seed (index starts from 0). The array has size **size**.

For example:

Test	Result
int a[]={1,2,5}; cout << digitExtraction(122443,a,3);	223
cout <<midSquare(9452);	3403

Answer: (penalty regime: 0, 0, 0 %)

Reset answer

```
1 long int midSquare(long int seed)
2 {
3     int square = seed * seed;
4     return (square / 100) % 10000; // Bo 2 cuoi lay 4 cai tiep theo
5 }
6 long int moduloDivision(long int seed, long int mod)
7 {
8     return seed % mod; // chia lay du thoi
9 }
10 long int digitExtraction(long int seed,int* extractDigits,int size)
11 {
12     vector<int> digits;
13     while(seed > 0) {
14         digits.push_back(seed % 10);
15         seed /= 10;
16     }
17     std::reverse(digits.begin(), digits.end());
18     long int sum = 0;
19     for (int i = 0; i < size; ++i) {
20         if (extractDigits[i] < int(digits.size()))
21             sum = sum * 10 + digits[extractDigits[i]];
22     }
23     return sum;
24 }
```

	Test	Expected	Got	
✓	<pre>int a[]={1,2,5}; cout << digitExtraction(122443,a,3);</pre>	223	223	✓
✓	<pre>cout <<midSquare(9452);</pre>	3403	3403	✓

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 3

Chính xác

Điểm 1,00 của 1,00

There are n people, each person has a number between 1 and 100000 ($1 \leq n \leq 100000$). Given a number $target$. Two people can be matched as a **perfect pair** if the sum of numbers they have is equal to $target$. A person can be matched no more than 1 time.

Request: Implement function:

```
int pairMatching(vector<int>& nums, int target);
```

Where $nums$ is the list of numbers of n people, $target$ is the given number. This function returns the number of **perfect pairs** can be found from the list.

Example:

The list of numbers is {1, 3, 5, 3, 7} and $target = 6$. Therefore, the number of **perfect pairs** can be found from the list is 2 (pair (1, 5) and pair (3, 3)).

Note:

In this exercise, the libraries `iostream`, `string`, `cstring`, `climits`, `utility`, `vector`, `list`, `stack`, `queue`, `map`, `unordered_map`, `set`, `unordered_set`, `functional`, `algorithm` has been included and `namespace std` are used. You can write helper functions and classes. Importing other libraries is allowed, but not encouraged, and may result in unexpected errors.

For example:

Test	Result
<pre>vector<int>items{1, 3, 5, 3, 7}; int target = 6; cout << pairMatching(items, target);</pre>	2
<pre>int target = 6; vector<int>items{4,4,2,1,2}; cout << pairMatching(items, target);</pre>	2

Answer: (penalty regime: 0, 0, 0, 5, 10, ... %)

Reset answer

```
1 int pairMatching(vector<int>& nums, int target) {
2     unordered_map<int, int> m;
3     int pairs = 0;
4     for(int i = 0; i < int(nums.size()); ++i) {
5         if(m[target - nums[i]] > 0) {
6             ++pairs;
7             --m[target - nums[i]];
8         }
9         else {
10            ++m[nums[i]];
11        }
12    }
13    return pairs;
14 }
```

	Test	Expected	Got	
✓	<pre>vector<int>items{1, 3, 5, 3, 7}; int target = 6; cout << pairMatching(items, target);</pre>	2	2	✓

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

BÁCH KHOA E-LEARNING



WEBSITE

HCMUT

MyBK

BKSI

LIÊN HỆ

📍 268 Lý Thường Kiệt, P.14, Q.10, TP.HCM

☎ (028) 38 651 670 - (028) 38 647 256 (Ext: 5258, 5234)

✉ elearning@hcmut.edu.vn

