

## CÁCH ĐÁNH GIÁ ĐIỂM THỰC HÀNH

### HỌC PHẦN: IT3150 – Project 1- 2023.1

#### I. Quy định, yêu cầu:

- Tài liệu và nội dung thực hành chấm điểm trên hệ thống:  
<https://lab.soict.hust.edu.vn/>
- Bài tập trên lớp chấm điểm tự động (các bài không chấm trên hệ thống làm vào máy tính → làm báo cáo thực hành – Theo mẫu).
- Hạn nộp báo cáo trên Teams (Bài tập trên lớp + Bài tập về nhà): 1 tuần.

#### II. Đánh giá điểm thực hành

- Chuyên cần (đúng giờ, nghiêm túc trong giờ học) - Điểm danh trên Teams: 10%
- Báo cáo thực hành (bài tập trên lớp + Về nhà) theo mẫu nộp trên Teams: 40%
- Trắc nghiệm – Form trên Teams: 10%
- Kiểm tra thực hành: 40%. (Tiết 2,3 buổi thực hành thứ 5).

**Điểm thưởng: 5% → 10% (Cho Mục 1,2 điểm TB từ 9-10).**

Tham gia thực hành đúng giờ đầy đủ theo thời khóa biểu (nếu có lý do không đi thực hành đúng kíp được thì gửi mail xin phép thực hành bù trước 1 ngày qua mail [hoalt@soict.hust.edu.vn](mailto:hoalt@soict.hust.edu.vn), Tiêu đề: đăng ký học bù – IT3040 – MaLopTH.

Các kíp có thể bù:

TT	Thời gian, địa điểm, Tuần học	Mã nhóm	Mã lớp
1			
2			
3			
4			
5			
6			
7			

**Nếu nghỉ không có lý do 3 buổi, không thực hành bù thì điểm chuyên cần, báo cáo và BTVN coi như 0 điểm thực hành.**

Contents

Vấn đề: Chuỗi lưu trữ và tìm kiếm .....	3
Vấn đề: Băm trên chuỗi .....	6
Problem: Kiểm tra xuất hiện .....	9
Bài toán: Tính tổng cặp dãy bằng một số .....	12

Figure 1 source code bài 1 .....	4
Figure 2 source code bài 1 .....	4
Figure 3 Test bài 1 .....	5
Figure 4 Test bài 1 .....	5
Figure 5 Test bài 1 .....	6
Figure 6 Test bài 1 .....	6

## Vấn đề: Chuỗi lưu trữ và tìm kiếm

---

Cơ sở dữ liệu chứa một chuỗi khóa  $k_1, k_2, \dots, k_n$  là các chuỗi ( $1 \leq n \leq 100000$ ). Thực hiện một chuỗi hành động gồm hai loại:

- find  $k$ : tìm và trả về 1 nếu  $k$  tồn tại trong cơ sở dữ liệu, và trả về 0, nếu không
- Insert  $k$ : chèn khóa  $k$  vào cơ sở dữ liệu và trả về 1 nếu chèn thành công ( $k$  không tồn tại trong cơ sở dữ liệu) và trả về 0 nếu chèn thất bại ( $k$  tồn tại trong cơ sở dữ liệu)

Lưu ý rằng độ dài của bất kỳ khóa nào đều lớn hơn 0 và nhỏ hơn hoặc bằng 50.

### Đầu vào

Hai khối thông tin. Khối đầu tiên chứa khóa ( $k_1, k_2, \dots, k_n$ ) trên mỗi dòng. Khối đầu tiên được kết thúc bằng một dòng chứa \*. Khối thứ hai là một chuỗi các hành động của hai find được mô tả ở trên: mỗi dòng chứa 2 chuỗi: cmd và  $k$  trong đó cmd = find hoặc Insert và  $k$  là key (tham số của hành động). Khối thứ hai được kết thúc bằng một dòng chứa \*\*\*. Lưu ý rằng số lượng hành động có thể lên tới 100000.

### Đầu ra

Mỗi dòng chứa kết quả (0 hoặc 1) của hành động tương ứng.

Source code

```
//C++
#include <bits/stdc++.h>

using namespace std;
set<string> keys;
int main()
{
    ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
    string s;
    while (getline(cin, s)) {
        if (s[0] == '*') {
            break;
        }
        auto t=keys.insert(s);
    }
    while (getline(cin, s)) {
        if (s == "****") {
            break;
        }
    }
    stringstream ss(s);
    string s1;
    string s2;
    ss >> s1 >> s2;
    if (s1[0] == 'f') {
        auto t = keys.find(s2);
```

Figure 1 source code bai 1

```
ss >> s1 >> s2;
if (s1[0] == 'f') {
    auto t = keys.find(s2);
    if (t == keys.end()) {
        cout << 0 << "\n";
    }
    else {
        cout << 1 << "\n";
    }
}
else {
    auto t = keys.insert(s2);
    cout << t.second << "\n";
}
}
return 0;
}
```

Figure 2 source code bai 1

**Test:**

## Input

```
computer
university
school
technology
phone
*
find school
find book
insert book
find algorithm
find book
insert book
***
```

Figure 3 Test bài 1

## Correct output

```
1
0
1
0
1
0
```

## User output

```
1
0
1
0
1
0
```

Figure 4 Test bài 1

## Input

```

isvspolwzkvflyyipeptos
gnijwxemupblnn
fqyoisizadytfgahzldaccmikvljggkmzlmwxcyyjnafd
cqsfpplzuoqdaawixvlkysxcwhhgbjokvlinpdpe
akycqbifkpsntsbsatgnozegkhowvocpuqipebdeoyo
gvxajnvddjhhtyxapblycvamhltxtfujbmt
svdfaiuyynzdwzytyudxregwkrhtuhgoyrwdjlofp
agsmeeswcwsdkbjtxripbchspubszqwyxqtzd
ojwfpqpoavndeqbupkcaryebclykjasjtiffbbya
wkaiqijwnfhjbdzqelqafj
*
find xzmrargurnqzrbklhryekpdzebn
insert xzmrargurnqzrbklhryekpdzebn
find eejpcqntxmmytecyqvufyzfm
insert eejpcqntxmmytecyqvufyzfm
find hrynmsnilvsstsgpgfbgguqpqsmvqymevjyx
insert hrynmsnilvsstsgpgfbgguqpqsmvqymevjyx
find eznqnipacsgobnxcqxqfjjsxptquqbuxdgrqev

```

Figure 5 Test bài 1

## Correct output

```

0
1
0
1
0
1
0
1
1
1
1

```

## User output

```

0
1
0
1
0
1
0
1
1
1
1

```

Figure 6 Test bài 1

## Vấn đề: Băm trên chuỗi

Cho một chuỗi  $s[1\dots k]$  là một chuỗi các ký tự được lấy từ  $\{'a', \dots, 'z'\}$ . Cho số nguyên dương  $m$ , mã băm của  $s$  được xác định theo công thức:

$H(s) = (s[1]*256^{k-1} + s[2]*256^{k-2} + \dots + s[k]*256^0) \bmod m$  (số nguyên  $m$  là một tham số)

Cho dãy các chuỗi  $k_1, k_2, \dots, k_n$ , tính mã băm tương ứng

### Đầu vào

Dòng 1:  $n$  và  $m$  ( $1 \leq n, m \leq 100000$ )

Dòng  $i+1$  ( $i = 1, 2, \dots, n$ ): chứa chuỗi  $k_i$  (độ dài mỗi chuỗi nhỏ hơn hoặc bằng 200)

### Đầu ra

Mỗi dòng chứa mã băm tương ứng của  $n$  chuỗi cho trước

Source code

```
//C++
#include <bits/stdc++.h>

using namespace std;
int n, m;
int calcPos(string s) {
    int val = 0;
    int len = s.length();

    for (int i = 0; i < len; i++) {
        int t = s[i];
        val = (val * 256 + t) % m;
    }
    return val;
}
int main()
{
    ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
    string s;
    getline(cin, s);
    stringstream ss(s);
    ss >> n >> m;
```

```
int main()
{
    ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
    string s;
    getline(cin, s);
    stringstream ss(s);
    ss >> n >> m;

    for (int i = 0; i < n; i++) {

        getline(cin, s);
        cout << calcPos(s) << "\n";
    }
    return 0;
}
```

## Test:

Input

```
4 1000
a
ab
abc
abcd
```

Correct output

```
97
930
179
924
```

User output

```
97
930
179
924
```



Input

```
5 10
hq
h
meayl
l
dxf
```

Correct output

```
7
4
6
8
2
```

User output

```
7
4
6
8
2
```

## Problem: Kiểm tra xuất hiện

---

Cho dãy số nguyên  $A_1, A_2, \dots, A_n$  với mỗi số nguyên  $A_i$  kiểm tra xem có số  $A_j$  nào bằng  $A_i$  hay không với  $j < i$ .

### Input

Dòng đầu chứa số  $n$  ( $1 \leq n \leq 100,000$ )

Dòng hai chứa  $n$  số nguyên  $A_1, A_2, \dots, A_n$  ( $1 \leq A_i \leq 1,000,000,000$ )

### Output

Ghi ra  $n$  dòng, dòng thứ  $i$  in ra 1 nếu tồn tại  $A_j = A_i$  với  $j < i$ , ngược lại in ra 0.

## TEST:

Source code

```
//C++
#include <bits/stdc++.h>

#include <unordered_set>
using namespace std;
unordered_set<int> all;
int main()
{
    ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
    int n;
    cin >> n;

    for (int i = 0; i < n; i++) {
        int j;
        cin >> j;
        auto t = all.find(j);
        if (t == all.end()) {
            cout << 0<<"\n";
        }
        else {
            cout << 1<<"\n";
        }
        all.insert(j);
    }
    return 0;
```

Input

100000

69979 301413357 791550397 160567226 13748 66291 77477 597007 215706823 392289611 14747 37215529 3461

Correct output

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

1

0

0

1

1

0

0

0

1

1

0

0

0

0

1

1

0

User output

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

1

0

0

1

1

0

0

0

1

1

0

0

0

0

1

1

0

Input

```
10
8 1 7 5 9 6 8 2 1 3
```



Correct output

```
0
0
0
0
0
0
0
1
0
1
0
```



User output

```
0
0
0
0
0
0
0
1
0
1
0
```



## Bài toán: Tính tổng cặp dãy bằng một số

Cho dãy  $a_1, a_2, \dots, a_n$  trong đó các phần tử đôi một khác nhau và 1 giá trị nguyên dương  $M$ . Hãy đếm số  $Q$  các cặp  $(i, j)$  sao cho  $1 \leq i < j \leq n$  và  $a_i + a_j = M$ .

### Dữ liệu

- Dòng 1: ghi  $n$  và  $M$  ( $1 \leq n, M \leq 1000000$ )
- Dòng 2: ghi  $a_1, a_2, \dots, a_n$

### Kết quả

Ghi ra giá trị  $Q$

## Source code

```
//C++
#include <bits/stdc++.h>
#include <unordered_set>
using namespace std;
unordered_set<int> all;
int main()
{
    ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
    int n,m;
    cin >> n >> m;
    int dem=0;
    for (int i = 0; i < n; i++) {
        int j;
        cin >> j;
        auto t = all.find(m-j);
        if (t != all.end()) {
            dem++;
        }

        all.insert(j);
    }
    cout << dem;
    return 0;
}
```

## Input

```
5 6
5 2 1 4 3
```

## Correct output

```
2
```

## User output

```
2
```

Input

1000 1500

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42

Correct output

250

User output

250