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

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

Điểm thưởng: $5\% \rightarrow 10\%$ (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, 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

| Bài thực hành số 1 – Tuần 31 | Error! Bookmark not defined. | |
|---|------------------------------|--|
| Bài tập 1.Tính tổng mảng n phần tử | Error! Bookmark not defined. | |
| Bài tập 2. Liệt kê dãy số nguyên có 3 chữ số chia hết cho n | 4 | |
| Bài tập 3. So sánh chênh lệch giá điện | Error! Bookmark not defined. | |
| Bài tập 4. Chuyển đổi Text thành in hoa | 7 | |
| Bài tập 5. Trích xuất ngày tháng năm từ chuỗi | 9 | |
| Bài tập 6. Liệt kê và tỉnh bình phương các số từ 1 tới n | 11 | |
| Bài tập 7. Đếm số lẻ và số chẵn trong mỗi dãy | 12 | |
| Bài tập 8. Cộng trừ nhân chia của A và B | 14 | |
| Bài tập 9. Chuyển đổi giờ phút giây thành giây | 16 | |
| Bài tập 10. Giải phương trình bậc 2 | 17 | |
| Bài tập 11. Truy vấn cơ bản trên mảng; | 19 | |
| Bài tập 12. Chuỗi k chẵn | 22 | |
| Bài tập 13. Đếm số từ | 24 | |
| Bài tâp 14. Thay thế văn bản | 25 | |

Cho dãy số nguyên a1, a2, ..., an. Tính tổng Q các phần tử của dãy này.

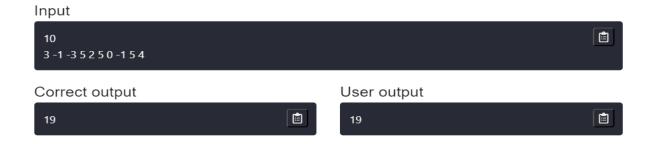
Đầu vào

- Dòng 1: chứa n (1 <= n <= 10000)
- Dòng 2: chứa a1, a2, ..., an (-10000 <= ai <= 10000)

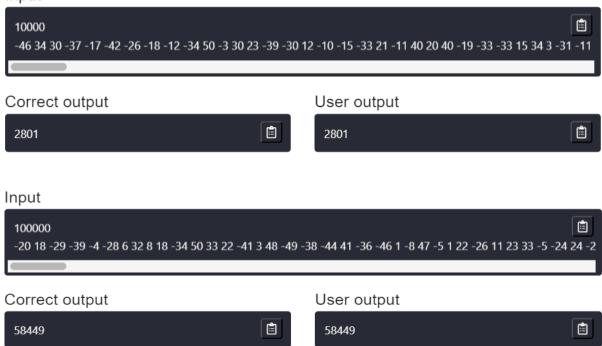
đầu ra

Viết giá trị của Q

Test:



Input



Bài toán 2: Liệt kê dãy số nguyên có 3 chữ số chia hết cho n

Cho số nguyên dương n, tìm mọi số nguyên có 3 chữ số chia hết cho n.

Đầu vào

Dòng 1: chứa số nguyên dương n (1 <= n <= 999)

đầu ra

Viết dãy số tìm được (các phần tử cách nhau bằng ký tự SPACE)

```
#PYTHON
#Mai Minh Hoàng
#20215381
#khai báo n81 và nhập từ bàn phím,chuyển n81 sang số nguyên
n81 = int(input())
#khai báo mảng m81

m81 = []
#duyệt qua tất cả các số có 3 chữ số
for i81 in range(100, 1000):
#nếu số đó chia hết cho n81 thì thêm vào mảng m81
if i81 % n81 == 0: m81.append(i81)
#duyệt qua tất cả phần tử mảng m81
#duyệt qua tất cả phần tử mảng m81
#in ra màn hình, cách nhau bởi khoảng trắng
print(i81, end=' ')
```

Test:



Bài toán 3: So sánh chênh lệch giá điện theo đề xuất mới của EVS

Hiện tại giá điện đang được tính theo cấp bậc gồm 6 cấp (điều chỉnh từ 05/04/2023), với giá thấp nhất (bậc 1) là 1.728 VNĐ và giá cấp cao nhất là 3.015 VNĐ. Tuy nhiên, EVEN đang đề xuất một cách tính giá điện bậc thang mới chỉ bao gồm 5 cấp độ như hình dưới, giá thấp nhất (bậc 1) khoảng 1,728 đồng một kWh và cao nhất (bậc 5) là 3,457 đồng một kWh.

Biểu giá hành động Phương án 5 bậc

Mức sử dụng Giá (*) Mức sử dụng Giá

1 0-50 kWh 1,728 0-100 kWh 1,728

2 51-100 kWh 1,786 101-200 kWh 2,074

- 3 101-200 kWh 2,074 201-400 kWh 2,612
- 4 201-300 kWh 2,612 401-700 kWh 3,111
- 5 301-400 kWh 2.919 701 kWh trở lên 3.457
- 6 401 kWh trở lên 3,015
- (*) Giá chưa bao gồm thuế GTGT

Hãy xây dựng chương trình nhập số kwh điện của một hộ tiêu thụ và so sánh xem bảng giá mới đang lấy ý kiến trúc thì tiền điện thụ của đó sẽ tăng thêm hay giảm đi bao nhiêu.

INPUT: là số kwh điện thụ động của hộ (là số nguyên)

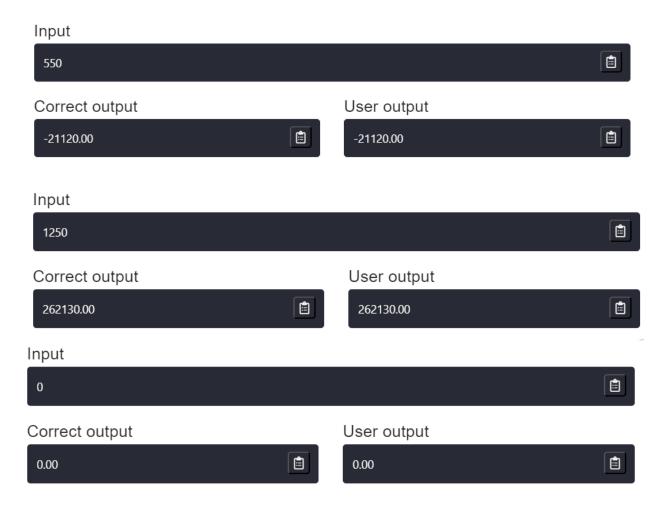
OUTPUT: là chênh lệch giữa giá mô tả và giá theo cấp bậc 6 đang được áp dụng.

VAT sẽ được thu là 10% và kết quả sẽ được lấy tới 2 phân tích chữ số Source code

```
CPP ▼
```

```
#include <br/>
#inclu
```

Test:



Vấn đề 4: Chuyển đổi TEXT thành chữ hoa

Cho một TEXT, hãy viết chương trình chuyển đổi TEXT thành chữ hoa.

Đầu vào

VĂN BẢN

đầu ra

TEXT trong đó các ký tự được chuyển đổi thành chữ hoa

Test:



Vấn đề 5: Trích xuất Năm, Tháng, Ngày từ Chuỗi YYYY-MM-DD

Cho một ngày là một chuỗi có định dạng YYYY-MM-DD (trong đó YYYY là năm, MM là tháng (tháng í từ 1 đến 12) và DD là ngày (ngày từ 1 đến 31))). Trích xuất năm, tháng và ngày.

Đầu vào

• Dòng 1: chứa chuỗi s

đầu ra

 nếu s không có định dạng YYYY-MM-DD thì viết KHÔNG ĐÚNG. Nếu không, hãy viết năm, tháng và ngày cách nhau bằng ký tự SPACE

Source code

```
#include <bits/stdc++.h>
    using namespace std;
//hàm kiểm tra ký tự có phải là số từ 0 tới 9 không
 8 bool check(char a_81)
 9 - {
              //Ascii thì trả về true

if (a_81 >= '0' && a_81 <= '9')return true;
//ngược lại trả về false
13
              return false;
      }
     int main()
17 - {
              string s_81;
              cin >> s_81;
              if (s_81.size() != 10) cout << "INCORRECT";
//neu co độ dài là 10
else</pre>
              {
                     (s_81[4] != '-' || s_81[7] != <u>'-'</u>)
                       cout << "INCORRECT";</pre>
                 //duyệt qua tất cả ký tự trong xâu đã nhập
for (int i_81 = 0; i_81 < 10; i_81++)
                       //bỏ qua ký tự thứ 4 và 7 là ký tự -
if ((i_81 != 4) && (i_81 != 7))
                             //kiểm tra xem ký tự đó có phải là chữ số không if (check(s_81[i_81]) == false)
                                   //néu khác thì in ra INCORRECT
cout << "INCORRECT";</pre>
                //chuyển 4 ký tự đầu tiên của xâu thành 1 số 4 chữ số giá trị là năm
int y_81 = (s_81[0] - '0') * 1000 + (s_81[1] - '0') * 100 + (s_81[2] - '0') * 10 + s_81[3] - '0';
//tương tự ở trên nhưng lấy ra tháng
int m_81 = (s_81[5] - '0') * 10 + s_81[6] - '0';
```

```
//chuyển 4 ký tự đầu tiên của xâu thành 1 số 4 chữ số giá trị là nằm
int y_81 = (s_81[0] - '0') * 1000 + (s_81[1] - '0') * 100 + (s_81[2] - '0') * 10 + s_81[3] - '0';

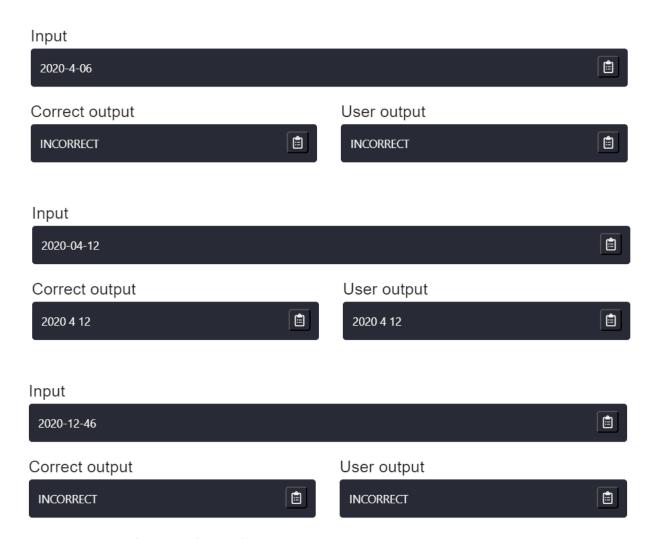
//tương tự ở trên nhưng lấy ra tháng
int m_81 = (s_81[5] - '0') * 10 + s_81[6] - '0';

//tương tự ở trên và lấy ra ngày
int d_81 = (s_81[8] - '0') * 10 + s_81[9] - '0';

//nếu ngày có giá trị từ 1 tới 31 và tháng có giá trị từ 1 tới 12 thì in ra kết quả
if (m_81 >= 1 && m_81 <= 12 && d_81 >= 1 && d_81 <= 31) cout << y_81 << " " << m_81 << " " << d_81;

//nêu ngày tháng không nằm trong các giá trị đúng ở trên thì in ra INCORRECT
else cout << "INCORRECT";
}
return 0;
```

Test:



Bài toán 6: Liệt kê tất cả các số từ 1 đến n và các bình phương của nó

Cho một số nguyên n, in các số từ 1 đến n và các bình phương của nó.

Đầu vào

• Dòng 1: chứa số nguyên dương n (1 <= n <= 100)

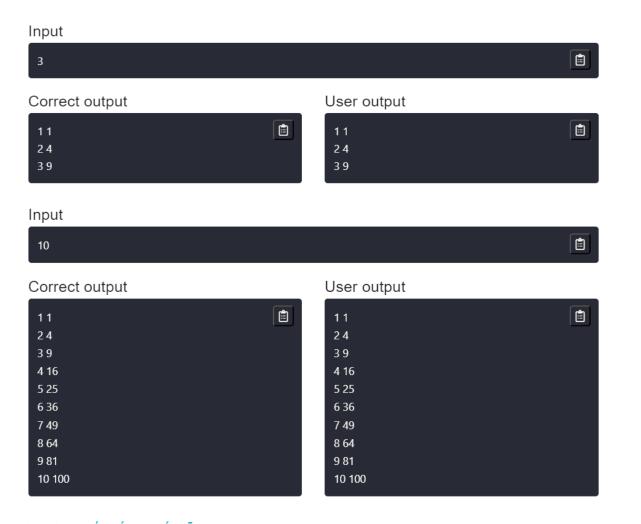
đầu ra

Mỗi dòng i (i = 1,...,n): chứa i và i^2 (các phần tử cách nhau một ký tự SPACE)

Source code

```
1 #PYTHON
2 #Mai Minh Hoàng
3 #20215381
4 #khai báo và nhập số nguyên n_81
5 n_81=int(input())
6 #duyệt tất cả các số nguyên từ 1 tới n
7 * for i_81 in range(1,n_81+1):
    #in ra mỗi số đó và bình phương của nó
9 print(i_81,i_81**2)
10
```

Test:



Bài toán 7: Đếm số lẻ và số chẵn trong một dãy

Cho dãy số nguyên a1, a2, ..., an. Đếm số phần tử lẻ và số phần tử chẵn của dãy.

Đầu vào

- Dòng 1: chứa số nguyên dương n (1 <= n <= 100000)
- Dòng 2: chứa a1, a2, ..., an. (1 <= ai <= 1000000)

đầu ra

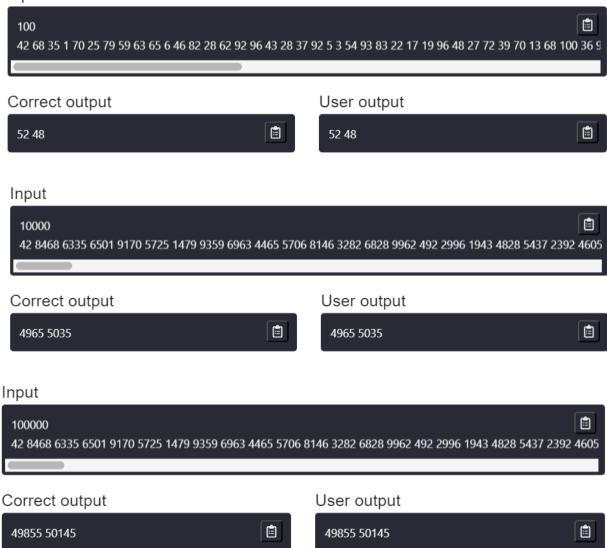
• Viết số phần tử lẻ và số phần tử chẵn (cách nhau bằng ký tự SPACE)

Source code

```
4 #include <bits/stdc++.h>
5 //khai báo không gian tên
6 using namespace std;
7 int main()
8 - {
        int n_81;
        //biến để lưu số phần tử là số lẻ
        int le_81=0;
        int chan_81 = 0;
        cin >> n_81;
        long *m_81 = new Long[n_81];
        //nhập lần lượt n_81 phần tử vào mảng m_81 for (int i_81 = 0; i_81 < n_81; i_81++) [
             cin >> m_81[i_81];
         //duyệt qua tất cả phần tử đã nhập
        for (int i_81 = 0; i_81 < n_81; i_81++) {
    //néu là số chẵn thì chan_81 cộng thêm 1
    if (m_81[i_81] % 2 == 0) {</pre>
                  chan_81 += 1;
         for (int i_81 = 0; i_81 < n_81; i_81++) {
              if (m_81[i_81] % 2 == 0) {
                   chan_81 += 1;
              else {
                   le_81 += 1;
         cout << le_81 << " " << chan_81;</pre>
         return 0;
   }
```

Test:

Input



Bài toán 8: Cộng trừ nhân chia của A và B

Cho 2 số nguyên a và b. Tính a+b, ab, a*b, a/b.

Đầu vào

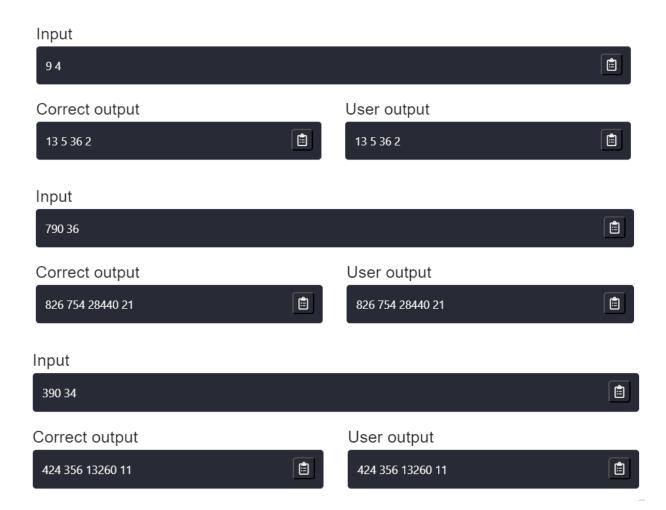
Dòng 1 chứa 2 số nguyên a và b (1 <= a,b <= 1000)

đầu ra

Viết a+b, ab, a*b, a/b (4 số nguyên cách nhau bằng ký tự SPACE)

Source code

Test:



Vấn đề 9: Chuyển đổi hh:mm:ss thành giây

Cho một thời điểm là một chuỗi có định dạng hh:mm:ss (trong đó hh (0 <= hh <= 23) là giờ, mm (0 <= mm <= 59) là phút và ss (0 <= ss <= 59) là số thứ hai). Chuyển đổi thời điểm này thành giây (kết quả = hh*3600 + mm*60 + ss).

Đầu vào

Dòng 1: chứa chuỗi s biểu thị thời điểm.

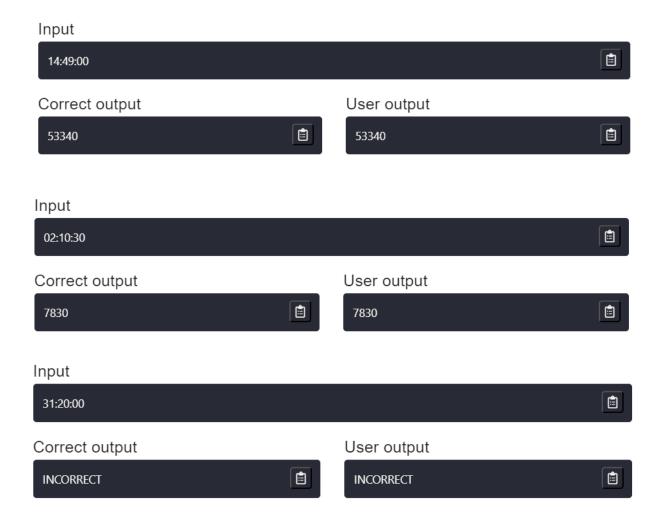
đầu ra

nếu s không có định dạng hh:mm:ss thì viết KHÔNG ĐÚNG. Ngược lại ghi giá trị đã chuyển đổi. Source code

```
//chuyển 2 ký tự đầu tiên của xâu thành 1 số 4 chữ số giá trị là giờ
int h_81 =(s_81[0] - '0') * 10 + s_81[1] - '0';
//tương tự ở trên nhưng lấy ra phút
int m_81 = (s_81[3] - '0') * 10 + s_81[4] - '0';
//tương tự ở trên và lấy ra giây
int ss_81 = (s_81[6] - '0') * 10 + s_81[7] - '0';
//nếu giờ có giá trị từ 0 tới 23 và phút.giây có giá trị từ 0 tới 59 thì in ra tổng số giây
if (m_81 >= 0 && m_81 <= 59 && ss_81 >= 0 && ss_81 <= 59 && h_81 >= 0 && h_81 <= 23)

cout << h_81*3600+m_81*60+ss_81;
//nếu giờ phút giây không nằm trong các giá trị đúng ở trên thì in ra INCORRECT
else cout << "INCORRECT";
}
return 0;
```

Test:



Bài toán 10: Giải phương trình đa thức bậc 2

Cho phương trình $ax^2 + bx + c = 0$. Tìm nghiệm của phương trình đã cho.

Đầu vào

Dòng 1 chứa 3 số nguyên dương a, b, c

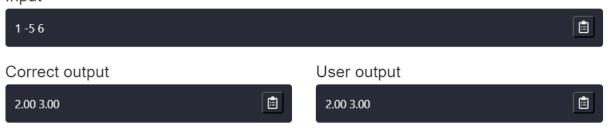
đầu ra

- Viết KHÔNG GIẢI nếu phương trình đã cho không có nghiệm
- Viết x0 (2 chữ số sau dấu thập phân) nếu phương trình đã cho có một nghiệm x0
- Viết x1, x2 với x1 < x2 (2 chữ số sau dấu thập phân) nếu phương trình đã cho có hai nghiệm phân biệt x1, x2

Source code

Test:

Input





Vấn đề 11: Truy vấn cơ bản trên mảng

Cho dãy số nguyên a1, a2, ..., an. Thực hiện một chuỗi truy vấn trên chuỗi này bao gồm:

- find-max: trả về phần tử lớn nhất của dãy đã cho
- find-min: trả về phần tử nhỏ nhất của dãy đã cho
- sum: trả về tổng các phần tử của dãy đã cho
- find-max-segment ij: trả về phần tử lớn nhất của dãy con từ chỉ số i đến chỉ số j (i <= j)

Đầu vào

- Khối đầu tiên chứa thông tin về chuỗi đã cho với đinh dang sau:
 - o Dòng 1: chứa số nguyên dương n (1 <= n <= 10000)
 - o Dòng 2: chứa n số nguyên a1, a2, ..., an (-1000 <= ai <= 1000)
 - o Khối đầu tiên được kết thúc bởi ký tự *
- Khối thứ hai chứa một chuỗi các truy vấn được xác định ở trên, mỗi truy vấn nằm trên một dòng. Khối thứ hai được kết thúc bằng 3 ký tự ***

đầu ra

Viết kết quả của mỗi truy vấn vào một dòng tương ứng

Source code

```
4 #include <bits/stdc++.h>
5 #include <vector> // Sử dụng để làm việc với vector.
6 #include <sstream> // Sử dụng để thực hiện phân tích cú pháp chuỗi.
8 using namespace std; // Sử dụng không gian tên
10 int sum_81(const vector<int>& mang_81) {
11    int s_81 = 0;
12    for (int i_81 : mang_81) {
13        s_81 += i_81;
          return s_81;
16 }
int findMaxSegment_81(const vector<int>& mang_81, int start_81, int end_81) {
          int max_81 = mang_81[start_81 - 1];
for (int i_81 = start_81 - 1; i_81 < end_81; i_81++) {
    if (mang_81[i_81] > max_81) {
                   max_81 = mang_81[i_81];
               }
               }
         return max_81;
    }
    int main() {
         int n_81;
         cin >> n 81; // Nhập số lượng phần tử trong vector.
         int max_81 = -100; // Khởi tạo max_81 với giá trị ban đầu.
         int min_81 = 100; // Khởi tạo min_81 với giá trị ban đầu.
         vector int> mang_81; // Khởi tạo vector mang_81 để lưu trữ các số nguyên.
         for (int i_81 = 0; i_81 < n_81; i_81++) {
               int t_81;
               cin >> t_81; // Nhập từng phần tử của vector mang_81.
               mang_81.push_back(t_81); // Thêm phần tử vào vector.
               // Tìm giá trị lớn nhất và nhỏ nhất trong vector mang_81.
               if (t_81 > max_81) {
                   max_81 = t_81;
               if (t_81 < min_81) {</pre>
                    min_81 = t_81;
         }
```

```
vector<string> lists; // Khởi tạo vector để lưu trữ danh sách các lệnh.
        while (true) {
             string s;
             getline(cin, s); // Đọc một dòng từ đầu vào.
             if (s == "*" || s.empty()) {
                 continue; // Nếu dòng là "*" hoặc rỗng, tiếp tục vòng lặp.
             else if (s == "***") {
                 break; // Nếu đồng là "***", thoát khỏi vòng lặp.
             }
            lists.push_back(s); // Thêm dòng vào danh sách lệnh.
         for (const string& s : lists) {
             if (s == "find-max") {
                 cout << max_81 << endl; // In ra giá trị lớn nhất trong vector mang_81.
             else if (s == "find-min") {
                 cout << min_81 << endl; // In ra giá tri nhỏ nhất trong vector mang_81.</pre>
             }
            else if (s.find("find-max-segment") < 999999) {</pre>
                istringstream iss(s);
                string cmd;
                int start, end;
               //lấy ra giá trị
iss >> cmd >> start >> end;
               cout << findMaxSegment_81(mang_81, start, end) << endl; // Goi hàm để tìm giá trị lớn nhất
           }
else if (s == "sum") {
               cout << sum_81(mang_81) << endl; // In ra tổng các phần tử trong vector mang_81.</pre>
        }
93 }
```

Test:

Input



Vấn đề 12: Chuỗi k chẵn

Cho dãy số nguyên a1, a2, . . ., MỘT. Dãy con k được định nghĩa là một dãy gồm k phần tử liên tiếp: ai, ai+1, . . ., ai+k-1. Trọng số của dãy con k là tổng các phần tử của nó.

Cho các số nguyên dương k và m. Tính số Q của dãy k sao cho trọng số là chẵn.

Đầu vào

- Dòng 1: chứa 2 số nguyên dương n, k (1 <= n <= 100000, 1 <= k <= n/2)
- Dòng 2: chứa a1, a2, . . ., MỘT. (1 <= ai <= 10000)

đầu ra

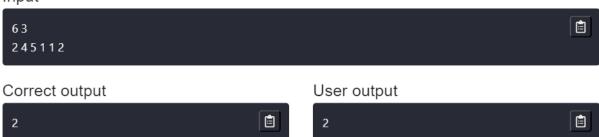
Viết giá trị Q

Source code

```
2 #Mai Minh Hoàng
 5 n_81, k_81 = map(int, input().split())
8 mang_81 = list(map(int, input().split()))
11 even_count_81 = 0
14 sum_81 = sum(mang_81[:k_81])
17 if sum_81 % 2 == 0:
       even_count_81 += 1
21 for i_81 in range(k_81, n_81):
        sum_81 -= mang_81[i_81 - k_81]
21 for i_81 in range(k_81, n_81):
       sum_81 -= mang_81[i_81 - k_81]
       sum_81 += mang_81[i_81]
       if sum_81 % 2 == 0:
          even_count_81 += 1
33 print(even_count_81)
```

Test:

Input



Input



Vấn đề 13: Đếm số từ

Cho một Text, viết chương trình đếm số Q từ (bỏ qua các ký tự SPACE, TAB, LineBreak) của Text này

Đầu vào

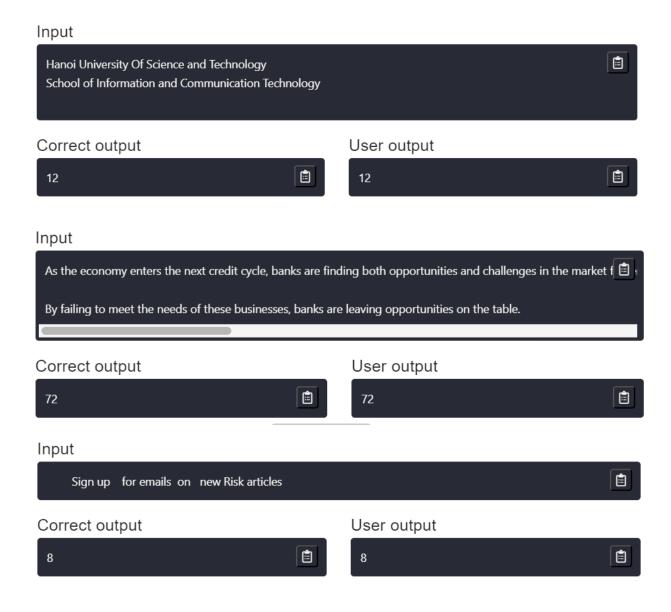
văn bản

đầu ra

Viết số Q của từ

Source code

Test:



Vấn đề 14: Thay thế văn bản

Cho văn bản T và 2 mẫu P1, P2 đều là các ký tự chuỗi (không chứa các ký tự xuống dòng, độ dài không vượt quá 1000). Hãy thay thế các P1 trong T bằng xâu P2.

Data

· Dòng 1: xâu P1

· Dòng 2: xâu P2

· Dòng 3: văn bản T

Kết quả :

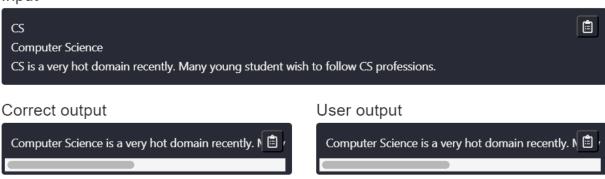
· Ghi văn bản sau khi thay thế

Source code

```
1 #PYTHON
2 #Mai Minh Hoàng
3 #20215381
4 #nhập xâu P1
5 P1=input()
6 #nhập xâu P2
7 P2=input()
8 #nhập xâu từ bàn phím
9 # và thay thế tất cả xâu P1 trong xâu đó thành P2 và in ra màn hình
10 print(input().replace(P1,P2))
```

Test:

Input



Input

