

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, 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 tính 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 tập 14. Thay thế văn bản.	25

Bài toán 1: Tính tổng mảng

Cho dãy số nguyên a_1, a_2, \dots, a_n . Tính tổng Q các phần tử của dãy này.

Đầu vào

- Dòng 1: chứa n ($1 \leq n \leq 10000$)
- Dòng 2: chứa a_1, a_2, \dots, a_n ($-10000 \leq a_i \leq 10000$)

đầu ra

- Viết giá trị của Q

```

3 //20215381
4 #include <bits/stdc++.h>
5
6 int main()
7 {
8     //khai báo biến nguyên n81 là số phần tử
9     int n81;
10    //khai báo s81 lưu tổng tất cả phần tử
11    long s81=0;
12    //nhập vào số phần tử
13    std::cin>>n81;
14    //khai báo mảng động chứa n81 phần tử
15    int *mang81=new int[n81];
16    //nhập từng phần tử trong mảng
17    for(int i81=0;i81<n81;i81++){
18        std::cin>>mang81[i81];
19    }
20 }
21 //duyet qua tất cả phần tử và tính tổng của chúng, lưu vào biến s81
22 for(int i81=0;i81<n81;i81++){
23     s81+=mang81[i81];
24 }
25 //in kết quả
26 std::cout<<s81;
27 return 0;
28 }
29

```

Test:

Input

10
3 -1 -3 5 2 5 0 -1 5 4

Correct output

19

User output

19

Input

10000

-46 34 30 -37 -17 -42 -26 -18 -12 -34 50 -3 30 23 -39 -30 12 -10 -15 -33 21 -11 40 20 40 -19 -33 -33 15 34 3 -31 -11

Correct output

2801

User output

2801

Input

100000

-20 18 -29 -39 -4 -28 6 32 8 18 -34 50 33 22 -41 3 48 -49 -38 -44 41 -36 -46 1 -8 47 -5 1 22 -26 11 23 33 -5 -24 24 -2

Correct output

58449

User output

58449

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 ($1 \leq n \leq 999$)

đầu ra

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

```
1 #PYTHON
2 #Mai Minh Hoàng
3 #20215381
4 #khai báo n81 và nhập từ bàn phím, chuyển n81 sang số nguyên
5 n81 = int(input())
6 #khai báo mảng m81
7 m81 = []
8 #duyet qua tất cả các số có 3 chữ số
9 for i81 in range(100, 1000):
10 #nếu số đó chia hết cho n81 thì thêm vào mảng m81
11     if i81 % n81 == 0: m81.append(i81)
12 #duyet qua tất cả phần tử mảng m81
13 for i81 in m81:
14 # in ra màn hình, cách nhau bởi khoảng trắng
15     print(i81, end=' ')
16
```

Test:

Input

400



Correct output

400 800



User output

400 800



Input

7



Correct output

105 112 119 126 133 140 147 154 161 168 175 182



User output

105 112 119 126 133 140 147 154 161 168 175 182



Input

17



Correct output

102 119 136 153 170 187 204 221 238 255 272 289



User output

102 119 136 153 170 187 204 221 238 255 272 289

**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, EVN 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 ▾

```
3 //20215381
4 #include <bits/stdc++.h>
5 //khai báo không gian tên
6 using namespace std;
7 //hàm tính tổng tiền phải trả sau khi thay đổi cách tính tiền với tham số đầu vào n81 là số kWh điện đã dùng
8 float totalsau(int n81){
9     //với số kWh <=100 thì tính ra giá tiền phải trả sau khi đổi cách tính
10     if(n81<=100){
11         return (float)n81*1728*1.1;
12     }
13     //với số kWh <=200 và >100 thì tính ra giá tiền phải trả sau khi đổi cách tính
14     else if(n81<=200){
15         return (float)(n81-100)*2074*1.1+totalsau(100);
16     }
17     //với số kWh <=400 và >200 thì tính ra giá tiền phải trả sau khi đổi cách tính
18     else if(n81<=400){
19         return (float)(n81-200)*2612*1.1+totalsau(200);
20     }
21     //với số kWh <=700 và >400 thì tính ra giá tiền phải trả sau khi đổi cách tính
22     else if(n81<=700){
23         return (float)(n81-400)*3111*1.1+totalsau(400);
24     }
25     //với số kWh >700 thì tính ra giá tiền phải trả sau khi đổi cách tính
26     else{
27         return (float)(n81-700)*3457*1.1+totalsau(700);
28     }
29 }
```

```
26 //hàm tính tổng tiền phải trả trước khi thay đổi cách tính tiền với tham số đầu vào n81 là số kWh điện đã dùng
27 float totaltruoc(int n81){
28     //với số kWh <=50 thì tính ra giá tiền phải trả trước khi đổi cách tính
29     if(n81<=50){
```

```

30     return (float)n81*1728*1.1;
31     //với số kwh >50 và <=100 thì tính ra giá tiền phải trả trước khi đổi cách tính
32 }else if(n81<=100){
33     return (float)(n81-50)*1786*1.1+totaltruoc(50);
34     //với số kwh >100 và <=200 thì tính ra giá tiền phải trả trước khi đổi cách tính
35 }else if(n81<=200){
36     return (float)(n81-100)*2074*1.1+totaltruoc(100);
37     //với số kwh >200 và <=300 thì tính ra giá tiền phải trả trước khi đổi cách tính
38 }else if(n81<=300){
39     return (float)(n81-200)*2612*1.1+totaltruoc(200);
40     //với số kwh >300 và <=400 thì tính ra giá tiền phải trả trước khi đổi cách tính
41 }else if(n81<=400){
42     return (float)(n81-300)*2919*1.1+totaltruoc(300);
43     //với số kwh >400 thì tính ra giá tiền phải trả trước khi đổi cách tính
44 }else{
45     return (float)(n81-400)*3015*1.1+totaltruoc(400);
46 }
47 }
48 //hàm chính
49 int main()
50 {
51     //khai báo và nhập số kwh đã dùng
52     int n81;
53     cin>>n81;
54     //in ra màn hình kết quả sau khi lấy giá tiền nếu tính cách tính mới trừ đi giá tiền nếu tính cách cũ
55     printf("%.2f",totalsau(n81)-totaltruoc(n81));

```

Test:

Input

550

Correct output

-21120.00

User output

-21120.00

Input

1250

Correct output

262130.00

User output

262130.00

Input

0

Correct output

0.00

User output

0.00

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

Source code

JAVA ▾

```
2 //Mai Minh Hoàng 20215381
3 import java.util.*;
4
5 @SuppressWarnings({"unchecked", "deprecation"})
6 public class Main {
7     //hàm main để chạy chương trình
8     public static void main(String[] args) {
9         //khai báo sc81 để đọc dữ liệu từ bàn phím
10        Scanner sc81= new Scanner(System.in);
11        //biến s81 để lưu trữ kết quả cuối cùng
12        String s81="";
13        //mỗi khi còn có dòng kế tiếp
14        while(sc81.hasNext())
15        {
16            //biến ss81 để nhận giá trị từng dòng text và chuyển nó thành in hoa
17            String ss81=sc81.nextLine().toUpperCase();
18            //gán thêm giá trị từng dòng vừa rồi vào kết quả cuối cùng và xuống hàng
19            s81+=ss81+'\n';
20            //nếu gặp phải dòng null(không còn bất kỳ giá trị nào nữa) thì kết thúc vòng lặp
21            if(ss81==null) {
22                break;
23            }
24        }
25        //in ra kết quả là văn bản đã in hoa
26        System.out.println(s81);
27    }
28 }
```

Test:

Input

What do we mean by skinny design?

Benedict Sheppard: Happy to start there. A skinny design is perhaps too complex a term because we're simply sayi

Christian Johnson: Thank you, Ben. So what I'm hearing is a big cost component, and I'm sure in the current enviroi

Correct output

WHAT DO WE MEAN BY SKINNY DESIGN?

BENEDICT SHEPPARD: HAPPY TO START THERE. A SKI

CHRISTIAN JOHNSON: THANK YOU, BEN. SO WHAT I

User output

WHAT DO WE MEAN BY SKINNY DESIGN?

BENEDICT SHEPPARD: HAPPY TO START THERE. A SKI

CHRISTIAN JOHNSON: THANK YOU, BEN. SO WHAT I

Input

Additionally, we will highlight the importance of cross-functional collaboration in product design and the need for sta

Correct output

ADDITIONALLY, WE WILL HIGHLIGHT THE IMPORTANC

User output

ADDITIONALLY, WE WILL HIGHLIGHT THE IMPOR

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

```

3 //20215381
4 #include <bits/stdc++.h>
5 //Khai báo không gian tên
6 using namespace std;
7 //hàm kiểm tra ký tự có phải là số từ 0 tới 9 không
8 bool check(char a_81)
9 {
10     //nếu ký tự a_81 có số thứ tự lớn hơn hoặc bằng thứ tự của ký tự 0 và bé hơn bằng thứ t
11     //Ascii thì trả về true
12     if (a_81 >= '0' && a_81 <= '9') return true;
13     //ngược lại trả về false
14     return false;
15 }
16 int main()
17 {
18     //khai báo chuỗi s_81 chứa xâu cần định dạng
19     string s_81;
20     //nhập vào xâu
21     cin >> s_81;
22     //nếu xâu vừa nhập có độ dài khác 10 thì in ra INCORRECT
23     if (s_81.size() != 10) cout << "INCORRECT";
24     //nếu có độ dài là 10
25     else
26     {
27         //nếu ký tự thứ 4 hoặc thứ 7 trong xâu khác ký tự - thì in ra INCORRECT
28         if (s_81[4] != '-' || s_81[7] != '-')
29         {
30             cout << "INCORRECT";
31             return 0;
32         }
33         //duyet qua tất cả ký tự trong xâu đã nhập
34         for (int i_81 = 0; i_81 < 10; i_81++)
35         {
36             //bỏ qua ký tự thứ 4 và 7 là ký tự -
37             if ((i_81 != 4) && (i_81 != 7))
38             {
39                 //kiểm tra xem ký tự đó có phải là chữ số không
40                 if (check(s_81[i_81]) == false)
41                 {
42                     //nếu khác thì in ra INCORRECT
43                     cout << "INCORRECT";
44                     return 0;
45                 }
46             }
47         }
48         //chuyển 4 ký tự đầu tiên của xâu thành 1 số 4 chữ số giá trị là năm
49         int y_81 = (s_81[0] - '0') * 1000 + (s_81[1] - '0') * 100 + (s_81[2] - '0') * 10 + s_81[3] - '0';
50         //tương tự ở trên nhưng lấy ra tháng
51         int m_81 = (s_81[5] - '0') * 10 + s_81[6] - '0';
52         //tương tự ở trên và lấy ra ngày

```

```

48 //chuyển 4 ký tự đầu tiên của xâu thành 1 số 4 chữ số giá trị là năm
49 int y_81 = (s_81[0] - '0') * 1000 + (s_81[1] - '0') * 100 + (s_81[2] - '0') * 10 + s_81[3] - '0';
50 //tương tự ở trên nhưng lấy ra tháng
51 int m_81 = (s_81[5] - '0') * 10 + s_81[6] - '0';
52 //tương tự ở trên và lấy ra ngày
53 int d_81 = (s_81[8] - '0') * 10 + s_81[9] - '0';
54 //nếu ngày có giá trị từ 1 tới 31 và tháng có giá trị từ 1 tới 12 thì in ra kết quả
55 if (m_81 >= 1 && m_81 <= 12 && d_81 >= 1 && d_81 <= 31) cout << y_81 << " " << m_81 << " " << d_81;
56 //nếu ngày tháng không nằm trong các giá trị đúng ở trên thì in ra INCORRECT
57 else cout << "INCORRECT";
58 }
59 return 0;
60 }

```

Test:

Input

2020-4-06

Correct output

INCORRECT

User output

INCORRECT

Input

2020-04-12

Correct output

2020 4 12

User output

2020 4 12

Input

2020-12-46

Correct output

INCORRECT

User output

INCORRECT

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 ($1 \leq n \leq 100$)

Đầu ra

- Mỗi dòng i ($i = 1, \dots, n$): 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 #khởi 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):
8     #in ra mỗi số đó và bình phương của nó
9     print(i_81,i_81**2)
10 |
```

Test:

Input

3

Correct output

1 1
2 4
3 9

User output

1 1
2 4
3 9

Input

10

Correct output

1 1
2 4
3 9
4 16
5 25
6 36
7 49
8 64
9 81
10 100

User output

1 1
2 4
3 9
4 16
5 25
6 36
7 49
8 64
9 81
10 100

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

Cho dãy số nguyên a_1, a_2, \dots, a_n . Đếm số phần tử lẻ và số phần tử chẵn của dãy.

Đầu vào

- Dòng 1: chứa số nguyên dương n ($1 \leq n \leq 100000$)
- Dòng 2: chứa a_1, a_2, \dots, a_n . ($1 \leq a_i \leq 1000000$)

đầu ra

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

Source code

```

2 //Mai Minh Hoàng
3 //20215381
4 #include <bits/stdc++.h>
5 //khai báo không gian tên
6 using namespace std;
7 int main()
8 {
9     //khai báo số phần tử
10    int n_81;
11    //biến để lưu số phần tử là số lẻ
12    int le_81=0;
13    //biến để lưu số phần tử là số chẵn
14    int chan_81 = 0;
15    //nhập số phần tử
16    cin >> n_81;
17    //khai báo mảng để chứa n_81 phần tử
18    long *m_81 = new long[n_81];
19    //nhập lần lượt n_81 phần tử vào mảng m_81
20    for (int i_81 = 0; i_81 < n_81; i_81++) {
21        cin >> m_81[i_81];
22    }
23    //duyet qua tất cả phần tử đã nhập
24    for (int i_81 = 0; i_81 < n_81; i_81++) {
25        //nếu là số chẵn thì chan_81 cộng thêm 1
26        if (m_81[i_81] % 2 == 0) {
27            chan_81 += 1;
28        }
29        //duyet qua tất cả phần tử đã nhập
30        for (int i_81 = 0; i_81 < n_81; i_81++) {
31            //nếu là số chẵn thì chan_81 cộng thêm 1
32            if (m_81[i_81] % 2 == 0) {
33                chan_81 += 1;
34            }
35            //nếu là số lẻ thì le_81 cộng thêm 1
36            else {
37                le_81 += 1;
38            }
39        }
40        //in ra số phần tử là số chẵn và số phần tử là số lẻ
41        cout << le_81 << " " << chan_81;
42        return 0;
43    }
44 }

```

Test:

Input

100

42 68 35 1 70 25 79 59 63 65 6 46 82 28 62 92 96 43 28 37 92 5 3 54 93 83 22 17 19 96 48 27 72 39 70 13 68 100 36 9

Correct output

52 48

User output

52 48

Input

10000

42 8468 6335 6501 9170 5725 1479 9359 6963 4465 5706 8146 3282 6828 9962 492 2996 1943 4828 5437 2392 4605

Correct output

4965 5035

User output

4965 5035

Input

100000

42 8468 6335 6501 9170 5725 1479 9359 6963 4465 5706 8146 3282 6828 9962 492 2996 1943 4828 5437 2392 4605

Correct output

49855 50145

User output

49855 50145

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 ($1 \leq a, b \leq 1000$)

Đầu ra

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

Source code

```

1 //CPP
2 //Mai Minh Hoàng
3 //20215381
4 #include <bits/stdc++.h>
5 //khai báo không gian tên biến
6 using namespace std;
7 int main() {
8     //khai báo 2 số a,b
9     int a_81, b_81;
10    //nhập a và b
11    cin >> a_81 >> b_81;
12    //in ra kết quả
13    cout << a_81 + b_81 << " " << a_81 - b_81 << " " << a_81 * b_81 << " " << a_81 / b_81;
14    return 0;
15 }
16

```

Test:

Input

9 4



Correct output

13 5 36 2



User output

13 5 36 2



Input

790 36



Correct output

826 754 28440 21



User output

826 754 28440 21



Input

390 34



Correct output

424 356 13260 11



User output

424 356 13260 11



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 ($0 \leq hh \leq 23$) là giờ, mm ($0 \leq mm \leq 59$) là phút và ss ($0 \leq ss \leq 59$) 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

```

1 //CPP
2 //Mai Minh Hoàng
3 //20215381
4 #include <bits/stdc++.h>
5 //Khai báo không gian tên
6 using namespace std;
7 //hàm kiểm tra ký tự có phải là số từ 0 tới 9 không
8 bool check(char a_81)
9 {
10     //nếu ký tự a_81 có số thứ tự lớn hơn hoặc bằng thứ tự của ký tự 0 và bé hơn bằng thứ tự ký tự 9 thì
11     //Ascii thì trả về true
12     if (a_81 >= '0' && a_81 <= '9') return true;
13     //ngược lại trả về false
14     return false;
15 }
16 int main()
17 {
18     //khai báo chuỗi s_81 chứa xâu cần định dạng
19     string s_81;
20     //nhập vào xâu
21     cin >> s_81;
22     //nếu xâu vừa nhập có độ dài khác 8 thì in ra INCORRECT
23     if (s_81.size() != 8) cout << "INCORRECT";
24     //nếu có độ dài là 10
25     else
26     {
27         //nếu ký tự thứ 2 hoặc thứ 5 trong xâu khác ký tự : thì in ra INCORRECT
28         if (s_81[2] != ':' || s_81[5] != ':')
29         {
30             cout << "INCORRECT";
31             return 0;
32         }
33         //duyệt qua tất cả ký tự trong xâu đã nhập
34         for (int i_81 = 0; i_81 < 8; i_81++)
35         {
36             //bỏ qua ký tự thứ 2 và 5 là ký tự -
37             if ((i_81 != 2) && (i_81 != 5))
38             {
39                 //kiểm tra xem ký tự đó có phải là chữ số không
40                 if (check(s_81[i_81]) == false)
41                 {
42                     //nếu khác thì in ra INCORRECT
43                     cout << "INCORRECT";
44                     return 0;
45                 }
46             }
47         }
48     }
49 }

```



```

47     }
48     //chuyển 2 ký tự đầu tiên của xâu thành 1 số 4 chữ số giá trị là giờ
49     int h_81 = (s_81[0] - '0') * 10 + s_81[1] - '0';
50     //tương tự ở trên nhưng lấy ra phút
51     int m_81 = (s_81[3] - '0') * 10 + s_81[4] - '0';
52     //tương tự ở trên và lấy ra giây
53     int ss_81 = (s_81[6] - '0') * 10 + s_81[7] - '0';
54     //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
55     if (m_81 >= 0 && m_81 <= 59 && ss_81 >= 0 && ss_81 <= 59 && h_81 >= 0 && h_81 <= 23)
56         cout << h_81*3600+m_81*60+ss_81;
57     //nếu giờ phút giây không nằm trong các giá trị đúng ở trên thì in ra INCORRECT
58     else cout << "INCORRECT";
59 }
60 return 0;
61 }

```

Test:

Input

14:49:00

Correct output

53340

User output

53340

Input

02:10:30

Correct output

7830

User output

7830

Input

31:20:00

Correct output

INCORRECT

User output

INCORRECT

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 $x_1 < x_2$ (2 chữ số sau dấu thập phân) nếu phương trình đã cho có hai nghiệm phân biệt x1, x2

Source code

```
7 int main() {
8     //khai báo lần lượt các hệ số của biểu thức
9     int a_81, b_81, c_81;
10    //nhập vào các số đó
11    cin >> a_81 >> b_81 >> c_81;
12    //tính delta của pt bậc 2
13    double delta_81 = b_81 * b_81 - 4 * a_81 * c_81;
14    //delta<0 thì phương trình vô nghiệm
15    if (delta_81 < 0) {
16
17        cout << "NO SOLUTION" << endl;
18    }
19    //delta=0 thì có 1 nghiệm kép
20    else if (delta_81 == 0) {
21
22        double x0_81 = -b_81 / (2.0 * a_81);
23        //in ra lấy 2 số sau dấu phẩy
24        printf("%.2f", x0_81);
25    }
26    //delta>0 thì pt có 2 nghiệm phân biệt
27    else {
28        //nghiệm thứ 1
29        double x1_81 = (-b_81 - sqrt(delta_81)) / (2.0 * a_81);
30        //nghiệm thứ 2
31        double x2_81 = (-b_81 + sqrt(delta_81)) / (2.0 * a_81);
32        printf("%.2f %.2f", x1_81, x2_81);
33    }
```

Test:

Input

1 -5 6

Correct output

2.00 3.00

User output

2.00 3.00

Input

1 -6 9



Correct output

3.00



User output

3.00



Input

1 1 6



Correct output

NO SOLUTION



User output

NO SOLUTION



Vấn đề 11: Truy vấn cơ bản trên mảng

Cho dãy số nguyên a_1, a_2, \dots, a_n . 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 ($i \leq j$)

Đầu vào

- Khối đầu tiên chứa thông tin về chuỗi đã cho với định dạng sau:
 - Dòng 1: chứa số nguyên dương n ($1 \leq n \leq 10000$)
 - Dòng 2: chứa n số nguyên a_1, a_2, \dots, a_n ($-1000 \leq a_i \leq 1000$)
 - 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

```

2 //Mai Minh Hoàng
3 //20215381
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.
7
8 using namespace std; // Sử dụng không gian tên
9
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;
14     }
15     return s_81;
16 }
17
18 int findMaxSegment_81(const vector<int>& mang_81, int start_81, int end_81) {
19     int max_81 = mang_81[start_81 - 1];
20     for (int i_81 = start_81 - 1; i_81 < end_81; i_81++) {
21         if (mang_81[i_81] > max_81) {
22             max_81 = mang_81[i_81];
23         }
24         else {
25             continue;
26         }
27     }
28     return max_81;
29 }
30
31 int main() {
32     int n_81;
33     cin >> n_81; // Nhập số lượng phần tử trong vector.
34
35     int max_81 = -100; // Khởi tạo max_81 với giá trị ban đầu.
36     int min_81 = 100; // Khởi tạo min_81 với giá trị ban đầu.
37     vector<int> mang_81; // Khởi tạo vector mang_81 để lưu trữ các số nguyên.
38
39     for (int i_81 = 0; i_81 < n_81; i_81++) {
40         int t_81;
41         cin >> t_81; // Nhập từng phần tử của vector mang_81.
42         mang_81.push_back(t_81); // Thêm phần tử vào vector.
43
44         // Tìm giá trị lớn nhất và nhỏ nhất trong vector mang_81.
45         if (t_81 > max_81) {
46             max_81 = t_81;
47         }
48         if (t_81 < min_81) {
49             min_81 = t_81;
50         }
51     }
52 }

```

```

53     vector<string> lists; // Khởi tạo vector để lưu trữ danh sách các lệnh.
54
55     while (true) {
56         string s;
57         getline(cin, s); // Đọc một dòng từ đầu vào.
58
59         if (s == "*" || s.empty()) {
60             continue; // Nếu dòng là "*" hoặc rỗng, tiếp tục vòng lặp.
61         }
62         else if (s == "***") {
63             break; // Nếu dòng là "***", thoát khỏi vòng lặp.
64         }
65
66         lists.push_back(s); // Thêm dòng vào danh sách lệnh.
67     }
68
69     for (const string& s : lists) {
70
71         if (s == "find-max") {
72             cout << max_81 << endl; // In ra giá trị lớn nhất trong vector mang_81.
73         }
74         else if (s == "find-min") {
75             cout << min_81 << endl; // In ra giá trị nhỏ nhất trong vector mang_81.
76         }
77         //tìm xem xâu s có chứa "find-max-segment" không, nếu không có nó sẽ trả lại số nguyên
78         else if (s.find("find-max-segment") < 999999) {
79             //khởi tạo đối tượng để từ find-max-segment a b tách ra a và b là start và end
80             istringstream iss(s);
81             string cmd;
82             int start, end;
83             //lấy ra giá trị
84             iss >> cmd >> start >> end;
85             cout << findMaxSegment_81(mang_81, start, end) << endl; // Gọi hàm để tìm giá trị lớn nhất
86         }
87         else if (s == "sum") {
88             cout << sum_81(mang_81) << endl; // In ra tổng các phần tử trong vector mang_81.
89         }
90     }
91
92     return 0;
93 }

```

Test:

Input

```

5
1 4 3 2 5
*
find-max
find-min
find-max-segment 1 3
find-max-segment 2 5
sum
***

```

Correct output

```

5
1
4
5
15

```

User output

```

5
1
4
5
15

```

Vấn đề 12: Chuỗi k chẵn

Cho dãy số nguyên a_1, a_2, \dots , MỘT. Dãy con k được định nghĩa là một dãy gồm k phần tử liên tiếp: $a_i, a_{i+1}, \dots, a_{i+k-1}$. 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 ($1 \leq n \leq 100000, 1 \leq k \leq n/2$)
- Dòng 2: chứa a_1, a_2, \dots , MỘT. ($1 \leq a_i \leq 10000$)

đầu ra

- Viết giá trị Q

Source code

```

1 #PYTHON
2 #Mai Minh Hoàng
3 #20215381
4 # Nhập giá trị n_81 và k_81
5 n_81, k_81 = map(int, input().split())
6
7 # Nhập mảng
8 mang_81 = list(map(int, input().split()))
9
10 # Khởi tạo biến đếm số lượng mảng con tổng chẵn
11 even_count_81 = 0
12
13 # Tính tổng k phần tử đầu tiên trong mảng mang_81
14 sum_81 = sum(mang_81[:k_81])
15
16 # Kiểm tra nếu tổng đó là số chẵn
17 if sum_81 % 2 == 0:
18     even_count_81 += 1
19
20 # Duyệt qua các mảng con
21 for i_81 in range(k_81, n_81):
22     # Trừ đi phần tử đầu tiên của mảng con trước đó
23     sum_81 -= mang_81[i_81 - k_81]
24
25 # Duyệt qua các mảng con
26 for i_81 in range(k_81, n_81):
27     # Trừ đi phần tử đầu tiên của mảng con trước đó
28     sum_81 -= mang_81[i_81 - k_81]
29
30     # Cộng thêm phần tử cuối mảng con mới
31     sum_81 += mang_81[i_81]
32
33     # Kiểm tra nếu tổng mới là số chẵn
34     if sum_81 % 2 == 0:
35         even_count_81 += 1
36
37 # In ra số lượng tổng chẵn
38 print(even_count_81)

```

Test:

Input

```
63
2 4 5 1 1 2
```

Correct output

2

User output

2

Input

```
100000 20000
3 1 2 9 8 1 8 7 7 6 5 3 6 2 6 6 5 9 8 6 5 5 8 2 3 9 3 7 2 1 8 5 4 5 9 7 8 5 10 8 8 3 3 6 6 6 7 10 2 9 7 2 4 6 7 8 3 3 7 9 9 3
```

Correct output

39835

User output

39835

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

```
1 //JAVA
2 //Mai Minh Hoàng
3 //20215381
4 import java.util.*;
5
6 @SuppressWarnings({"unchecked", "deprecation"})
7 public class Main {
8     public static void main(String[] args) {
9         //khai báo biến sc_81 để nhập dữ liệu
10        Scanner sc_81=new Scanner(System.in);
11        //khai báo biến để lưu tổng số từ
12        int dem_tong_81=0;
13        //lặp cho tới khi không còn dòng nào
14        while (sc_81.hasNextLine()) {
15            //s_81 là dòng tiếp theo để xét xem có chứa bao nhiêu từ
16            String s_81 = sc_81.nextLine();
17            //tách nó thành mảng các xâu là các từ
18            String[] allWords_81 = s_81.split(" ");
19            //lặp qua các phần tử trong mảng
20            for (String word : allWords_81) {
21                //nếu nó không rỗng(không phải ký tự space hoặc tab) thì biến dem_tong_81 cộng thêm 1
22                if (!word.isEmpty()) {
23                    dem_tong_81++;
24                }
25            }
26        }
27    }
28 }
```

```
24        }
25    }
26    }
27    //in ra kết quả
28    System.out.println(dem_tong_81);
29    //đóng nhập dữ liệu
30    sc_81.close();
31 }
32 }
```


Test:

Input

Hanoi University Of Science and Technology
School of Information and Communication Technology



Correct output

12



User output

12



Input

As the economy enters the next credit cycle, banks are finding both opportunities and challenges in the market f



By failing to meet the needs of these businesses, banks are leaving opportunities on the table.

Correct output

72



User output

72



Input

Sign up for emails on new Risk articles



Correct output

8



User output

8



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

CS
Computer Science
CS is a very hot domain recently. Many young student wish to follow CS professions.

Correct output

Computer Science is a very hot domain recently. N

User output

Computer Science is a very hot domain recently. N

Input

AI
Artificial Intelligence
AI is a key technology.



Correct output

Artificial Intelligence is a key technology.



User output

Artificial Intelligence is a key technology.



Input

VRP
Vehicle Routing Problem
VRP is very popular research problem, started in 1959. Recently, many variants of VRP are proposed.



Correct output

Vehicle Routing Problem is very popular research



User output

Vehicle Routing Problem is very popular research

