

Viết chương trình thực hiện các công việc sau:

- Nhập các số nguyên dương N, A và B.
- Tính tổng các số trong khoảng từ 1 đến N chia hết cho A hoặc B.

Ví dụ: N = 10, A = 2, B = 3

Kết quả = 42 (2 + 3 + 4 + 6 + 8 + 9 + 10)

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 3000 byte.
- Thời gian tính toán (không tính thời gian nhập liệu) không quá 2 giây.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Tham khảo các hình dưới đây để biết quy cách vào ra dữ liệu của chương trình:

```
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\52.py =====
N = 10
A = 2
B = 3
Ket qua = 42
>>>
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\52.py =====
N = 100
A = 5
B = 3
Ket qua = 2418
```

Viết chương trình thực hiện các công việc sau:

- Nhập số nguyên dương N.
- Tính giá trị hàm F(N) theo công thức dưới đây và in ra màn hình với độ chính xác 7 số sau dấu phẩy.

$$F(N) = \frac{1}{\sqrt{1}} + \frac{2}{\sqrt{1+2}} + \dots + \frac{N}{\sqrt{1+2+\dots+N}}$$

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 3000 byte.
- Thời gian tính toán (không tính thời gian nhập liệu) không quá 2 giây.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Tham khảo các hình dưới đây để biết quy cách vào ra dữ liệu của chương trình:

```
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\24.py =====
N = 1
F(1) = 1.0000000
>>>
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\24.py =====
N = 2
F(2) = 2.1547005
>>>
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\24.py =====
N = 3
F(3) = 3.3794454
>>>
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\24.py =====
N = 100
F(100) = 138.3165108
```


Viết chương trình thực hiện các công việc sau:

- Nhập số nguyên dương N, sau đó nhập N giá trị, các giá trị này thuộc nhiều loại dữ liệu và được thêm vào nhiều danh sách khác nhau.
- Nếu giá trị kiểu nguyên hoặc kiểu thực, thêm vào danh sách A.
- Nếu không phải kiểu nguyên hoặc kiểu thực, thêm vào danh sách B.
- In ra tổng các phần tử trong A.
- In ra các phần tử thuộc danh sách B, các phần tử được nối với nhau bởi dấu gạch giữa.

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 3000 byte.
- Thời gian tính toán (không tính thời gian nhập liệu) không quá 2 giây.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Tham khảo các hình dưới đây để biết quy cách vào ra dữ liệu của chương trình:

```
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\44.py =====
Nhập N: 6
Nhập giá trị thu 1: 1
Nhập giá trị thu 2: a
Nhập giá trị thu 3: 2
Nhập giá trị thu 4: b
Nhập giá trị thu 5: 3.5
Nhập giá trị thu 6: c
Tổng các phần tử của A = 6.5
B = a-b-c
>>>
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\44.py =====
Nhập N: 3
Nhập giá trị thu 1: a
Nhập giá trị thu 2: 1232!!!
Nhập giá trị thu 3: xyz
Tổng các phần tử của A = 0
B = a-1232!!!-xyz
```


Viết chương trình thực hiện các công việc sau:

- Nhập một chuỗi S từ bàn phím.
- Nếu S chứa một số lượng lẻ các dấu chấm than, hãy thêm một dấu chấm than vào cuối S. In ra chuỗi S sau khi được xử lý.

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 3000 byte.
- Thời gian tính toán (không tính thời gian nhập liệu) không quá 2 giây.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Tham khảo các hình dưới đây để biết quy cách vào ra dữ liệu của chương trình:

```
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\34.py =====
Nhập S: !
Chuỗi S sau khi xử lý: !!
>>>
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\34.py =====
Nhập S: abcd!!
Chuỗi S sau khi xử lý: abcd!!
>>>
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\34.py =====
Nhập S: 123..456
Chuỗi S sau khi xử lý: 123..456
>>>
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\34.py =====
Nhập S: 123!..456
Chuỗi S sau khi xử lý: 123!..456!
```

Tính và in ra màn hình giá trị sin của các góc 0 độ, 10 độ, 20 độ,..., 100 độ; mỗi giá trị được ghi trên một dòng.

Xem ví dụ dưới đây để biết quy cách nhập/xuất dữ liệu của chương trình.

```
sin(0) = 0.0  
sin(10) = 0.17364817766693033  
sin(20) = 0.3420201433256687  
sin(30) = 0.49999999999999994  
sin(40) = 0.6427876096865393  
sin(50) = 0.766044443118978  
sin(60) = 0.8660254037844386  
sin(70) = 0.9396926207859083  
sin(80) = 0.984807753012208  
sin(90) = 1.0  
sin(100) = 0.984807753012208
```

Viết chương trình thực hiện các công việc sau:

- Nhập các số nguyên dương N, A và B.
- Tính tổng các số trong khoảng từ 1 đến N chia hết cho A hoặc B.

Ví dụ: N = 10, A = 2, B = 3

Kết quả = 42 (2 + 3 + 4 + 6 + 8 + 9 + 10)

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 3000 byte.
- Thời gian tính toán (không tính thời gian nhập liệu) không quá 2 giây.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Tham khảo các hình dưới đây để biết quy cách vào ra dữ liệu của chương trình:

```
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\S2.py =====
N = 10
A = 2
B = 3
Ket qua = 42
>>>
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\S2.py =====
N = 100
A = 5
B = 3
Ket qua = 2418
```

Viết chương trình thực hiện các công việc sau:

- Nhập số nguyên dương N .
- Tính giá trị hàm $F(N)$ theo công thức dưới đây và in ra màn hình với độ chính xác 6 số sau dấu phẩy.

$$F(N) = \frac{1}{1} + \frac{2}{1+2} + \dots + \frac{N}{1+2+\dots+N}$$

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 9999 byte.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Dưới đây là minh họa quy cách vào ra dữ liệu và trình bày màn hình:

```
===== RESTART: C:/TEMP/2.py =====
N = 1
F(1) = 1.000000
>>>
===== RESTART: C:/TEMP/2.py =====
N = 100
F(100) = 8.394557
>>>
===== RESTART: C:/TEMP/2.py =====
N = 10000
F(10000) = 17.575412
>>>
===== RESTART: C:/TEMP/2.py =====
N = 1000000
F(1000000) = 26.785455
```

Một chuỗi nhị phân S được gọi là không-lặp-bậc- k nếu không có hai chuỗi con liên tiếp độ dài k của chúng trùng nhau.

Hãy nhập và kiểm tra xem một chuỗi có phải không lặp bậc k ?

Ví dụ: $s = 1010, k = 3 \Rightarrow$ đúng
 $s = 1010, k = 2 \Rightarrow$ sai

Giải thích: $k = 2$, dãy s chứa 2 dãy con 10 lặp lại liên tiếp
 $k = 3$, dãy s không chứa 2 dãy con liên tiếp độ dài 3 giống nhau

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 9999 byte.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Dưới đây là minh họa quy cách vào ra dữ liệu và trình bày màn hình:

```
***** RESTART: C:\TEMP2\DAYKHONGLAP.py *****
S = 1010
k = 2
Day lap bac 2
>>>
***** RESTART: C:\TEMP2\DAYKHONGLAP.py *****
S = 10101
k = 3
Day khong lap
```


Viết chương trình thực hiện các công việc sau:

- Nhập số nguyên dương N.
- Đếm xem có bao nhiêu cách phân tích số N thành tổng của đúng 4 số nguyên dương chẵn.
- In ra màn hình số cách đếm được.

Giải thích:

Với $N = 10$, ta có 4 cách phân tích như sau:

$$10 = 2 + 2 + 2 + 4$$

$$10 = 2 + 2 + 4 + 2$$

$$10 = 2 + 4 + 2 + 2$$

$$10 = 4 + 2 + 2 + 2$$

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 3000 byte.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Tham khảo các hình dưới đây để biết quy cách vào ra dữ liệu của chương trình:

```
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\54.py =====
N = 5
Có 0 cách phân tích 5 thành tổng 4 số nguyên dương chẵn
>>>
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\54.py =====
N = 10
Có 4 cách phân tích 10 thành tổng 4 số nguyên dương chẵn
>>>
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\54.py =====
N = 16
Có 35 cách phân tích 16 thành tổng 4 số nguyên dương chẵn
>>>
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\54.py =====
N = 1000
Có 20584249 cách phân tích 1000 thành tổng 4 số nguyên dương chẵn
```


Nhập số đo góc A (đơn vị là độ, số thực), sau đó hãy tính $\sin(A)$ và in ra màn hình.

Xem ví dụ dưới đây để biết quy cách nhập/xuất dữ liệu của chương trình.

```
Nhap goc A = 90  
sin(90.0) = 1.0
```

Viết chương trình thực hiện các công việc sau:

- Nhập số nguyên dương N.
- Tính giá trị hàm $F(N)$ theo công thức dưới đây và in ra màn hình với độ chính xác 6 số sau dấu phẩy.

$$F(N) = \frac{1}{1} + \frac{2}{1+2} + \dots + \frac{N}{1+2+\dots+N}$$

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 9999 byte.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Dưới đây là minh họa quy cách vào ra dữ liệu và trình bày màn hình:

```
===== RESTART: C:/TEMP/2.py =====
N = 1
F(1) = 1.000000
>>>

===== RESTART: C:/TEMP/2.py =====
N = 100
F(100) = 8.394557
>>>

===== RESTART: C:/TEMP/2.py =====
N = 10000
F(10000) = 17.575412
>>>

===== RESTART: C:/TEMP/2.py =====
N = 1000000
F(1000000) = 26.785455
```


Viết chương trình thực hiện các công việc sau:

- Nhập 3 số nguyên a, b và c.
- Nếu các số a b c theo một thứ tự nào đó có thể tạo thành cấp số cộng thì in ra màn hình “Nhiệm vụ hoàn thành”, ngược lại in ra màn hình “Nhiệm vụ thất bại”.

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 3000 byte.
- Thời gian tính toán (không tính thời gian nhập liệu) không quá 2 giây.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Tham khảo các hình dưới đây để biết quy cách vào ra dữ liệu của chương trình:

```
===== RESTART: C:/TEMP/1.py =====
A = 12
B = 13
C = 14
Nhiệm vụ hoàn thành
>>>
===== RESTART: C:/TEMP/1.py =====
A = 3
B = 1
C = 5
Nhiệm vụ hoàn thành
>>>
===== RESTART: C:/TEMP/1.py =====
A = 0
B = 6
C = 11
Nhiệm vụ thất bại
```

$k = 3$, dãy s không chứa 2 dãy con liên tiếp độ dài 3 giống nhau

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 9999 byte.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Dưới đây là minh họa quy cách vào ra dữ liệu và trình bày màn hình:

```
===== RESTART: C:\TEMP2\DAYKHONGLAP.py =====
S = 1010
k = 2
Day lap bac 2
>>>
===== RESTART: C:\TEMP2\DAYKHONGLAP.py =====
S = 10101
k = 3
Day khong lap
```


Viết chương trình thực hiện các công việc sau:

- Nhập số nguyên dương N.
- Đếm xem có bao nhiêu cách phân tích số N thành tổng của đúng 4 số nguyên dương.
- In ra màn hình số cách đếm được.

Giải thích:

Với $N = 5$, ta có 4 cách phân tích như sau:

$$5 = 1 + 1 + 1 + 2$$

$$5 = 1 + 1 + 2 + 1$$

$$5 = 1 + 2 + 1 + 1$$

$$5 = 2 + 1 + 1 + 1$$

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 9999 byte.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Dưới đây là minh họa quy cách vào ra dữ liệu và trình bày màn hình:

```
===== RESTART: C:\TEMP\5.py =====
N = 3
Có 0 cách phân tích 3 thành tổng 4 số nguyên dương
>>>
===== RESTART: C:\TEMP\5.py =====
N = 4
Có 1 cách phân tích 4 thành tổng 4 số nguyên dương
>>>
===== RESTART: C:\TEMP\5.py =====
N = 5
Có 4 cách phân tích 5 thành tổng 4 số nguyên dương
>>>
===== RESTART: C:\TEMP\5.py =====
N = 6
Có 10 cách phân tích 6 thành tổng 4 số nguyên dương
>>>
===== RESTART: C:\TEMP\5.py =====
N = 60
Có 32509 cách phân tích 60 thành tổng 4 số nguyên dương
>>>
===== RESTART: C:\TEMP\5.py =====
N = 600
Có 35641099 cách phân tích 600 thành tổng 4 số nguyên dương
```


Nhập một dãy số nguyên từ bàn phím, các số được viết trên cùng một dòng, ngăn cách nhau bởi dấu phẩy; Hãy đảo ngược dãy và in ra màn hình.

Xem ví dụ dưới đây để biết quy cách nhập/xuất dữ liệu của chương trình.

Day so: 10,11,12
[12, 11, 10]

Day so: 9,8,7,6,5,5,5,5,0
[0, 5, 5, 5, 5, 6, 7, 8, 9]

Viết chương trình thực hiện các công việc sau:

- Nhập các số nguyên dương N, A và B.
- Tính tổng các số trong khoảng từ 1 đến N chia hết cho A hoặc B nhưng không đồng thời chia hết cho cả 2 số A và B.

Ví dụ: N = 10, A = 2, B = 3

Kết quả = 36 (2 + 3 + 4 + 8 + 9 + 10)

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 3000 byte.
- Thời gian tính toán (không tính thời gian nhập liệu) không quá 2 giây.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Tham khảo các hình dưới đây để biết quy cách vào ra dữ liệu của chương trình:

```
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\S3.py =====
N = 10
A = 2
B = 3
Ket qua = 36
>>>
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\S3.py =====
N = 120
A = 5
B = 3
Ket qua = 2880
>>>
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\S3.py =====
N = 4500
A = 7
B = 11
Ket qua = 2103622
```

Viết chương trình thực hiện các công việc sau:

- Nhập 3 số nguyên a, b và c.
- Nếu các số a b c là ba số nguyên phân biệt (tức là không có hai số có giá trị bằng nhau) thì in ra màn hình "BA SO PHAN BIET", ngược lại in ra màn hình "BA SO KHONG PHAN BIET".

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 3000 byte.
- Thời gian tính toán (không tính thời gian nhập liệu) không quá 2 giây.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Tham khảo các hình dưới đây để biết quy cách vào ra dữ liệu của chương trình:

```
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\14.py =====
A = 1
B = 2
C = 2
BA SO KHONG PHAN BIET
>>>
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\14.py =====
A = 14
B = -9
C = 70
BA SO PHAN BIET
>>>
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\14.py =====
A = 10
B = 10
C = 10
BA SO KHONG PHAN BIET
```


Viết chương trình thực hiện các công việc sau:

- Nhập số nguyên dương N, sau đó nhập N giá trị, các giá trị này thuộc nhiều loại dữ liệu và được thêm vào nhiều danh sách khác nhau.
- Nếu giá trị kiểu nguyên hoặc kiểu thực, thêm vào danh sách A.
- Nếu không phải kiểu nguyên hoặc kiểu thực, thêm vào danh sách B.
- In ra tổng các phần tử trong A.
- In ra các phần tử thuộc danh sách B, các phần tử được nối với nhau bởi dấu gạch giữa.

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 3000 byte.
- Thời gian tính toán (không tính thời gian nhập liệu) không quá 2 giây.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Tham khảo các hình dưới đây để biết quy cách vào ra dữ liệu của chương trình:

```
***** RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\44.py *****
Nhập N: 6
Nhập giá trị thu 1: 1
Nhập giá trị thu 2: a
Nhập giá trị thu 3: 2
Nhập giá trị thu 4: b
Nhập giá trị thu 5: 3.5
Nhập giá trị thu 6: c
Tổng các phần tử của A = 6.5
B = a.b.c
>>>
***** RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\44.py *****
Nhập N: 3
Nhập giá trị thu 1: a
Nhập giá trị thu 2: 1232!!!
Nhập giá trị thu 3: xyz
Tổng các phần tử của A = 0
B = a-1232!!!-xyz
```

Cho số nguyên dương N chẵn, hãy tính $P(N)$ là tổng bình phương các số chẵn:

$$P(N) = 2^2 + 4^2 + \dots (N-2)^2 + N^2$$

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 9999 byte.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Dưới đây là minh họa quy cách vào ra dữ liệu và trình bày màn hình:

```
===== RESTART: C:\TEMP2\TONGBINHPHUONG2.py =====  
N = 4  
P(4) = 20  
>>>  
===== RESTART: C:\TEMP2\TONGBINHPHUONG2.py =====  
N = 40  
P(40) = 11480  
>>>
```

Viết chương trình thực hiện các yêu cầu sau: Nhập số nguyên dương n và tính tổng

$$S = 1 - 1.2 + 1.2.3 - \dots + (-1)^{n-1} 1.2 \dots n$$

Chú ý: Xem ví dụ dưới đây để biết quy cách nhập/xuất dữ liệu của chương trình. Chú ý không xuất thừa dấu cách ở cuối dòng và kiểm soát ngoại lệ khi không nhập được số nguyên.

```
Nhap n = -1.1
Nhap sai! Dung chuong trinh!
```

```
Nhap n = -1
Nhap sai! Dung chuong trinh!
```

```
Nhap n = 3
Tong S = 5
```

```
Nhap n = 2
Tong S = -1
```

```
Nhap n = 20
Tong S = -2317323290554617019
```


Viết chương trình thực hiện các công việc sau:

- Nhập số nguyên dương N, sau đó nhập N giá trị, các giá trị này thuộc nhiều loại dữ liệu và được thêm vào nhiều danh sách khác nhau.
- Nếu giá trị kiểu nguyên hoặc kiểu thực, thêm vào danh sách A.
- Nếu không phải kiểu nguyên hoặc kiểu thực, thêm vào danh sách B.
- In ra màn hình tổng các phần tử của A, sau đó in ra danh sách B với các phần tử được sắp xếp giảm dần.

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 3000 byte.
- Thời gian tính toán (không tính thời gian nhập liệu) không quá 2 giây.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Tham khảo các hình dưới đây để biết quy cách vào ra dữ liệu của chương trình:

```
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\42.py =====  
Nhập N: 6  
Nhập giá trị thu 1: 1.2  
Nhập giá trị thu 2: 7  
Nhập giá trị thu 3: abc  
Nhập giá trị thu 4: -6  
Nhập giá trị thu 5: !!!  
Nhập giá trị thu 6: 12
```

Cho số nguyên dương N lẻ, hãy tính $P(N)$ là tổng bình phương các số lẻ theo công thức dưới đây:

$$P(N) = 1^2 + 3^2 + \dots (N-2)^2 + N^2$$

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 9999 byte.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Dưới đây là minh họa quy cách vào ra dữ liệu và trình bày màn hình:

```
----- RESTART: C:\TEMP2\TONGBINHPHUONG3.py -----
N = 5
P(5) = 35
>>>
----- RESTART: C:\TEMP2\TONGBINHPHUONG3.py -----
N = 55
P(55) = 29260
```

Viết chương trình thực hiện các công việc sau:

- Nhập một chuỗi S từ bàn phím.
- Nếu S không chứa dấu chấm than nào, hãy thêm hai dấu chấm than vào cuối S; Ngược lại, nếu S chứa số dấu chấm than lẻ, hãy thêm một dấu chấm than vào cuối S. In ra chuỗi S sau khi được xử lý.

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 3000 byte.
- Thời gian tính toán (không tính thời gian nhập liệu) không quá 2 giây.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Tham khảo các hình dưới đây để biết quy cách vào ra dữ liệu của chương trình:

```
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\32.py =====  
Nhập S: 123  
Chuỗi S sau khi xử lý: 123!!  
>>>  
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\32.py =====  
Nhập S: 1!2!3!  
Chuỗi S sau khi xử lý: 1!2!3!!  
>>>  
===== RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\32.py =====  
Nhập S: 1!!23  
Chuỗi S sau khi xử lý: 1!!23
```

Cho 4 đoạn thẳng lần lượt có độ dài A, B, C và D. Chọn 3 trong 4 đoạn thẳng nếu may mắn ta có thể tạo thành một tam giác với 3 cạnh độ dài tương ứng với 3 đoạn vừa chọn (không kể tam giác suy biến), tất nhiên có những bộ số mà ta không tạo được tam giác nào.

Hãy nhập 4 độ dài và tính tổng diện tích các tam giác có thể tạo được.

Chú ý:

- Nếu không có phương án tạo tam giác nào, hãy in ra kết quả là -1.
- Nếu tồn tại tam giác, hãy in ra tổng diện tích tất cả các tam giác với độ chính xác 5 chữ số thập phân.

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 9999 byte.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Dưới đây là minh họa quy cách vào ra dữ liệu và trình bày màn hình:

```
===== RESTART: C:\TEMP2\TAMGIACSUM.py =====
Do dai A = 1
Do dai B = 2
Do dai C = 3
Do dai D = 4
Ket qua = 2.90474
>>>
===== RESTART: C:\TEMP2\TAMGIACSUM.py =====
Do dai A = -1
Do dai B = 2
Do dai C = 3
Do dai D = 0
Ket qua = -1
```


Một chuỗi nhị phân S được gọi là không-lặp-bậc-k nếu không có hai chuỗi con liên tiếp độ dài k của chúng trùng nhau.

Hãy nhập và kiểm tra xem một chuỗi có phải không lặp bậc k?

Ví dụ: s = 1010, k = 3 => đúng
 s = 1010, k = 2 => sai

Giải thích: k = 2, dãy s chứa 2 dãy con 10 lặp lại liên tiếp
 k = 3, dãy s không chứa 2 dãy con liên tiếp độ dài 3 giống nhau

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 9999 byte.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Dưới đây là minh họa quy cách vào ra dữ liệu và trình bày màn hình:

```
===== RESTART: C:\TEMP2\DAYKHONGLAP.py =====
S = 1010
k = 2
Day lap bac 2
>>>
===== RESTART: C:\TEMP2\DAYKHONGLAP.py =====
S = 10101
k = 3
Day khong lap
```

Viết chương trình thực hiện các công việc sau:

- Nhập số nguyên dương N, sau đó nhập N giá trị, các giá trị này thuộc nhiều loại dữ liệu và được thêm vào nhiều danh sách khác nhau.
- Nếu giá trị kiểu nguyên hoặc kiểu thực, thêm vào danh sách A.
- Nếu không phải kiểu nguyên hoặc kiểu thực, thêm vào danh sách B.
- In ra màn hình tổng các phần tử của A, sau đó in ra danh sách B với các phần tử được sắp xếp giảm dần.

Yêu cầu cho mọi bài giải:

- Mã nguồn viết bằng python không quá 3000 byte.
- Thời gian tính toán (không tính thời gian nhập liệu) không quá 2 giây.
- Để được điểm tối đa, dữ liệu nhập xuất cần viết chính xác theo nội dung yêu cầu, bao gồm cả chữ hoa, chữ thường, dấu cách và các kí hiệu khác.

Tham khảo các hình dưới đây để biết quy cách vào ra dữ liệu của chương trình:

```
***** RESTART: C:\Nam.Test\FineApps2\FinePython\bin\Debug\data\42.py *****
Nhập N: 6
Nhập giá trị thu 1: 1.2
Nhập giá trị thu 2: 7
Nhập giá trị thu 3: abc
Nhập giá trị thu 4: -6
Nhập giá trị thu 5: lll
Nhập giá trị thu 6: 12
Tổng các phần tử của A = 14.2
B = ['abc', 'lll']
```