Faculty of Electrical, Electronic & Computer Technology	SAXONY EGYPT UNIVERSITY FOR APPLIED SCIENCE AND TECHNOLOGY
Computer Programming	
ECT 121	
Dr. Amina Elhawary	
Eng. Heba Mohsen	

Sheet 4

1) Problems :

1. Given a number *N*. Print all **even** numbers between **1** and ***N*** inclusive in separate lines.

Input

Only one line containing a number *N*.

Output

Print the answer according to the required above. If there are no **even** numbers print **-1**.

Examples

Input
10
Output
2 4 6 8 10

2. Given N numbers. **Count** how many of these values are **even, odd, positive** and **negative**.

Input

First line contains one number *N* number of values.

Second line contains *N* numbers .

Output

Print four lines with the following format:

First Line: "Even: *X*", where *X* is the number of **even** numbers in the given input.

Second Line: "Odd: *X*", where *X* is the number of **odd** numbers in the given input.

Third Line: "Positive: *X*", where *X* is the number of **positive** numbers in the given input.

Fourth Line: "Negative: *X*", where *X* is the number of **negative** numbers in the given input.

Examples

Input
5 -5 0 -3 -4 12
Output
Even: 3 Odd: 2 Positive: 1 Negative: 3

3. Given multiple lines each line contains a number *X* which is a password. Print "**Wrong**" if the password is **incorrect** otherwise, print "**Correct**" and **terminate** the program.

   Note: The "**Correct**" password is the number **1999**.

   Input

   The input contains several passwords.

   Each line contains a number *X* .

   Output

   Print "**Wrong**" if the password is typed **wrong** otherwise, print "**Correct**" if the password is typed **correctly**.

Input
2200 1020 1999 1000 9999

Output
Wrong Wrong Correct

4. Given a number *N*, and *N* numbers, find **maximum** number in these *N* numbers.

   Input

   First line contains a number *N* .

   Second line contains *N* numbers *X* .

   Output

   Print the **maximum** number.

   Examples :

Input
5 1 8 5 7 5
Output
8

5. Given a number *N*. Print the **maltiplication table** of the number from **1** to **12**
   Input
   Only one line containing a number *N* .
   Output
   Print **12** lines according to the required above.

   Examples :

Input
1
Output
1 * 1 = 1 1 * 2 = 2 1 * 3 = 3

```
1 * 4 = 4
1 * 5 = 5
1 * 6 = 6
1 * 7 = 7
1 * 8 = 8
1 * 9 = 9
1 * 10 = 10
1 * 11 = 11
1 * 12 = 12
```

6. Given a number N. Print **2** lines that contain the following respectively:
   1. Print N in a reversed order and **not leading zeroes**.
   2. If N is a **palindrome number** print "**YES**" otherwise, print "**NO**.

Note:

A **palindrome number** is a number that reads the same forward or backward.

For example: 12321, **101** are **palindrome numbers**, while **1201**, **221** are **not**.

A leading zero is any 0 digit that comes before the first nonzero digit in a number for example : numbers **(005 , 01 , 0123 , 02 , 000250)** are leading zeroes but **(5 , 123 , 20 ,2500)** not leading zeroes numbers .

Input

Only one line containing a number N.

Output

Print the answer required above.

Examples :

Input
12121
Output
12121 YES

Input
160
Output
61 NO

7. Given a number *N*. Print all the **divisors** of *N* in ascending order.
   Input
   Only one line containing a number *N* .
   Output
   Print all **positive divisors** of *N*, one number per line.

   Examples :

Input
6
Output
1 2 3 6

Input
7
Output
1 7

Input
4
Output
1 2 4

8. Given two numbers *A* and *B*. Print the **greatest common divisor** between (*A*, *B*).
   Note: The greatest common divisor (**GCD**) of two or more integers, which are not all zeroes, **is the largest positive integer that divides each of the integers**.
   For example:
   the **GCD** of **8** and **12** is **4**.
   because the numbers that divides both **8** and **12** are **(1,2,4)** and **4** is the largest one .
   Input
   Only one line containing two numbers *A* and *B* .
   Output
   Print the **GCD** of *A* and *B*.

   Examples :

Input
12 8
Output
4

Input
3 7
Output
1

9. Given two numbers A and B. Print all **lucky numbers** between A and B **inclusive**.

   Note:

   The **Lucky number** is any positive number that its decimal representation contains only **4** and **7**.

   For example: numbers **4, 7, 47** and **744** are **lucky** and numbers **5, 17** and **174** are **not**.

   Input

   Only one line containing two numbers A and B.

   Output

   Print all **lucky numbers** between A and B **inclusive** separated by a space. If there is **no lucky number** print **-1**.

   Examples :

Input
4 20
Output
4 7

Input
8 15
Output
-1

10. Given **3** lines of input described as follow:
    1. First line contains a symbol SS.
    2. Second line contains a number NN.
    3. Third line contains NN numbers.

    For each number XiXi in the NN numbers print a new line that contains the symbol SS repeated XiXi time.

    Input

The first line contains a symbol S can be (+,−,∗,/).
The second line an number N.
The third line contains N numbers .
Output
Print the answer required above.

Examples :

Input
+ 5 5 2 4 3 7
Output
+++++ ++ ++++ +++ +++++++

11. Given a number *N*. Print the **digits of that number** from right to left separated by space.

    Input
    First line contains a number *T* (1 ≤ *T* ≤ 10) number of test cases.
    Next *T* lines will contain a number *N* (0 ≤ *N* ≤ 10^9)
    Output
    For each test case print a single line contains the **digits of the number** separated by space.

Examples :

Input
4 121 39 123456 1200
Output
1 2 1 9 3 6 5 4 3 2 1 0 0 2 1

12. Given multiple lines each line contains two numbers *N* and *M*.

For each line print a single line contains:

- The numbers between *N* and *M* inclusive separated by single space.
- The message " sum =".
- The **summation** of all numbers between *N* and *M* inclusive.

Note: The program should be *TERMINATED* as soon as any of these two numbers is less than or equal to zero and don't print any thing.

For more clarification see the examples below.

Input

The input contains multiple line.

Each line contains two numbers *N* and *M* .

It's **guaranteed** that the last line of the input will contain a number that is less than or equal to zero.

Output

For each line print the answer according to the required above in a single line.

Input
5 2 5 7 5 -1
Output
2 3 4 5 sum =14 5 6 7 sum =18

Input
5 2 6 3 5 0
Output
2 3 4 5 sum =14 3 4 5 6 sum =18

13. Given a number *N*. Print a pyramid that has *N* rows.

For more clarification see the example below.

Input

Only one line containing a number *N* (1 ≤ *N* ≤ 99).

Output

Print the answer according to the required above.

Input
4
Output
<pre>   *   *** ***** *******</pre>

14. Given a number *N*. Print a diamond that has 2*N* rows.

For more clarification see the example below.

Input

Only one line containing number *N* (1 ≤ *N* ≤ 99).

Output

Print the answer according to the required above.

Input
4
Output
<pre>   *   *** ***** ******* ******* *****   ***   *</pre>

2) Tracing Questions

1. How many turns does this loop make?

```
#include<iostream>
using namespace std;
int main() {
    unsigned char n = 150;
    for (unsigned char i = 0; i < 2 * n; ++i) {
    }
}
```

2. How many turns does this loop make?

```
#include <iostream>
using namespace std;
int main() {
   for (;;)
        cout << "blank";
   return 0;
}
```

3. What is the error here ?

```
#include<iostream>
int main() {
  int i;
  for (i = 0; i < 100; i++);
    std::cout << i << std::endl;
}
```

4. What is The Output of the following Code?

```
#include<iostream>
using namespace std;
int main() {
   int counter = 0;
   for (int i = 0; i < 5; i++) {
        for (int j = i; j < 5; j++) {
             for (int k = j; k < 5; k++) {
                  counter++;              }    }    }
   cout << counter;
}
```

5. What is The Output of the following Code?

```
#include <iostream>
using namespace std;
int main() {
     int i = 0, x = 0;
     do {
          if (i % 5 == 0) {
               cout << x;
               x++;
          }
          i++;
     } while (i < 10);
     cout << x;
     return 0;
}
```