



# Computer Programming Olympiad

A project of the Institute of IT Professionals South Africa

Ph: 021-448 7864 • Fax: 021-447 8410 • PO Box 13013, MOWBRAY, 7705 • info@olympiad.org.za • www.olympiad.org.za

---

## Programming Olympiad 2016 : Round 2

Online only on 7 September 2016

### 1. Palindromic Sum

Adding any natural number to its reverse and repeating the operation with the resulting sum will, in nearly all cases, eventually give a palindromic sum. (A palindromic number is a number which reads the same from right to left or from left to right. 323 is palindromic while 321 is not).

For example:

$38 + 83 = 121$  (a palindromic number in one step)

$69 + 96 = 165$

$165 + 561 = 726$

$726 + 627 = 1353$

$1353 + 3531 = 4884$  (a palindromic number in 4 steps)

Write a program that asks for a natural number and then calculates how many steps it will take to convert that number into a palindromic number as described above.

#### Examples:

Input: 38

Output: 1

Input: 69

Output: 4

**Test your program with the following and type or paste the answers in the correct block on the web page:**

a) 42

b) 527

c) 739

d) 89

[For the mathematicians: It is not known if this procedure will ever terminate for the number 196.]

## 2.Ad Infinitum

When converting ordinary fractions into decimal fractions one sometimes ends up with a set of decimal digits that keep on repeating (so-called recurring decimals). For example  $\frac{1}{3} = 0.3333333$  etc.

Write a program that will convert an ordinary fraction into a decimal fraction and that places the recurring digits in brackets.

For example:

$\frac{22}{5} = 4.4$  (no recurring decimal digits so no brackets)

$\frac{1}{3} = 0.(3)$  (three recurs ad infinitum)

### **Examples:**

Input: Fraction?  $\frac{1}{3}$

Output: 0.(3)

Input: Fraction?  $\frac{9}{7}$

Output: 1.(285714)

**Test your program with the following and type or paste the answers in the correct block on the web page:**

a:  $\frac{3}{8}$

b:  $\frac{3}{7}$

c:  $\frac{99}{43}$

d:  $\frac{45}{46}$

## 3.Word Chain

A *word chain* is an ordering of a set of words such that each word is exactly one letter different from the previous word.

For example, in the set of words {CARE HARE CART}, CARE and CART only differ in the last letter, and HARE and CARE differ in the first. Thus a valid word chain for this set of words might be

HARE CARE CART

Write a program that, given a set of such words, outputs a valid word chain.

Output the word chain as a sequence of words (in the correct order) separated by one space.

If there is no correct answer, output the word "impossible".

For test case (a), if there are multiple valid chains, you may output any one of them.

For test cases (b), (c) and (d), if there are multiple valid chains, output the chain (string) that comes first alphabetically.

## Examples:

Input: DOG CAT COW

Output: Impossible

Input: CARE HARE CART

Output: CART CARE HARE

[Note: HARE CARE CART would also be a valid word chain, but does not come first alphabetically.]

Input: PAT BUT PET BUF POT RAT PUT

Output: BUF BUT PUT PET POT PAT RAT

[ Note : BUF BUT PUT POT PET PAT RAT would also be a valid word chain, but does not come first alphabetically; The two strings are identical until the 11<sup>th</sup> letter; then the one has an E and the other an O. E comes before O alphabetically.

**Test your program with the following and type or paste the answers in the correct block on the web page:**

a) POP MAP MOP

b) BOB HOT FOB COT FOR LOT NOT NOR

c) CUT HOP COT CAR BAT HUT CAT HAT BAR

d) WALL TAIL TALL WALK BALL WAIL TALK