Computer Olympiad
South African Computer Olympiad: a project of the Computer Society of 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 2013: Round 1

Not to be used before 26 July 2013

- This paper is for ALL candidates.
- Each correct answer earns 5 marks.
- You have 2 hours to attempt as many questions as possible.
- Programs that produce 3 correct answers can earn additional marks for readability, conciseness, and for appropriate comments and variable names.
- Indicate the question, your name, surname and the language and version used at the start of every program e.g. “Q3 Sam King, Python 2.7”
- Save your program as Qn Name Surname e.g. Q3 Sam King

DO NOT MODIFY ANY FILES AFTER THE END OF THE CONTEST AS THIS WILL DISQUALIFY YOU

---

1. ALPHABET SQUARE

[Adapted from the ICPSC]

Assign each letter of the alphabet (upper or lower case) a number according to its ranking: A = a = 1, B = b = 2, C = c = 3, D = d = 4, E = e = 5, etc. Given a single letter as input, print a square consisting of the given letter with side length equal to the ranking of the letter, as in the examples.

Examples:
Input: `D`
Output:

```
D D D D
D D D D
D D D D
D D D D
```

Input: `e`
Output:

```
e e e e e
e e e e e
e e e e e
e e e e e
e e e e e
```

Use the following values to test your program
a: `C`
b: `g`
c: `T`

2. SIMPLE CODING

[by Phoenix Rhymer of Durban Road High]

A, E, I, O and U are frequently used letters in English. J, K, Q, X and Z are the least used. If you replaced A, E, I, O and U respectively with J, K, Q, X, Z it would make your “code” difficult to read, but if you change it back, would you still be able to read it?
Write a program that will take a sentence, change all A, E, I, O, U to J, K, Q, X, Z respectively, print it out, then change all J, K, Q, X and Z to A, E, I, O, U and print it out again. (Upper- and lowercase remain upper- and lowercase, punctuation and spaces remain the same)

Example:
Input: `An aeroplane! Look in the sky`
Output:

```
Jn jkrxpljnk! Lxxk qn thk sky
An aeroplane! Looe in the sey
```

Use the following sentences to test your program
a: `The quick brown fox jumped over the lazy dog.`
b: `Orange rats, brown rats, grey rats, tawny rats.`
c: `Look at my book. Is it the junior version?`

Supported by Mark Shuttleworth, the University of Cape Town and the Centre for High Performance Computing.

Sponsored by Standard Bank

3. WELL-ORDERED NUMBERS

[Adapted from the ICPSC]

A number is well-ordered when its digits are in a numerically ascending order. E.g. 147 is well-ordered but 174 is not. In a well-ordered number, each of the digits 1 - 9 may only be used once.
Write a program that will calculate the sum of all the well-ordered numbers that are possible with a fixed number of digits.

Example:
Input: `2`
Output: `1440`

[Explanation: The program added up all the well-ordered two-digit numbers.
12 + 13 + 14 + 15 + 16 + 17 + 18 + 19 +
23 + 24 + 25 + 26 + 27 + 28 + 29 +
34 + 35 + 36 + 37 + 38 + 39 +
45 + 46 + 47 + 48 + 49 +
56 + 57 + 58 + 59 +
67 + 68 + 69 +
78 + 79 +
89]

Use the following values to test your program
a: `6`
b: `5`
c: `3`

4. ANAGRAM

[by Graham van Rensburg of Westville Boys' High]

An anagram of a word contains the same letters as the original word, but in a different order.

Write a program that will calculate how many anagrams can be made with a given word – not counting the word itself. Anagrams that have the same letter-order may only be counted once. The words given will always be lower case.

Example 1:
Input: `cat`
Output: `5`
[Explanation: the accepted anagrams for "cat" are: cta, act, atc, tac, tca.
A total of 5]

Example 2:
Input: `see`
Output: `2`
[Explanation: the accepted anagrams for see are: ese, ees. Although "ees" can be made up in two ways by rearranging the letters of "see" , it only counts as a single anagram, because the final word is the same.
A total of 2]

Use the following words to test your program
a: `hound`
b: `foxhound`
c: `bookkeeper`

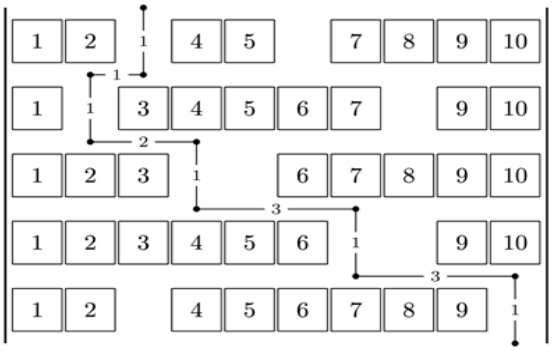
The final (external) evaluators may use additional words to test your program. These words will not have more than 15 letters.

5. SHORTEST PATH THROUGH THE HALL

[by Alan Smithee of Hulsbos High]

The hall is packed wall to wall with rows of chairs, but in each row there are exactly two chairs missing. The chairs in each row have numbers from 1 to 100. Write a program that will calculate the length of the shortest path from the front to the back of the hall.

Each chair is 1 unit wide and each row is 1 unit deep (from the front of a chair to the front of the chair behind it). It is not possible to move diagonally. You may start in front of any gap in the front row and end behind any gap in the last row. You always walk through the middle of a gap. Illustrated is the shortest path through a hall, with five rows of chairs. In the illustration the hall is only 10 chairs wide instead of 100.



The first number in the input will contain the number n – the number of rows. The next n lines will have two numbers, separated by a space, indicating where the gaps are.

Example
Input:
```
5
3 6
2 8
4 5
7 8
3 10
```

Output: `14`

Use the following sets of data to test your program

a:
```
4
5 8
9 3
1 5
2 7
```

b:
```
6
14 84
15 88
95 96
17 42
15 62
10 100
```

c:
```
10
2 42
67 12
50 51
98 5
14 92
11 17
28 79
13 37
100 1
86 83
```

The final (external) evaluators may use additional cases to test your program.
These cases may have up 100 rows.