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 2018 : Round 1

Not to be used before 13 August 2018

1. This paper is for ALL participants.
2. All answers must be **TYPED** or **PASTED** on your Answer Sheet. Handwritten Answer Sheets will be disqualified.
3. Each correct answer for questions 1 and 2 earns 8 marks while a correct answer for question 3 earns 9 marks.
4. You have 60 minutes to attempt as many questions as possible.
5. Programs should be readable, concise, and use appropriate variable names.
6. Indicate the question, your name, surname and the language and version used in a comment at the start of every program, e.g. "***Q3 Sam King, Python 2.7***"
7. Save your program as Qn Name Surname, e.g. ***Q3 Sam King***
8. You may assume that the user input will satisfy the problem specification and so you do not need to validate the input.
9. Do not write code to produce only specific answers, as the external judges may use other test cases.
10. After the contest you may be given time to print out your Answer Sheet. Do not make any changes during this time.
11. Make sure you save the programs you have created and the Answer Sheet in a place where your teacher can find them.
12. **DO NOT MODIFY ANY FILES AFTER THE END OF THE CONTEST AS THIS WILL LEAD TO YOUR DISQUALIFICATION**.

Question 1: Cellphone

A cellphone service provider has offered a new subscription plan. The specific pricing for daytime, evening and weekend usage is given below:

Daytime	**Evening**	**Weekends**
100 free minutes then 80c per minute	70c per minute	50c per minute

Your task is to write a program to determine how much the new subscription plan will cost, given how many minutes are used for daytime, evening and weekends. These are given as three non-negative integers in the order of **daytime**, **evening**, then **weekend**. Your program should output the total price in **cents** the subscription plan costs, as shown in the examples given below.

Examples:

Input:	45 0 1	Output:	50c
Input:	162 61 66	Output:	12530c
Input:	82 89 15	Output:	6980c

In the first example given, 45 **daytime** minutes, 0 **evening** minutes and 1 **weekend** minute were used. As the first 100 daytime minutes are free, no cost is allocated for daytime minutes. As no evening minutes were used, no cost is allocated for the evening, and as only 1 minute was used on the weekend, the total cost comes to **50c**.

Test your program with the following cases:

1 a) 97 1 0

1 b) 38 72 94

1 c) 193 24 34

1 d) 378 342 919

Question 2: Rövarspråket

Mark's friend Jenny speaks a language called Rövarspråket. You must assist Mark by developing a program that will help him translate a given phrase into Rövarspråket. The rules are simple. For each word, every **consonant** is **doubled** up and the vowel 'o' is inserted in-between. Vowels and spaces are left intact.

The input phrase is given simply as a lowercase sentence. Output should be a lowercase sentence giving the Rövarspråket translation of the input.

Your task is to write a program that will provide you with the correct output.

Examples:

Input:	ruler	Output:	roruloleror
Input:	stubborn	Output:	sostotubobboborornon
Input:	caps lock	Output:	cocapopsos lolocockok

Test your program with the following cases:

2 a) computer

2 b) excellence

2 c) expanding possibilities

2 d) what a tangled web we weave

Question 3: Perfect

A classic problem in mathematics is to determine if a given positive integer is **perfect**. We call a positive integer, n, **perfect** if it is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself. (Note that a proper divisor of n is a positive integer **less** than n which **evenly divides** n. For example, the proper divisors of 6 are 1, 2 and 3, and the proper divisors of 8 are 1, 2 and 4).

Examples of **perfect** numbers include 6 and 28, noting that the proper divisors of 6 are 1, 2 and 3 and that 1 + 2 + 3 = 6. The proper divisors of 28 are 1, 2, 4, 7 and 14, and also that 1 + 2 + 4 + 7 + 14 = 28.

As the vast majority of numbers are not perfect, we can furthermore divide non-perfect numbers into two sets. We denote a positive integer n as **abundant** if the sum of its proper divisors is **greater** than n, otherwise we denote n as **deficient** if the sum of its proper divisors is **less** than n.

Note that 9 is a deficient number, since the proper divisors of 9 are 1 and 3, and 1 + 3 = 4 which is **less** than 9. Also, 12 is an abundant number, since the proper divisors of 12 are 1, 2, 3, 4 and 6, and 1 + 2 + 3 + 4 + 6 = 16 which is **greater** than 12.

Your task is to write a program that, given a single positive integer n, will determine if it is **perfect**, **deficient**, or **abundant**.

Examples:

Input:	12	Output:	Abundant
Input:	9	Output:	Deficient
Input:	28	Output:	Perfect

Test your program with the following cases:

3 a) 48

3 b) 110

3 c) 496

3 d) 945