# Exercises on Basic Syntax, Functions and Arrays

# Problem 1 – Operations

Write a program that reads two integer numbers from the console, then reads one of the following four instructions (as a single symbol): +, -, \*, / and performs the respective operation on the two numbers, with the first number as a left operand and the second number as a right operand (+ is addition, - is subtraction, \* is multiplication and / is division).

If the user enters a symbol different than one of the four operations, the program should print try again and allow the user to enter the operation again.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 3  - | -1 |
| 12 5  ?  A  5  \* | try again  try again  try again  60 |

# Problem 2 – Index Of

Write a program that reads an integer array from the console (given by size, followed by the elements of the array), followed by an integer number and prints the index of that number in the array on the console (if the number does not appear in the array, print -1).

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  1 2 3  2 | 1 |
| 4  101 42 13 69  69 | 3 |
| 5  -2 10 5 -14 35  14 | -1 |

# Problem 3 – Remove

Write a program that reads an integer array from the console (given by size, followed by the elements of the array), followed by an integer number and removes all occurrences of that number from the array, then prints the array

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 7  13 10 10 1 4 2 10  10 | 13 1 4 2 |
| 5  13 42 19 21 103  55 | 13 42 19 21 103 |

# Problem 4 – Noise

Write a program that reads a positive integer number and returns its square root (print the result as cout prints double numbers)

The number will be entered with “noise” in it, i.e. there will be symbols that are not digits. These symbols should be ignored. The last symbol of the input of the number will always be . (dot) and there will be no other . (dot) in the number.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 25. | 5 |
| ,,-2!!as\*\*dsa5\*-. | 5 |
| -9abc. | 3 |

## Problem 5 – Noise Test Generator

Write a program that generates test input and expected output data for Problem 4 – Noise. The program should generate a random input value on each execution, print that input value on the console on a single line, then print another line with the expected output value for that input. Hint: lookup srand() and rand()

### Examples

|  |
| --- |
| **Output** |
| 25.  5 |
| ,,-2!!as\*\*dsa5\*-.  5 |
| -9abc.  3 |

## Problem 6 - Triples of Latin Letters

Write a program to read an integer **n** and print all **triples** of the first **n small Latin letters**, ordered alphabetically:

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 | aaa  aab  aac  aba  abb  abc  aca  acb  acc  baa  bab  bac  bba  bbb  bbc  bca  bcb  bcc  caa  cab  cac  cba  cbb  cbc  cca  ccb  ccc |

## Problem 7 – Print Name of Number

Write a program that, given an integer number in the range [0, 9999], prints the name of that number in English.

Simplifications:

* Use lowercase English letters only
* Don’t place “and” (e.g. 957 is nine hundred fifty seven, NOT nine hundred and fifty seven)
* Skip 0 digits, except for the number 0 (e.g. 0 -> zero; 101 -> one hundred one; 1001 -> one thousand one)
* Don’t print dashes (e.g. print 75 as seventy five, NOT seventy-five)

|  |  |
| --- | --- |
| **Input** | **Output** |
| 0 | zero |
| 101 | one hundred one |
| 957 | nine hundred fifty seven |

## Problem 8 – Multiply Evens Sum by Odds Sum

Create a program that reads an **integer number** and **multiplies the sum of all its even digits** by **the sum of all its odd digits**:

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 12345 | 54 | 12345 has **2 even digits** - 2 and 4. Even digits has **sum of 6**.  Also it has **3 odd digits** - 1, 3 and 5. Odd digits has **sum of 9**.  **Multiply 6 by 9** and you get **54**. |
| -12345 | 54 |  |

## Problem 9 – Sum Adjacent Equal Numbers

Write a program to sum all adjacent equal numbers in an array of decimal numbers, starting from left to right.

* After two numbers are summed, the obtained result could be equal to some of its neighbors and should be summed as well (see the examples below).
* Always sum the leftmost two equal neighbors (if several couples of equal neighbors are available).

The array will be defined by two lines – the first line will contain the size of the array and the second will contain the elements of the array.

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Explanation** |
| 4  3 3 6 1 | 12 1 | **3 3** 6 1 🡪 **6 6** 1 🡪 12 1 |
| 6  8 2 2 4 8 16 | 16 8 16 | 8 **2 2** 4 8 16 🡪 8 **4 4** 8 16 🡪 **8 8** 8 16 🡪 16 8 16 |
| 6  5 4 2 1 1 4 | 5 8 4 | 5 4 2 **1 1** 4 🡪 5 4 **2 2** 4 🡪 5 **4 4** 4 🡪 5 8 4 |