

**CS101 Lab Quiz 1 - Batch A**  
**15 April 2024 - 20:30 hrs to 23:00 hrs**  
**3 Compulsory Questions - 40 Marks (10, 10, 20)**

**Instructions:**

- Keep your ID card on the table for ready reference. If your ID card isn't with you, you won't be allowed to appear for the quiz/exam.
  - Keep your phones, tablets, notes, bags, books, etc., near the instructor's platform.
  - Rough sheets will be provided to you.
  - **Create a folder on your Desktop and name it submission\_YourRollNumber**  
**E.g.: If my roll number is 23k1234 then my folder name is submission\_23k1234**
  - **Create all three programs in the newly created folder.**
  - Name the program files as mentioned in this pdf only.
  - No clarifications will be provided for any question by anyone (TAs/Instructor). When in doubt, make suitable assumptions, state them clearly as comments in your program file itself, and proceed to solve the problem.
  - Please note that your answers should NOT include any programming concept that hasn't been covered in the class. You are not allowed to use any advanced concepts of C/C++ like strings, vectors, etc.. Such solutions if found will NOT be graded.
  - Marks will be given for each hidden test case that passes.
  - At this stage of the course, we expect you to write correct code and fix compilation and logical errors by yourselves. Hence, there will not be any partial marks for any errors. Marks will be given for each hidden test case that passes. Cries like only one semicolon is missing or instead of <, <= should have been written, etc. will NOT be considered. TAs will not make any changes to your program. It is your responsibility to make it work.
  - TAs would be around to help you with respect to the logistics like saving programs, compiling, submitting, etc. They will not help you debug the error or resolve your issues related to syntax/logical errors in your program.
  - All the Best!
-

Write a C++ program to take an integer N as input from the user and then accept elements of a matrix of size NxN. The elements of a matrix entered by the user will be floating point numbers (positive, negative, or zero). Thereafter, compute the square of the matrix just entered and print it in the matrix format. Recall that the square of matrix **A** is the product, **AA**. Go through the example for more details.

**Input Format:**

- The first line is an integer N denoting the size of the matrix.
- N subsequent lines, each representing the elements of an NxN matrix. Every line contains N floating point numbers, each separated by a space

**Output Format:**

- Display the square of the matrix.
- The output should consist of N lines, each representing a row in the matrix. Within each row, the numbers should be separated by a space.

**Assume the following:**

- The value of variable N entered by the user will always be between 1 and 10 (both inclusive)
- Each element of the matrix entered by the user will be in between -1000.0 and 1000.0 (both inclusive)
- You need not write any conditional statements to check these values. Assume that the user will enter only the values as mentioned above.

**Example Input**

```
2
3 -2
1.5 -0.5
```

**Example Output**

```
6 -5
3.75 -2.75
```

**Explanation**

Matrix of size 2x2 input by the user is

$$\begin{pmatrix} 3 & -2 \\ 1.5 & -0.5 \end{pmatrix}$$

The square of the matrix is

$$\begin{pmatrix} 3 & -2 \\ 1.5 & -0.5 \end{pmatrix} \times \begin{pmatrix} 3 & -2 \\ 1.5 & -0.5 \end{pmatrix} = \begin{pmatrix} 6 & -5 \\ 3.75 & -2.5 \end{pmatrix}$$

**Note:**

- Do not write any C++ statements for printing general messages. For example, the following **should NOT** be present in your program:
  - a. `cout << "Enter a number:"`,
  - b. `cout << "The computed answer is"`, etc.

In addition, **do not** print unnecessary spaces unless specified in the program.

# Practice Test Cases:

Input	Output
2 1 0 0 1	1 0 0 1
6 9.1 -4.8 -6.4 1.7 -4.7 9.8 8.3 -6.9 1.7 9.9 8.6 8 -0.7 -9.2 -8.7 4.2 9.1 9.5 -3.3 9.5 -9.5 -7.7 -4.2 -4.2 -6.3 0.5 4.3 -5 -3.1 2.2 -5.1 -9 2.4 -1.4 2.7 7.7	21.47 -26.08 -23.56 -62.24 -108.4 47.96 -110.58 18.48 -117.51 -177.49 -129.52 81.23 -196.28 105.83 86.56 -219.95 -175.2 -87.58 128.76 0.239998 164.93 134.71 44.78 -55.83 -31.38 -81.62 43.21 63.22 109.59 14.23 -174.45 -16.75 39.85 -101.19 -13.29 -28.07
3 1 2 3 1 2 3 1 2 3	6 12 18 6 12 18 6 12 18
4 5.4 9.4 9.8 6 7.5 9.7 -0.1 3.3 -1 -0.1 2 3.8 -9.9 -9.1 -4.5 3.6	30.46 86.36 44.58 122.26 80.68 134.57 57.48 88.51 -45.77 -45.15 -22.89 14.95 -152.85 -213.64 -121.31 -93.57
3 0 0 0 1 34 21 1 2 3	0 0 0 55 1198 777 5 74 51
6 26 72 39 33 48 93 52 14 75 33 90 93 93 0 45 24 85 85 86 23 81 12 7 55 91 63 16 49 93 68 26 72 39 33 48 93	17671 13359 15237 9987 20202 26157 22501 17065 14193 11853 23196 29097 18612 18723 12271 11407 20442 27479 14064 11191 11953 7843 16458 23273 21575 19316 17103 12855 23654 31025 17671 13359 15237 9987 20202 26157
8 17 90 36 78 5 37 73 8 77 21 77 63 99 58 4 85 42 1 45 41 24 64 9 5 1 64 82 4 2 25 51 71 16 39 49 73 29 23 33 6 44 11 21 44 21 77 81 63 78 78 56 54 42 66 59 96 82 42 0 84 19 3 8 87	16867 15080 20668 15391 14155 17909 13436 23569 17641 19861 19313 27873 10310 17832 18492 19286 7034 9022 10000 10822 3984 11606 12113 9032 19325 9085 12461 17452 12391 14647 6939 18802 9948 11190 15522 11421 8087 12449 10303 14184 17729 17658 14166 19749 8726 16657 17902 22750 25788 23168 22510 30730 16470 22574 20245 29794 12906 18690 14516 18637 7339 9027 12476 18830

10	27121	37588	32810	27627	14901	34719	20064	34892
59 67 94 58 7 50 88 29 79 93	27994	33535						
31 91 19 26 5 29 8 36 37 7	11626	17806	14465	13800	7389	12698	9726	15905 15837
49 10 4 31 39 96 66 85 48 27	13596							
64 60 46 10 18 6 12 58 35 91	19566	24834	29149	24637	16340	25273	17817	25058
52 19 34 23 97 40 52 5 28 41	24696	30301						
12 29 52 68 61 91 16 79 76 65	17875	25786	20762	21752	8856	24146	17222	22603
58 92 77 5 2 47 11 14 5 98	18983	18077						
58 57 69 92 32 66 84 85 71 65	17870	19962	20524	15355	15962	21020	15045	15895
26 35 94 78 56 7 21 72 86 50	16478	22157						
37 97 42 66 2 97 13 65 11 19	22662	26781	28378	28899	21895	30365	19417	33886
	28461	29065						
	16241	26220	16649	19636	7966	28329	14514	23626
	18042	15809						
	30967	39168	37300	31197	19162	35375	24338	38047
	32411	40172						
	24693	25710	26110	24613	18548	25875	21467	31217
	25854	27038						
	18253	25666	20955	21429	12441	24819	15333	27370
	24101	24057						

The Mandelbrot set is a set of complex numbers  $\{Z_0, Z_1, Z_2 \dots\}$

It is generated using the iteration/recursion of a simple mathematical formula given below

$$Z_{n+1} = Z_n^2 + C$$

for  $n \geq 0$  where  $Z_0 = 0$

and

$C = (a + bi)$ , a complex number

Write a C++ program to accept a complex number from the user i.e. two floating point numbers (real and imaginary part) and an integer  $n$  from the user. Compute and print the value of  $Z_n$ .

**Input Format:**

- The first line consists of two space-separated decimal values denoting real and imaginary parts of a complex number.
- The second line consists of a single integer  $n$

**Output Format:**

- The output should be two numbers separated by a space.
- The first number denotes the real part, and the second the imaginary part.

**Assume the following:**

- The value of variable  $n$  entered by the user will always be between 1 and 20 (both inclusive)
- The real and imaginary parts of the complex number  $C$ , entered by the user will be between -50.0 and 50.0 (both inclusive) and can be real-valued floating point numbers.
- You need not write any conditional statements to check these values. Assume that the user will enter only the values as mentioned above.

**Example Input**

1 1

2

**Example Output**

1 3

**Explanation:**

$$Z_0 = 0, C = 1 + 1i$$

$$Z_1 = Z_0^2 + C = 0^2 + (1 + i) = 1 + i$$

$$Z_2 = Z_1^2 + C = (1 + i)^2 + (1 + i) = 1 + 3i$$

Here the real part is 1, and the imaginary part is 3

**Note:**

- Do not write any C++ statements for printing general messages. For example, the following **should NOT** be present in your program:
  - `cout << "Enter a number:"`,
  - `cout << "The computed answer is"`, etc.

In addition, **do not** print unnecessary spaces unless specified in the program.

**Practice Test Cases:**

Input	Output
1 1 5	-9407 -193
2.5 -3.5 3	-426.25 143.5
-1 1 3	-1 3
0 0 20	0 0
0.5 0.5 6	9.48659 9.32886
3 3 7	-1.02529e+41 2.62877e+42
1 0.5 10	-2.94139e+99 -7.65478e+97
3 4 9	2.73255e+185 -3.64759e+185

Write a C++ program to first accept a paragraph **P** as input from the user and store it in a character array. Thereafter, accept a word **W** from the user and store it in another character array. Find out all the occurrences of the word **W** in paragraph **P**. Note that the word **W** can also be a substring of a larger word. Print the start index of each occurrence. See the example for more details.

**Input Format:**

- The first input consists of a character array **P** that denotes a paragraph
- The second input consists of a character array **W** that denotes a word
- `cin.getline(sen, 1000);` is the syntax to input a character array called 'sen'.

**Output Format:**

- Print the starting index of each occurrence on a new line

**Assume the following:**

- For **W**, the user enters only valid lowercase alphabets
- For **P**, the user enters only valid lowercase alphabets and special characters.
- The max length of a **P** and **W** is 1000
- There will always be at least one occurrence of **W** in **P**

**Example Input**

**atmatcat** is the name of my dog who stares **at** the **cat**s and sits on a **mat**  
at

**Example Output**

0  
3  
6  
42  
50  
69

**Explanation**

**W** i.e. 'at' occurs six times, starting at 0, followed by 3, 6, 42, 50, and 69.

Note that the spaces matter. The **W** needs to align with a single word in the paragraph and not multiple words with spaces in between them. Example: If **P** is "the technician examined the **black box**, noting its sealed exterior. **blackbox** was not tampered" and **W** is "blackbox", then "black box" is not considered, and there is only one occurrence of the **W** in **P**.

**Note:**

- Do not write any C++ statements for printing general messages. For example, the following **should NOT** be present in your program:
  - a. `cout << "Enter a number:"`,
  - b. `cout << "The computed answer is"`, etc.

In addition, **do not** print unnecessary spaces unless specified in the program.

**Practice Test Cases:**

Input	Output
cat is a cat! cat	0 9
own occurs in brown. own	0 16
i say 0000. 0000. 0000. ooo	6 7 12 13 18 19
the technician examined the black box, noting its sealed exterior. blackbox was not tampered blackbox	67
i love dogs. dogs love me. i love cats too, but cats dont love me. dog	7 13
i am going to pass this test case. this test case is not the test case that i call a failing test case. case	29 45 66 98
in the bustling city, where skyscrapers pierce the sky and streets teem with life, the word "vibrant" echoes through conversations like a chorus. vibrant, describing the city's energy, its kaleidoscope of cultures, and its relentless pace, resonates deeply among its inhabitants. from the vibrant markets brimming with exotic goods to the vibrant nightlife pulsating with music and laughter, this word encapsulates the essence of urban life. it symbolizes the city's dynamism, its ever-changing landscape, and its ability to inspire and captivate. in this vibrant metropolis, where dreams collide and possibilities abound, the word "vibrant" reigns supreme, embodying the spirit of endless possibilities. vibrant	93 146 289 339 556 633
in the tranquil countryside, where rolling hills embrace the horizon and meadows sway in the gentle breeze, the word "serenity" holds sway over the landscape. serenity, evoking a sense of peace and calm, permeates the air like the fragrance of wildflowers. it whispers through the rustling leaves of ancient trees and dances across the surface of serene lakes. serenity, with its tranquil charm, invites contemplation and introspection, offering solace to weary souls seeking refuge from the chaos of modern life. In this idyllic countryside, where time seems to slow and nature reigns supreme, the word "serenity" reigns as a reminder of the beauty of simplicity and the power of stillness. serenity	118 159 361 605