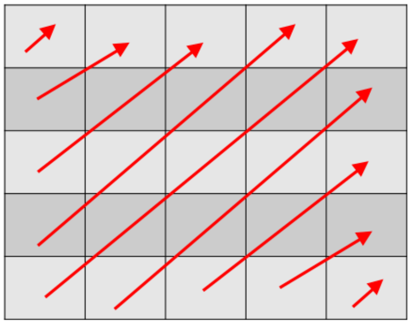
# CS101 Lab Quiz 2 - Batch C
## 18 April 2024 - 20:30 hrs to 23:00 hrs
## 3 Compulsory Questions - 40 Marks (10, 10, 20)

**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. Cribs 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!

## Filename: LQ02_C_Q1.cpp Traverse the matrix along the diagonals (10 Marks)

Write a C++ program to take a positive integer N as input from the user, followed by the elements of an integer matrix of size NxN. Traverse and print the matrix as shown below, starting from index 0,0, then going from (1,0 to 0,1), followed by (2,0 to 1,1 to 0,2), and so on, until (N - 1, N - 1).



**Input Format**

- The first line contains an integer N
- The subsequent N lines contain the elements of a matrix.
- Each line has N space-separated integers.

**Output Format:**

- Every diagonal should be on a new line.
- All the integers traversed along the same diagonal should be printed in the same line separated by space.

**Assume the following:**

- Assume that the value of N entered by the user will always be in between 1 and 30 (both inclusive)
- Assume that the value of each element of the matrix entered by the user will always be in between 0 and 500 (both inclusive)

***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 |
|---|---|
| 4<br>1 2 3 4<br>5 6 7 8<br>9 10 11 12<br>13 14 15 16 | 1<br>5 2<br>9 6 3<br>13 10 7 4<br>14 11 8<br>15 12<br>16 |

| | |
|---|---|
| 3<br>1 2 3<br>4 5 6<br>7 8 9 | 1<br>4 2<br>7 5 3<br>8 6<br>9 |
| 2<br>1 2<br>3 4 | 1<br>3 2<br>4 |
| 1<br>5 | 5 |
| 30<br>31 16 7 25 62 24 73 55 84 93 36 44 92 14 71 65 8 96 6 55 26 86 23 7 20 36 64 92 55 82<br>1 38 50 9 63 13 85 88 68 70 33 5 14 26 71 37 43 79 34 2 87 12 40 10 19 60 98 35 53 53<br>18 6 92 20 67 55 33 5 96 54 75 29 11 41 55 82 78 51 14 64 53 1 76 93 63 48 54 61 83 59<br>67 53 17 59 74 37 66 59 42 62 65 17 92 28 10 99 11 40 50 77 5 55 30 33 1 45 81 55 58 65<br>66 25 18 35 84 44 72 51 4 14 65 21 83 57 50 45 9 61 86 11 90 43 67 72 76 68 17 10 75 75<br>75 93 1 45 28 37 42 1 40 98 67 6 19 3 15 69 48 24 82 34 36 72 29 55 44 6 75 61 16 2<br>89 43 47 42 40 75 79 82 28 72 80 48 78 0 51 93 21 99 18 4 86 6 76 15 13 73 73 88 86 89<br>42 75 84 89 69 77 64 49 59 45 21 92 93 51 44 44 44 65 95 14 69 81 72 98 49 85 23 22 73 9<br>12 67 37 48 56 6 25 73 55 37 18 28 81 11 31 25 7 76 42 2 42 64 36 15 62 85 0 85 7 26<br>46 71 93 83 72 50 90 97 75 97 86 93 78 67 56 9 92 15 37 87 69 80 51 5 47 65 42 47 2 2<br>25 48 73 71 84 45 21 74 95 96 23 81 41 1 49 97 11 93 12 0 80 81 80 83 39 79 0 81 27 2 | 31<br>1 16<br>18 38 7<br>67 6 50 25<br>66 53 92 9 62<br>75 25 17 20 63 24<br>89 93 18 59 67 13 73<br>42 43 1 35 74 55 85 55<br>12 75 47 45 84 37 33 88 84<br>46 67 84 42 28 44 66 5 68 93<br>25 71 37 89 40 37 72 59 96 70 36<br>83 48 93 48 69 75 42 51 42 54 33 44<br>0 4 73 83 56 77 79 1 4 62 75 5 92<br>80 92 3 71 72 6 64 82 40 14 65 29 14 14<br>19 67 49 9 84 50 25 49 28 98 65 17 11 26 71<br>44 65 59 3 75 45 90 73 59 72 67 21 92 41 71 65<br>0 94 57 29 1 87 21 97 55 45 80 6 83 28 55 37 8<br>23 23 51 79 23 24 6 74 75 37 21 48 19 57 10 82 43 96<br>56 47 7 53 47 12 42 48 95 97 18 92 78 3 50 99 78 79 6<br>22 81 31 3 73 80 6 59 13 96 86 28 93 0 15 45 11 51 34 55<br>1 77 69 82 28 50 91 65 73 1 23 93 81 51 51 69 9 40 14 2 26<br>84 96 90 87 51 32 86 53 72 55 96 81 78 11 44 93 48 61 50 64 87 86<br>35 45 12 43 63 60 5 17 98 79 61 36 41 67 31 44 21 24 86 77 53 12 23<br>8 52 90 91 64 72 15 14 55 15 73 21 35 1 56 25 44 99 82 11 5 1 40 7<br>79 92 73 96 39 5 99 56 49 84 84 85 44 37 49 9 7 65 18 34 90 55 76 10 20 |

| | |
|---|---|
| 83 4 3 9 75 87 6 48 13 1 96 36 35 37 38 84 86 1 29 50 1 10 84 34 45 23 13 46 56 92<br>0 92 49 3 1 24 42 59 73 55 61 21 44 96 59 82 32 97 35 61 48 36 23 84 22 69 7 36 67 15<br>80 67 59 29 23 12 6 65 72 79 73 85 52 17 81 11 51 65 61 86 26 61 74 2 45 97 23 4 85 90<br>19 65 57 79 47 80 91 53 98 15 84 71 0 36 88 33 0 39 98 13 77 77 74 51 79 19 48 2 23 33<br>44 94 51 53 73 50 86 17 55 84 32 39 7 33 27 47 66 79 38 17 92 15 94 18 66 25 37 67 79 60<br>0 23 7 3 28 32 5 14 49 60 50 34 51 57 19 31 4 37 10 94 6 55 9 0 73 28 77 11 95 56<br>23 47 31 82 51 60 15 56 26 16 69 77 50 20 86 21 51 43 11 62 37 17 69 99 70 42 27 47 5 22<br>56 81 69 87 63 72 99 78 81 26 95 50 55 97 70 41 71 22 84 34 36 22 51 57 21 21 51 0 21 57<br>22 77 90 43 64 5 68 64 36 49 42 83 99 97 80 21 38 3 95 75 37 83 49 41 40 22 62 92 22 35<br>1 96 12 91 39 77 96 7 93 32 8 87 15 59 84 48 81 74 3 76 49 93 60 98 34 52 20 48 44 42<br>84 45 90 96 36 30 25 85 89 70 69 98 57 37 57 93 37 90 68 40 19 69 33 31 20 19 83 40 68 28<br>35 52 73 25 0 62 7 78 99 97 0 68 47 58 57 4 3 94 95 23 35 66 93 20 97 13 40 80 5 8<br>8 92 12 34 70 64 48 77 94 99 26 95 19 73 5 77 30 8 71 77 84 58 43 77 79 40 42 19 20 99<br>79 81 92 43 67 14 59 67 91 54 66 70 49 85 95 6 62 25 14 34 54 50 92 49 27 23 89 21 94 62<br>73 73 95 65 68 62 79 28 29 22 82 47 44 83 32 40 89 95 17 55 81 72 6 25 21 85 1 63 59 95 | 73 81 12 25 36 77 68 78 26 60 32 71 52 96 38 97 92 76 95 4 36 43 30 93 19 36<br>25 73 92 34 0 30 96 64 81 16 50 39 0 17 59 84 11 15 42 14 86 72 67 33 63 60 64<br>0 32 95 43 70 62 25 7 36 26 69 34 7 36 81 82 86 93 37 2 69 6 29 72 1 48 98 92<br>2 42 21 65 67 64 7 85 93 49 95 77 51 33 88 11 32 1 12 87 42 81 76 55 76 45 54 35 55<br>49 65 21 72 68 14 48 78 89 32 42 50 50 57 27 33 51 97 29 0 69 64 72 15 44 68 81 61 53 82<br>44 49 73 49 62 59 77 99 70 8 83 55 20 19 47 0 65 35 50 80 80 36 98 13 6 17 55 83 53<br>20 23 66 89 79 67 94 97 69 87 99 97 86 31 66 39 61 61 1 81 51 15 49 73 75 10 58 59<br>98 38 22 86 28 91 99 0 98 15 97 70 21 4 79 98 86 48 10 80 5 62 85 73 61 75 65<br>19 15 63 80 29 54 26 68 57 59 80 41 51 37 38 13 26 36 84 83 47 85 23 88 16 75<br>10 98 52 69 22 66 95 47 37 84 21 71 43 10 17 77 61 23 34 39 65 0 22 86 2<br>13 53 2 67 82 70 19 58 57 48 38 22 11 94 92 77 74 84 45 79 42 85 73 89<br>17 67 84 54 47 49 73 57 93 81 3 84 62 6 15 74 2 22 23 0 47 7 9<br>64 0 19 3 44 85 5 4 37 74 95 34 37 55 94 51 45 69 13 81 2 26<br>32 38 57 14 83 95 77 3 90 3 75 36 17 9 18 79 97 7 46 27 2<br>18 86 88 99 32 6 30 94 68 76 37 22 69 0 66 19 23 36 56 2<br>2 57 33 86 40 62 8 95 40 49 83 51 99 73 25 48 4 67 92<br>70 26 56 46 89 25 71 23 19 93 49 57 70 28 37 2 85 15<br>27 71 74 39 95 14 77 35 69 60 41 21 42 77 67 23 90<br>80 65 31 27 17 34 84 66 33 98 40 21 27 11 79 33<br>41 52 47 93 55 54 58 93 31 34 22 51 47 95 60<br>93 2 54 56 81 50 43 20 20 52 62 0 5 56<br>84 12 25 35 72 92 77 97 19 20 92 21 22<br>95 6 3 26 6 49 79 13 83 48 22 57<br>57 79 89 80 25 27 40 40 40 44 35<br>91 68 3 41 21 23 42 80 68 42<br>75 95 84 4 85 89 19 5 28<br>25 83 82 2 1 21 20 8 |

| | |
|---|---|
| 25 32 21 72 49 89 86 80 69 67 54 3 14 99 86 46 39 27 93 56 35 26 80 41 4 2 78 5 65 89<br>0 42 21 73 66 22 63 52 2 84 19 57 88 33 56 74 31 47 54 25 3 89 3 84 82 7 38 60 12 55<br>2 65 49 23 38 15 98 53 67 0 38 86 57 26 71 65 52 2 12 6 79 68 95 83 52 29 90 42 42 55<br>49 44 20 98 19 10 13 17 64 32 18 2 70 27 80 41 93 84 95 57 91 75 25 38 58 29 68 0 71 10 | 38 52 7 78 63 94 99<br>58 29 38 5 59 62<br>29 90 60 65 95<br>68 42 12 89<br>0 42 55<br>71 55<br>10 |
| 12<br>99 59 85 37 57 77 92 58 14 75 66 39<br>48 3 8 56 41 75 59 7 81 12 97 85<br>93 60 17 64 40 41 42 39 52 79 29 61<br>56 21 19 23 48 37 62 49 92 22 57 33<br>97 68 40 30 81 89 15 74 2 32 91 42<br>25 33 82 29 12 63 42 20 84 61 43 32<br>50 57 81 42 79 91 76 28 59 68 59 40<br>58 26 67 60 11 58 54 88 91 36 18 55<br>99 60 75 35 74 19 20 24 28 53 19 60<br>44 95 88 4 15 99 44 25 26 11 37 37<br>21 92 25 64 28 95 19 80 8 95 67 82<br>66 87 58 46 93 77 6 37 24 95 41 92 | 99<br>48 59<br>93 3 85<br>56 60 8 37<br>97 21 17 56 57<br>25 68 19 64 41 77<br>50 33 40 23 40 75 92<br>58 57 82 30 48 41 59 58<br>99 26 81 29 81 37 42 7 14<br>44 60 67 42 12 89 62 39 81 75<br>21 95 75 60 79 63 15 49 52 12 66<br>66 92 88 35 11 91 42 74 92 79 97 39<br>87 25 4 74 58 76 20 2 22 29 85<br>58 64 15 19 54 28 84 32 57 61<br>46 28 99 20 88 59 61 91 33<br>93 95 44 24 91 68 43 42<br>77 19 25 28 36 59 32<br>6 80 26 53 18 40<br>37 8 11 19 55<br>24 95 37 60<br>95 67 37<br>41 82<br>92 |
| 9<br>41 66 19 43 94 50 35 74 1<br>53 29 23 47 54 14 30 27 81<br>59 12 30 82 58 56 66 39 92<br>43 83 60 59 24 78 30 68 72<br>80 3 46 33 9 27 57 8 81<br>71 38 8 52 50 72 82 32 30<br>90 51 22 35 94 5 95 53 29<br>73 36 49 97 16 5 95 50 14 | 41<br>53 66<br>59 29 19<br>43 12 23 43<br>80 83 30 47 94<br>71 3 60 82 54 50<br>90 38 46 59 58 14 35<br>73 51 8 33 24 56 30 74<br>22 36 22 52 9 78 66 27 1 |

| | |
|---|---|
| 22 7 22 3 30 12 63 35 62 | 7 49 35 50 27 30 39 81<br>22 97 94 72 57 68 92<br>3 16 5 82 8 72<br>30 5 95 32 81<br>12 95 53 30<br>63 50 29<br>35 14<br>62 |
| 21<br>15 12 64 68 0 79 47 72 78 9 38 35 90 93 34 66 4 67 21 83 60<br>49 64 87 15 61 55 33 45 14 12 60 26 76 28 78 8 27 50 86 36 40<br>21 78 34 8 45 90 75 66 73 36 15 37 75 83 99 82 16 96 96 28 56<br>74 57 36 4 65 15 55 51 51 47 24 30 81 32 75 71 60 41 96 48 9<br>34 75 92 85 57 8 81 53 88 89 79 45 77 83 10 92 90 13 95 38 38<br>77 19 22 52 43 34 46 91 82 7 77 57 99 62 14 7 95 19 47 36 50<br>93 65 34 55 9 76 69 5 14 59 82 86 33 87 81 68 85 72 2 92 50<br>60 91 12 26 50 60 46 97 48 96 42 14 82 50 23 59 71 28 73 82 63<br>11 15 2 92 35 87 65 38 31 15 50 74 79 76 76 91 22 25 40 71 68<br>54 53 18 77 64 41 58 38 23 21 1 38 75 94 26 62 11 64 45 26 14<br>19 57 42 95 49 65 20 41 88 40 95 93 10 24 58 51 82 48 74 55 49<br>65 30 43 91 44 54 55 89 32 21 60 42 63 55 43 80 28 84 20 68 79<br>14 31 55 24 82 90 72 9 97 21 74 28 17 65 24 23 72 14 8 45 74<br>50 60 82 93 93 10 77 13 30 8 79 61 63 3 96 5 75 5 3 49 79<br>31 66 44 7 41 68 73 49 13 0 51 25 82 44 18 44 73 32 74 81 63<br>88 97 67 84 2 94 41 5 43 20 88 61 16 96 3 84 69 4 49 69 56<br>74 3 0 93 99 74 77 26 7 40 14 4 59 50 7 54 91 64 97 11 53<br>59 79 1 14 63 70 18 64 92 26 90 47 27 83 47 53 12 73 60 5 87 | 15<br>49 12<br>21 64 64<br>74 78 87 68<br>34 57 34 15 0<br>77 75 36 8 61 79<br>93 19 92 4 45 55 47<br>60 65 22 85 65 90 33 72<br>11 91 34 52 57 15 75 45 78<br>54 15 12 55 43 8 55 66 14 9<br>19 53 2 26 9 34 81 51 73 12 38<br>65 57 18 92 50 76 46 53 51 36 60 35<br>14 30 42 77 35 60 69 91 88 47 15 26 90<br>50 31 43 95 64 87 46 5 82 89 24 37 76 93<br>31 60 55 91 49 41 65 97 14 7 79 30 75 28 34<br>88 66 82 24 44 65 58 38 48 59 77 45 81 83 78 66<br>74 97 44 93 82 54 20 38 31 96 82 57 77 32 99 8 4<br>59 3 67 7 93 90 55 41 23 15 42 86 99 83 75 82 27 67<br>17 79 0 84 41 10 72 89 88 21 50 14 33 62 10 71 16 50 21<br>8 64 1 93 2 68 77 9 32 40 1 74 82 87 14 92 60 96 86 83<br>5 69 89 14 99 94 73 13 97 21 95 38 79 50 81 7 90 41 96 36 60<br>60 27 24 63 74 41 49 30 21 60 93 75 76 23 68 95 13 96 28 40<br>75 55 18 70 77 5 13 8 74 42 10 94 76 59 85 19 95 48 56<br>94 74 80 18 26 43 0 79 28 63 24 26 91 71 72 47 38 9<br>29 91 40 64 7 20 51 61 17 55 58 62 22 28 2 36 38<br>91 80 68 92 40 88 25 63 65 43 51 11 25 73 92 50 |

| | |
|---|---|
| 17 64 89 24 18 80 40 68 43 93 79 22 94 45 37 17 63 53 61 90 43<br>8 69 27 55 74 91 80 34 96 67 51 13 8 27 31 88 20 51 83 13 30<br>5 60 75 94 29 91 47 90 33 91 50 54 70 58 28 61 38 14 10 58 66 | 47 34 43 26 14 61 82 3 24 80 82 64 40 82 50<br>90 96 93 90 4 16 44 96 23 28 48 45 71 63<br>33 67 79 47 59 96 18 5 72 84 74 26 68<br>91 51 22 27 50 3 44 75 14 20 55 14<br>50 13 94 83 7 84 73 5 8 68 49<br>54 8 45 47 54 69 32 3 45 79<br>70 27 37 53 91 4 74 49 74<br>58 31 17 12 64 49 81 79<br>28 88 63 73 97 69 63<br>61 20 53 60 11 56<br>38 51 61 5 53<br>14 83 90 87<br>10 13 43<br>58 30<br>66 |

## Filename: LQ02_C_Q2.cpp Operation on String (10 Marks)

Write a C++ program to first accept a paragraph as input from the user. The paragraph may contain digits, lower and upper case letters, and other special characters (such as ‘,’, ‘;’, ‘$’, ‘%’, etc.). The words of the paragraph will be separated by exactly one space.

Afterward, accept a positive integer W. This indicates the width of the page (i.e. max number of characters that can be fit onto a line, including spaces, letters, digits, and special characters. Print the paragraph, character by character. As soon as the width of the page exceeds, print the next character on a new line. This may involve breaking of words, which is fine. Spaces also need to be printed. Have a look at the example input and output for more details.

**Example Input**:
Traffic siGnals are lights used to coNtRol the moveMENT of traffic.
15

**Example Output**

```
Traffic siGnals
 are lights use
d to coNtRol th
e moveMENT of t
raffic.
```

**Explanation**
Look at the table given below. For your reference, the columns of the line are denoted in blue and the spaces are denoted in gray.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T | r | a | f | f | i | c | ▨ | s | i | G | n | a | l | s |
| ▨ | a | r | e | ▨ | l | i | g | h | t | s | ▨ | u | s | e |
| d | ▨ | t | o | ▨ | c | o | N | t | R | o | l | ▨ | t | h |
| e | ▨ | m | o | v | e | M | E | N | T | ▨ | o | f | ▨ | t |
| r | a | f | f | i | c | . |  |  |  |  |  |  |  |  |

**Input Format**

- The first input is a paragraph as described above
- The second input is a positive integer W
- cin.getline(sen, 1000); is the syntax to input a character array called ‘sen’.

**Output Format**

- A character array printing the paragraph as described

**Assume the following**

- The paragraph entered by the user will have digits, lowercase and uppercase letters and special characters
- Assume that the value of W entered by the user will be between 1 and 50 (both inclusive)
- Assume that the length of the maximum word in the paragraph will always be less than W

**Practice Test cases**

| Input | Output |
|---|---|
| `Traffic siGnals are lights used to coNtRol the moveMENT of traffic.`<br>`15` | `Traffic siGnals`<br>` are lights use`<br>`d to coNtRol th`<br>`e moveMENT of t`<br>`raffic.` |
| `Popeye the Sailor Man is one of the most popular cartoon characters of all time.`<br>`20` | `Popeye the Sailor Ma`<br>`n is one of the most`<br>` popular cartoon cha`<br>`racters of all time.` |
| `Weather stations around the world measure different parts of weather.`<br>`16` | `Weather stations`<br>` around the worl`<br>`d measure differ`<br>`ent parts of wea`<br>`ther.` |
| `The air always has some water mixed with it.`<br>`15` | `The air always`<br>`has some water`<br>`mixed with it.` |
| `Flower is from the Middle English flour, which referred to both the ground grain and the reproductive structure in plants, before splitting off in the 17th century. It comes originally from the Latin name of the Italian goddess of flowers, Flora.`<br>`40` | `Flower is from the Middle English flour,`<br>` which referred to both the ground grain`<br>` and the reproductive structure in plant`<br>`s, before splitting off in the 17th cent`<br>`ury. It comes originally from the Latin`<br>`name of the Italian goddess of flowers,`<br>`Flora.` |

| | |
|---|---|
| Gardening can be done outside of the home, in city parks, botanical gardens, zoos, amusement parks, theme parks, and around tourist attractions. These types of gardens are cared for by people called gardeners or groundskeepers.<br>30 | Gardening can be done outside of the home, in city parks, botanical gardens, zoos, amusement parks, theme parks, and around tourist attractions. These types of gardens are cared for by people called gardeners or groundskeepers. |
| Gardening is the growing of plants such as flowers, shrubs and trees as a hobby or recreation. Some people also grow vegetables or fruit in their gardens. People do gardening outdoors in the soil in their backyard, or in pots or containers on their balcony or on their patio. Some people do gardening on a roof.<br>34 | Gardening is the growing of plants such as flowers, shrubs and trees as a hobby or recreation. Some people also grow vegetables or fruit in their gardens. People do gardening outdoors in the soil in their backyard, or in pots or containers on their balcony or on their patio. Some people do gardening on a roof. |
| People do indoor gardening inside their house or inside a building. Sometimes indoor gardening is done in greenhouses, which are special buildings where plants are grown. A greenhouse has a transparent glass or plastic roof and walls that let sunlight in. Grow tent is also used for indoor plants. Grow tent helps plant to grow in a more spectacular and develop a friendly environment. Water gardening is growing plants in ornamental water pools and ponds. People doing water gardening plant water lilies and other aquatic plants.<br>45 | People do indoor gardening inside their house or inside a building. Sometimes indoor gardening is done in greenhouses, which are special buildings where plants are grown. A greenhouse has a transparent glass or plastic roof and walls that let sunlight in. Grow tent is also used for indoor plants. Grow tent helps plant to grow in a more spectacular and develop a friendly environment. Water gardening is growing plants in ornamental water pools and ponds. People doing water gardening plant water lilies and other aquatic plants. |

**Filename: LQ02_C_Q3.cpp** **Knapsack Problem** **(20 Marks)**

In the knapsack problem, we have a knapsack (a type of bag) whose volume V is given. We also have a list of items, say 1, 2, …, N, that we would like to pack into the knapsack. Each item `i` has volume $V_i$ and a mass $M_i$. We can only pack a *subset* of items into the knapsack if their combined volume does not exceed V. Our aim is to pack a subset that fits in the knapsack, and has the maximum aggregate mass.

Mathematically, suppose S is a subset of {1, 2, …, N}. Let volume(S) denote the sum of the volumes of the items in S. Let mass(S) be the sum of the masses of the items in S. Our aim is to find

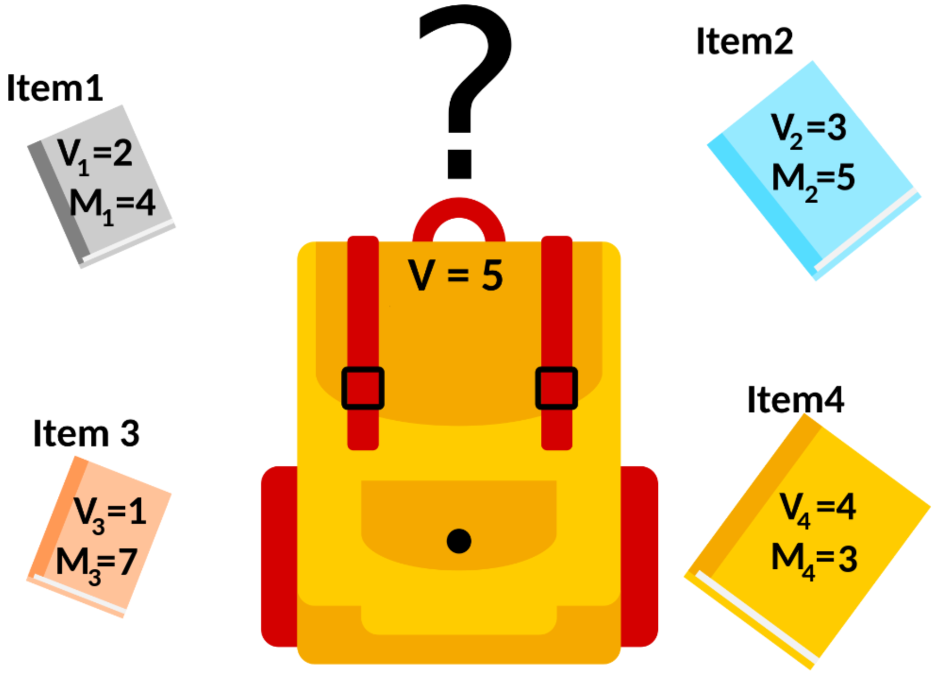
$$\underset{S\subseteq\{1,2,\ldots,n\},\text{volume}(S)\leq V}{\text{argmax}} \quad \text{mass}(S).$$

Write a C++ program to accept as input a variable N that denotes the total number of items, followed by two arrays, mass, and volume, for each of the N items, and finally, a variable, V, that denotes the volume of the knapsack.

Your task is to identify a set of items such that the sum of the masses associated with them is the maximum, while ensuring that their aggregate volume does not exceed V. Print the item numbers in ascending order.

**Let's take a look at an example. Consider N = 4.** Go through the figure given below the table.

| **Item** | **Mass** | **Volume** |
|---|---|---|
| 1 | 4 | 2 |
| 2 | 5 | 3 |
| 3 | 7 | 1 |
| 4 | 3 | 4 |
| **Note: The volume (V) of the knapsack is 5** | | |



To address this, we need to find a combination of items such that we have maximum mass and that volume does not exceed 5. The best subset is if we pack Item 2 and Item 3 as the total mass is 12 (5+7), and the capacity does not exceed (3+1). In any other combination i.e. (1 and 2) and (3 and 4) where the capacity does not exceed, the mass value is less than 12. Hence, we print the value 2 3.

This problem is not known to have any easy shortcuts. The best approach would be to enumerate the subsets of items, compare the ones satisfying the volume constraint, and selecting the one with the maximum total mass. Note that we restrict the number of items N to atmost 10

**Input Format**

- The first line is a positive integer N
- The second line is an array containing mass value for each of the N items, separated by a space
- The third line is an array containing volume value for each of the N items, separated by a space
- The fourth line is the volume capacity of the knapsack

**Output Format**

- Item numbers in increasing order, separated by a space.

**Assume**

- Assume that the value of N entered by the user will be between 1 and 10 (both inclusive)
- Assume that the value of capacity and each Mass and Volume entered by the user will be between 1 and 10000 (both inclusive)
- Assume that there will always be one solution (i.e. the best combination of masses producing the largest total mass and the capacity not exceeding)

**Practice Testcases**

| Input | Output |
|---|---|
| 4<br>4 5 3 7<br>50 120 100 20<br>100 | 1 4 |
| 5<br>2 3 4 5 6<br>10 20 30 40 50<br>100 | 1 2 3 4 |
| 6<br>4 5 3 7 6 8<br>100 80 75 85 110 120<br>150 | 6 |
| 7<br>35 24 11 8 25  18 12<br>53 42 11 8 52 81 21<br>88 | 1 3 7 |
| 10<br>20 25 30 35 40 45 50 55 60 65<br>15 18 21 24 27 30 33 36 39 42<br>100 | 6 7 8 |
| 7<br>1 14 2 13 3 12 4<br>6 6 6 6 6 6 6<br>18 | 2 4 6 |
| 9<br>20 25 30 75 80 85 90 95 100<br>15 18 21 48 51 54 57 60 63<br>75 | 3 6 |
| 7<br>92 4 43 83 84  6 44<br>7 22 14 22 10 11 12<br>35 | 1 5 7 |