***Maximum Capacity Consumed***

C371\_Coding\_October2022

**Topic**: Queue

**Difficulty Level:** Easy

**Question / Problem Statement**:

Jaziel has **2** rooms, and in each room, he has **N** bottles of mango juice, each bottle has a volume (in litres). The volume of **N** bottles for the first and second room is given in array **A** and array **B**.

Jaziel's task is to visit any room and drink the mango juice that contains the greatest litres in volume and whenever he drinks, that bottle becomes empty. The order of visiting rooms is given in array **V** of size **M** which contains integers, either 1 or 2 which means Jaziel visits either room 1 or room 2.

Write a program to find the total volume of mango juice Jaziel has drunk. If all the bottles are empty in any of the rooms and if Jaziel visits that room, then add 0 to the total volume.

**Note**

Jaziel may visit a room more than once.

Array **A** and **B** are in decreasing order.

**Function Description**

In the provided code snippet, implement the provided **maximumCapacityConsumed(...)** method using the variables to find the total volume of mango juice Jaziel has drunk. You can write your code in the space below the phrase **“WRITE YOUR LOGIC HERE”**.   
  
There will be multiple test cases running so the Input and Output should match exactly as provided.  
The base Output variable **result** is set to a default value of **-404** which can be modified. Additionally, you can add or remove these output variables.

**Input Format**

First line contains a pair of integers **N** and **M**.

Second line contains **M** integers separated by a single space - array **V**.

Third line contains **N** integers separated by a single space - array **A**.

Fourth line contains **N** integers separated by a single space - array **B**.

**Sample Input**

3 4 –denotes N and M.

1 2 2 1 –denotes array V.

5 4 3 –denotes array A.

3 2 1 –denotes array B.

**Constraints**

1 <= **N <=** 100.

1 <= **M** <= 100.

1 <= **Vi** <= 1000.

**Output Format**

Output should return the total volume of mango juice Jaziel has drunk.

**Sample Output**

14

**Explanation**

Initially the total volume was 0.

Given array V: [1 2 2 1], first visit room 1 and drink the mango juice that contains the greatest litres in volume in array A: [5 4 3] is 5. So, total volume = 5.

Now visit room 2, array B: [3 2 1], drink juice with volume 3. So, total volume = 8.

Now again visit room 2, array B: [2 1], drink juice with volume 2. So, total volume = 10.

Now again visit room 1, array A: [4 3], drink juice with volume 4. So, total volume = 14.

Finally, the total volume was 14 as we visited array V completely.

**Solution Steps**

1. Jaziel wants to drink the maximum volume of mango juice present in room 1 or in room 2. Both array A and B are in decreasing order so simply create two queues and store all the values of array A and B in queue 1 and queue 2.

2. Now the front value of the queue gives the maximum volume that Jaziel wants to drink so add that volume to total volume (initially total volume is 0) and remove it from the queue as it was drunk by Jaziel.

3. If any queue becomes empty then add 0 to total volume.

4. Finally return the total volume drunk by Jaziel.

**Running Solution in C++**:

#include <bits/stdc++.h>

using namespace std;

int main(){

//Declaring Variable N and M.

int N, M;

//take input N and M.

cin>>N>>M;

//create an array V of size M.

int V[M];

//taking array V inputs.

for(int idx=0;idx<M;idx++){

cin>>V[idx];

}

//create array A and B.

int A[N], B[N];

//taking array A inputs.

for(int idx=0;idx<N;idx++){

cin>>A[idx];

}

//taking array B inputs.

for(int idx=0;idx<N;idx++){

cin>>B[idx];

}

//create two queues for array A and B.

queue<int> qa, qb;

//insert array A elements in qa.

for(int idx=0;idx<N;idx++){

qa.push(A[idx]);

}

//insert array B elements in qb.

for(int idx=0;idx<N;idx++){

qb.push(B[idx]);

}

int total\_volume=0;

//now iterate to V array

for(int idx=0;idx<M;idx++){

if(V[idx]==1 && !qa.empty()){

//add front value which is maximum volume.

total\_volume+=qa.front();

//remove from the queue as it is drunk.

qa.pop();

}

else if(V[idx]==2 && !qb.empty()){

total\_volume+=qb.front();

qb.pop();

}

else

total\_volume+=0;

}

cout<<total\_volume<<"\n";

return 0;

}

Input:

3 6

1 2 2 1 2 2

5 4 1

7 3 1

Output:

20

**Test Cases [ Qty: 12]**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case No** | **Input** | **Output** | **Score** |
| 1 | 3 4  1 2 2 1  5 4 3  3 2 1 | 14 | 0 |
| 2 | 3 6  1 2 2 1 2 2  5 4 1  7 3 1 | 20 | 0 |
| 3 | 1 1  1  5  8 | 5 | 1 |
| 4 | 1 100  1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1  5  8 | 13 | 1 |
| 5 | 100 1  2  200 198 196 194 192 190 188 186 184 182 180 178 176 174 172 170 168 166 164 162 160 158 156 154 152 150 148 146 144 142 140 138 136 134 132 130 128 126 124 122 120 118 116 114 112 110 108 106 104 102 100 98 96 94 92 90 88 86 84 82 80 78 76 74 72 70 68 66 64 62 60 58 56 54 52 50 48 46 44 42 40 38 36 34 32 30 28 26 24 22 20 18 16 14 12 10 8 6 4 2  200 198 196 194 192 190 188 186 184 182 180 178 176 174 172 170 168 166 164 162 160 158 156 154 152 150 148 146 144 142 140 138 136 134 132 130 128 126 124 122 120 118 116 114 112 110 108 106 104 102 100 98 96 94 92 90 88 86 84 82 80 78 76 74 72 70 68 66 64 62 60 58 56 54 52 50 48 46 44 42 40 38 36 34 32 30 28 26 24 22 20 18 16 14 12 10 8 6 4 2 | 200 | 1 |
| 6 | 100 100  1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1  200 198 196 194 192 190 188 186 184 182 180 178 176 174 172 170 168 166 164 162 160 158 156 154 152 150 148 146 144 142 140 138 136 134 132 130 128 126 124 122 120 118 116 114 112 110 108 106 104 102 100 98 96 94 92 90 88 86 84 82 80 78 76 74 72 70 68 66 64 62 60 58 56 54 52 50 48 46 44 42 40 38 36 34 32 30 28 26 24 22 20 18 16 14 12 10 8 6 4 2  200 198 196 194 192 190 188 186 184 182 180 178 176 174 172 170 168 166 164 162 160 158 156 154 152 150 148 146 144 142 140 138 136 134 132 130 128 126 124 122 120 118 116 114 112 110 108 106 104 102 100 98 96 94 92 90 88 86 84 82 80 78 76 74 72 70 68 66 64 62 60 58 56 54 52 50 48 46 44 42 40 38 36 34 32 30 28 26 24 22 20 18 16 14 12 10 8 6 4 2 | 14300 | 1 |
| 7 | 8 12  1 2 1 1 1 2 2 1  90 88 78 64 54 45 44 40 38 30 25 15  80 78 68 54 44 35 34 30 28 20 15 10 | 447 | 1 |
| 8 | 100 100  1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 1  1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000  1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 | 100000 | 1 |
| 9 | 4 6  1 2 2 1 2 2  5 4 4 1  7 4 3 1 | 24 | 1 |
| 10 | 10 6  1 2 2 1 2 1  10 8 6 7 5 4 3 2 1 1  7 4 3 3 3 3 2 2 1 1 | 38 | 1 |
| 11 | 1 1  2  5  8 | 8 | 1 |
| 12 | 1 2  1 2  5  8 | 13 | 1 |

Plagiarism found – No

Clarity of the problem statement - Yes

Clarity of the example in the problem statement - Yes

Clarity of sample test cases - Yes

Clarity of test cases (Dual output) – Yes

Clarity of explanations - Yes

Provided Solution running – Yes

EEOC complaint (using abusive words/Indian Names/) - No

Similar Question in System - No

Difficulty Level – Easy

Question w.r.t Array / queue / two pointers concepts - Yes

Final Comment: **Accepted**