„Programming”
Big Project

Made by: Abrar Ahmed
Neptun code: OC7IP4
E-mail:abrarahmedpei@gmail.com

Course code: IP-18fPROGEG
Teacher’s name: Nikhazy Laszlo

2022. May 12.

Content

User documentation

Task

We have the result of a fishing competition in a matrix: M (i, j) means that fisher i caught M (i, j) pieces of fish type j.

Create a program that calculates how much the fishers caught of each type of fish.

Runtime environment

A Huawei PC that can run exe files, 64-bit operating system (e.g., Windows 11). No mouse needed.

Usage

Starting the program

The program can be found in the archived file by the name B3\bin\Release\a.exe. You can start the program by clicking the B3.exe file.

Program input

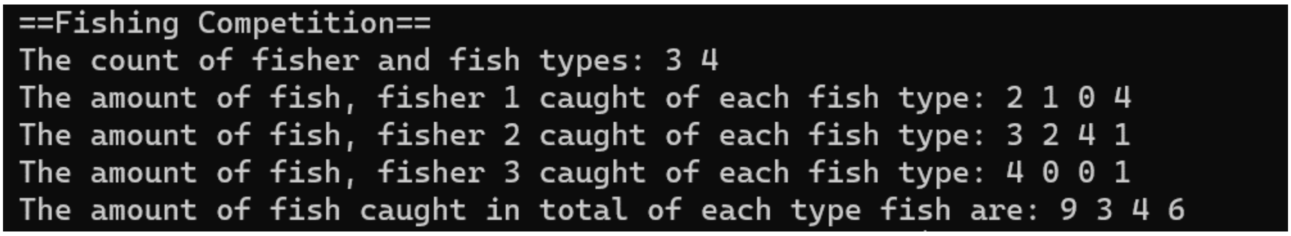
The program reads the input data from the keyboard in the following order:

#	Data	Explanation
1.	*M*	The count of fishers (1≤M≤100).
2.	*N*	The count of fish types (1≤N≤100).
3.	$a_{i,j}$	The number of fish a fisher caught of each fish type ($1 \le a_{i,j} \le 1000$).

Program output

The program writes out how many fish were caught in total of each type in the input order of fish types.

Sample input and output

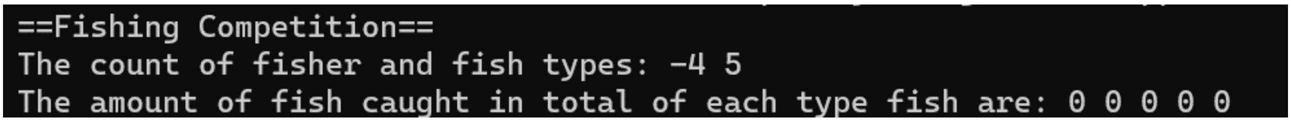


```
==Fishing Competition==
The count of fisher and fish types: 3 4
The amount of fish, fisher 1 caught of each fish type: 2 1 0 4
The amount of fish, fisher 2 caught of each fish type: 3 2 4 1
The amount of fish, fisher 3 caught of each fish type: 4 0 0 1
The amount of fish caught in total of each type fish are: 9 3 4 6
```

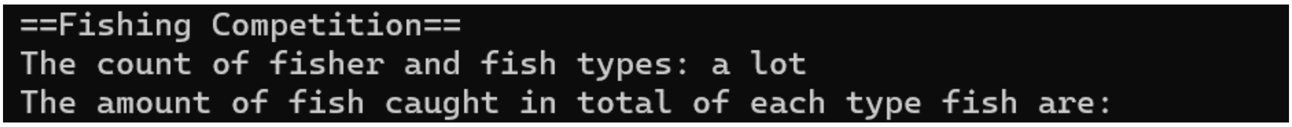
Possible errors

The input should be given according to the sample. If the number of measurements is not a whole number, it gives error. If the count of fishers is not in the range 1...100 for M or the count of fish types not in the range 1...100 for the N, it will cause a problem. If one of the measurements is not a number, or it is not in the range 1...1000 for number of fish a fisher caught of each fish type, it also will cause a problem. In the case of an error, the program displays an error message.

Sample of running in the case of invalid data:



```
==Fishing Competition==
The count of fisher and fish types: -4 5
The amount of fish caught in total of each type fish are: 0 0 0 0 0
```



```
==Fishing Competition==
The count of fisher and fish types: a lot
The amount of fish caught in total of each type fish are:
```

Developer documentation

Task

We have the result of a fishing competition in a matrix: M (i, j) means that fisher i caught M (i, j) pieces of fish type j.

Create a program that calculates how much the fishers caught of each type of fish.

Specification

Input: $M \in \mathbb{N}$, $N \in \mathbb{N}$, $a_{1...M,\ 1...N} \in \mathbb{N}^{MXN}$

Output: $sum_{1...N} \in \mathbb{N}^{N}$

Precondition: $1 <= M <= 100 \wedge 1 <= N <= 100$

$\forall i\ (1 <= i <= M) \wedge \forall j\ (1 <= i <= N) \wedge 1 <= a_{i,\ j} <= 1000$

Postcondition: $\forall i\ (1 <= i <= M) \wedge \forall j\ (1 <= i <= N)$

$$sum_{j} = \sum_{j=1}^{N} a_{i,\ j}$$

Developer environment

Huawei PC, an operating system capable of running exe files (e.g., Windows 11). mingw32-g++.exe C++compiler (v5.1), Code: Blocks (v17.12) developer tool.

Source code

All the sources can be found in the *B3* folder (after extraction). The folder structure used for development:

File	Explanation
B3\bin\Release\main.exe	Executable code
B3\obj\Release\main.o	Semi-compiled code
B3\main.cpp	C++ source code
B3\test1.txt	input test $file_1$
B3\test2.txt	input test $file_2$
B3\test3.txt	input test $file_3$
B3\test4.txt	input test $file_4$
B3\test5.txt	input test $file_5$
B3\Documentation.docx	documentation (this file)

Solution

Program parameters

Variables

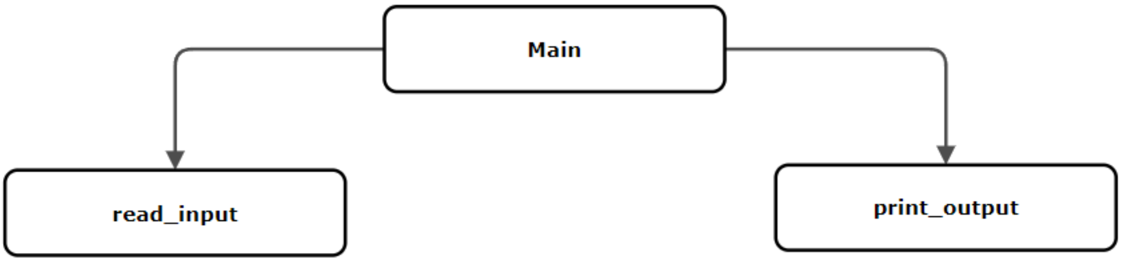
M: Integer
N: Integer
a: 2D Array (1...M, 1...N: Integer)
sum: Array (1...N: Integer)

The structure of the program

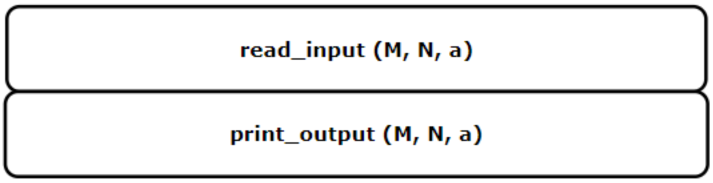
The modules used by the program, and their locations:

main.cpp– the program, in the source folder
bits/stdc++.h– – keyboard, console management etc., part of the C++ system

Structure of functions

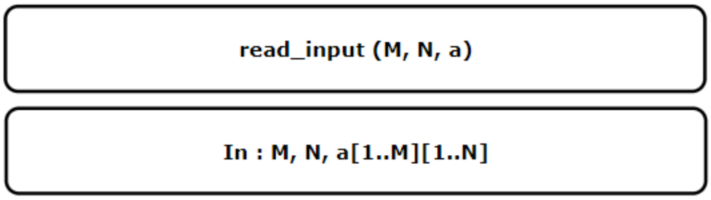


Main Program

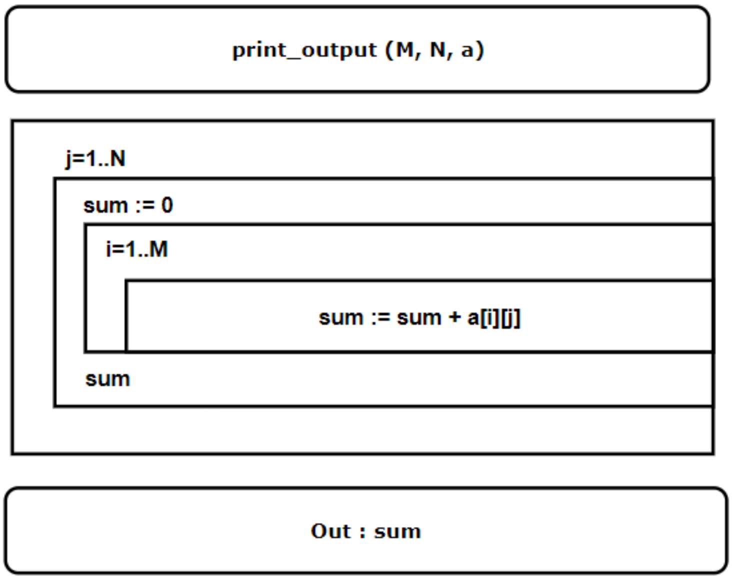


Sub Programs

read_inputs



print_output



The code

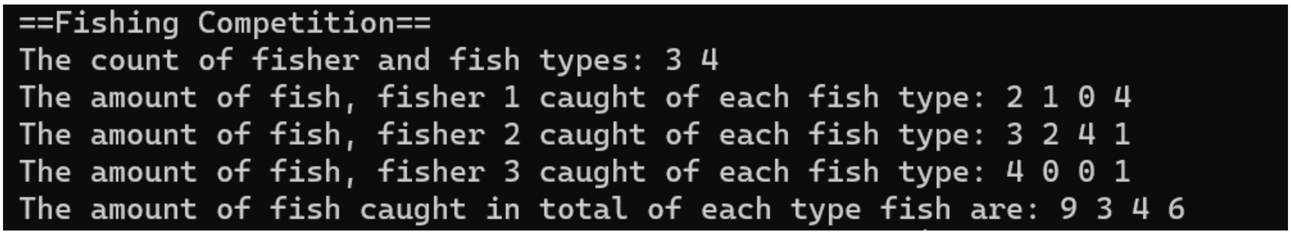
The content of the `main.cpp` file:

```
  /*Created by: Abrar Ahmed
  Neptun: OC7IP4
  E-mail: abrarahmedpei@gmail.com
  Task: „Big Project” ==Fishing Competition==*/
#include<bits/stdc++.h>
using namespace std;

void print_output(int M, int N, int a[100][100]){
    int sum;
    cout<<"The amount of fish caught in total of each type fish are: ";
    for(int j = 0;j < N; j++){
        sum = 0;
        for(int i = 0; i < M ; i++){
            sum = sum + a[i][j];
        }
        cout << sum <<" ";
    }
    cout << endl;
}

void read_input(int &M, int &N, int a[100][100])
{
    cout<<"==Fishing Competition=="<<endl;
    do{
        cout<<"The count of fisher and fish types: ";
        cin>>M>>N;
    }while(M<= 1 || M>=100 && N<=1 || N>=100);
    for(int i = 0; i < M; i++)
    {
        cout<<"The amount of fish, fisher "<<i+1<<" caught of each type fish: ";
        for(int j = 0; j < N; j++)
        {
            cin>>a[i][j];
        }
    }
}

int main()
{
    int a[100][100], M, N;

    read_input(M, N, a);
    print_output(M, N, a);

    return 0;
}
```

Testing

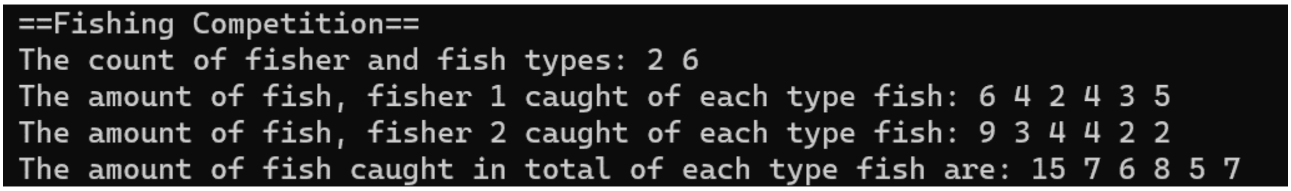
Valid test cases

1. *test case: test1.txt*



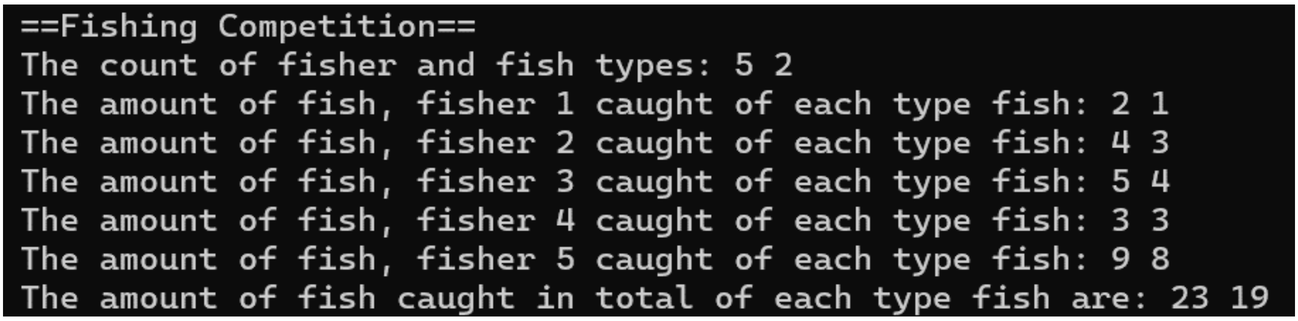
```
==Fishing Competition==
The count of fisher and fish types: 3 4
The amount of fish, fisher 1 caught of each fish type: 2 1 0 4
The amount of fish, fisher 2 caught of each fish type: 3 2 4 1
The amount of fish, fisher 3 caught of each fish type: 4 0 0 1
The amount of fish caught in total of each type fish are: 9 3 4 6
```

2. *test case: test2.txt*



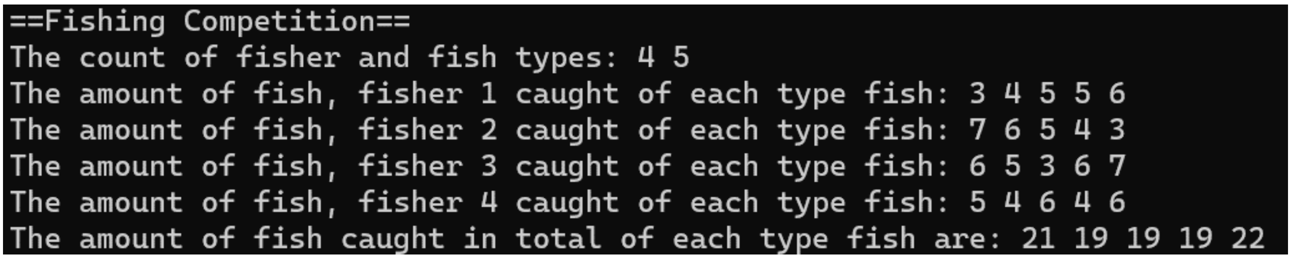
```
==Fishing Competition==
The count of fisher and fish types: 2 6
The amount of fish, fisher 1 caught of each type fish: 6 4 2 4 3 5
The amount of fish, fisher 2 caught of each type fish: 9 3 4 4 2 2
The amount of fish caught in total of each type fish are: 15 7 6 8 5 7
```

3. *test case: test3.txt*



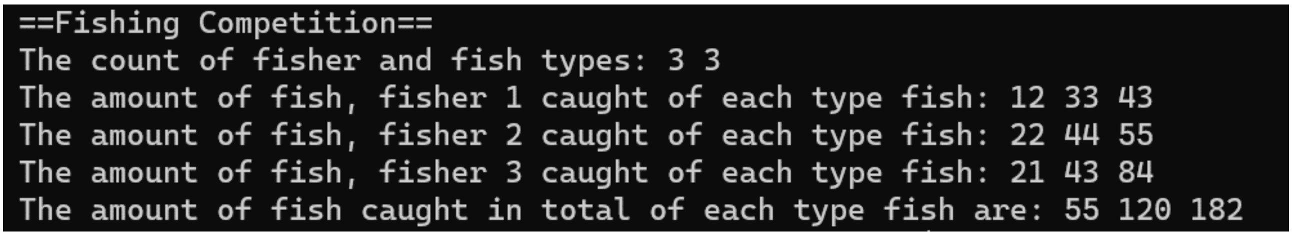
```
==Fishing Competition==
The count of fisher and fish types: 5 2
The amount of fish, fisher 1 caught of each type fish: 2 1
The amount of fish, fisher 2 caught of each type fish: 4 3
The amount of fish, fisher 3 caught of each type fish: 5 4
The amount of fish, fisher 4 caught of each type fish: 3 3
The amount of fish, fisher 5 caught of each type fish: 9 8
The amount of fish caught in total of each type fish are: 23 19
```

4. *test case: test4.txt*



```
==Fishing Competition==
The count of fisher and fish types: 4 5
The amount of fish, fisher 1 caught of each type fish: 3 4 5 5 6
The amount of fish, fisher 2 caught of each type fish: 7 6 5 4 3
The amount of fish, fisher 3 caught of each type fish: 6 5 3 6 7
The amount of fish, fisher 4 caught of each type fish: 5 4 6 4 6
The amount of fish caught in total of each type fish are: 21 19 19 19 22
```

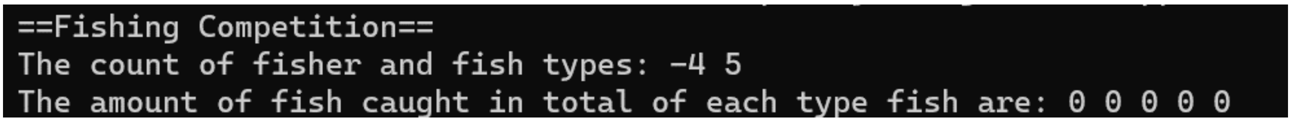
5. *test case: test5.txt*



```
==Fishing Competition==
The count of fisher and fish types: 3 3
The amount of fish, fisher 1 caught of each type fish: 12 33 43
The amount of fish, fisher 2 caught of each type fish: 22 44 55
The amount of fish, fisher 3 caught of each type fish: 21 43 84
The amount of fish caught in total of each type fish are: 55 120 182
```

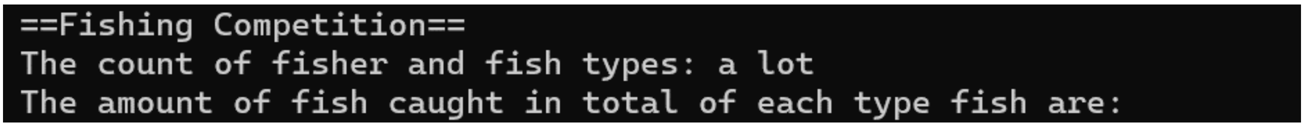
Invalid test cases

6. *test case*



```
==Fishing Competition==
The count of fisher and fish types: -4 5
The amount of fish caught in total of each type fish are: 0 0 0 0 0
```

7. *test case*



```
==Fishing Competition==
The count of fisher and fish types: a lot
The amount of fish caught in total of each type fish are:
```

Further development options

1. Reading from a file
2. Detection of wrong input file
3. Writing out the Location and ID of the errored file
4. Capability of running multiple times one after another