“PROGRAMMING” Big Project

Made by: Hussain Izhar
Neptun code: L9EIKJ
Email: izhar1022@gmail.com

Course code: IP-18fPROGEG
Teacher’s name: Menyhárt László Gábor

7TH July, 2020

CONTENTS

User DOCUMENTATION

Task

We have the weather forecast for the next N days for M settlements.
Write a program that gives the warmest settlement, which means that it is the settlement that has the longest period in which temperatures are above a given K temperature

Runtime environment

An HP PC that can run exe files, 32-bit operating system (e.g. Windows 10).
No mouse needed.

Usage

Starting the program

The program can be found in the archived file by the name `C3Assignment\bin\Release\a.exe`. You can start the program by clicking `c3assignement.exe` file.

Program Input

Data	Explanation
N	The count of days ($1 \leq N \leq 1000$)
M	The count of settlements (1<=M<=1000)
K	A temperature limit ($20 \leq K \leq 50$)
$Data_{i,j}$	i^{th} settlement for the j^{th} day (0<=$Data_{i,j}$ <=50)
…	
$Data_{M,N}$	N^{th} settlement for the M^{th} day (0 <= $Data_{N,M}$ <=50)

Program output

The program writes out the index of the settlement which has the longest period in which temperatures are above a given K temperature. If there is more than one solution, the output should be the smallest index. If there is no solution the output should be -1.

Sample input and output

Inputs:

```
N: 3
M: 3
K: 30
Settlement: 1
    1. Day's Temperature: 20
    2. Day's Temperature: 22
    3. Day's Temperature: 31
Settlement: 2
    1. Day's Temperature: 31
    2. Day's Temperature: 31
    3. Day's Temperature: 31
Settlement: 3
    1. Day's Temperature: 29
    2. Day's Temperature: 31
    3. Day's Temperature: 31
```

Outputs:

```
 Outputs:
The index of the settlement which has the longest period: 2
```

Possible errors

The input should be given according to the sample. If the Count of days and count of the settlements are not an integer number and the not in between 1 and 1000, it will cause an error. If the temperature limit is not an integer number and between 20 and 50, it will also cause an error. The expected temperature for the i^{th} settlement and for the j^{th} day are not integer numbers and not in between 0 and 50, it will cause and error. In the case of an error, the program displays an error message and asks for the repetition of the input.

Sample of running in the case of invalid data:

```
N: Three
Error! give me a number between 1 and 1000: -1
Error! give me a number between 1 and 1000: 10001
Error! give me a number between 1 and 1000: 3
M: Two
Error! give me a number between 1 and 1000: -1
Error! give me a number between 1 and 1000: 10001
Error! give me a number between 1 and 1000: 2
K: Twenty one
Error! give me a number between 20 and 50: 19
Error! give me a number between 20 and 50: 51
Error! give me a number between 20 and 50: 21
Settlement: 1
```

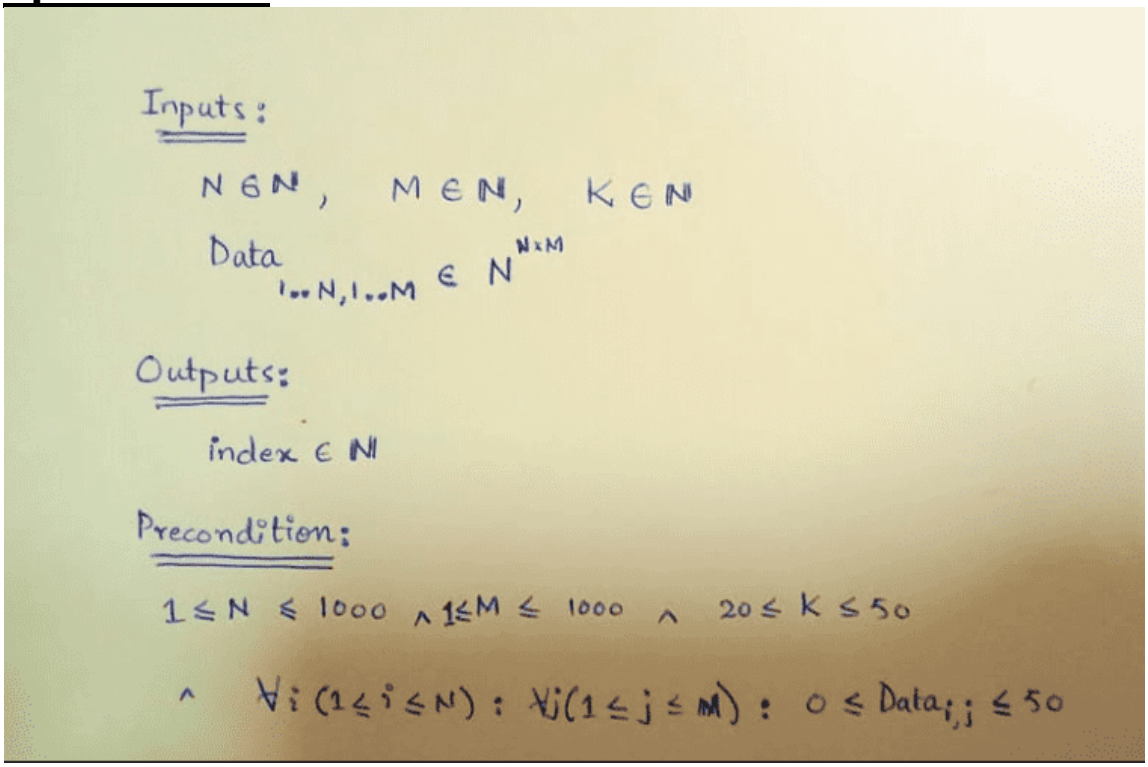
```
1. Day’s Temperature: Twenty
Error! give me a number between 0 and 50: -1
Error! give me a number between 0 and 50: 51
Error! give me a number between 0 and 50: 21
2. Day’s Temperature
```

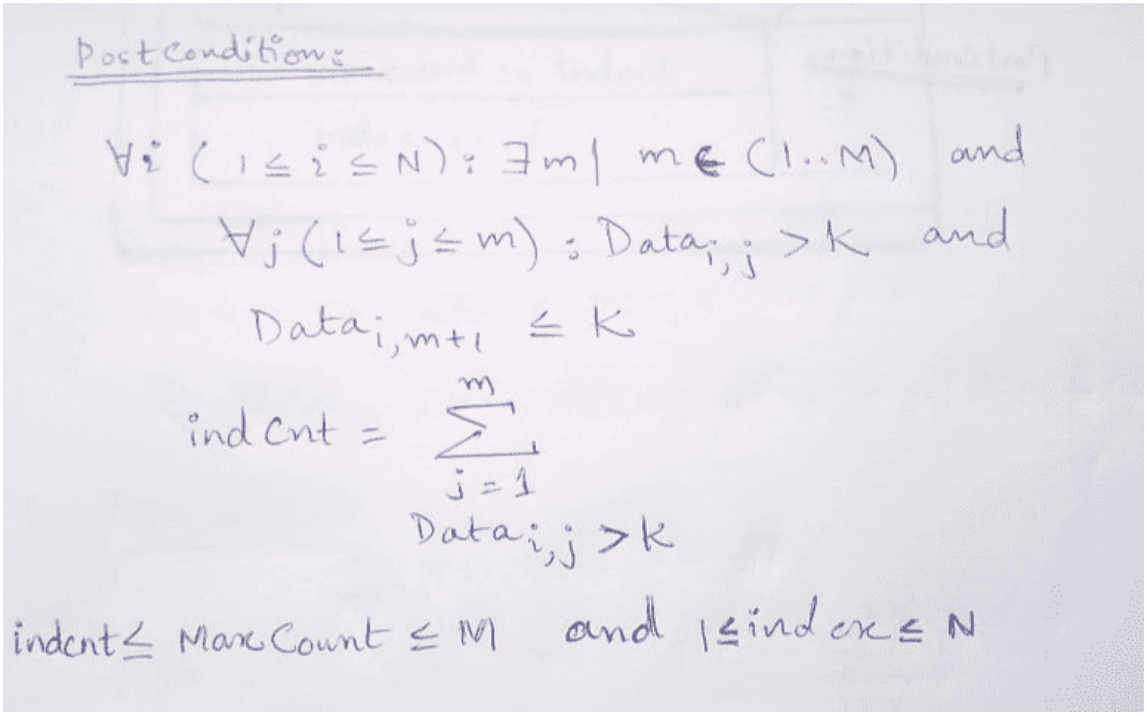
Developer documentation

Task

We have the weather forecast for the next N days for M settlements.
Write a program that gives the warmest settlement, which means that it is the settlement that has the longest period in which temperatures are above a given K temperature

Specification





Developer environment

HP PC, an operating system capable of running exe files (e.g. Windows 10). `mingw32-g++.exe` C++ compiler (v5.1), Code :: Blocks (v17.12) developer tool.

Source code

All the sources can be found in the `c3assingnment` folder (after extraction). The folder structure used for development:

File Name	Explanation
`C3Assignment\bin\Release\a.exe.`	Executable File
`C3Assignment\bin\Release\main.o`	Object File/Semi Compiled
`C3Assignment\main.cpp`	C++ source Code
`C3Assignment\TestCase1.txt`	Input File$_1$
`C3Assignment\TestCase2.txt`	Input File$_2$
`C3Assignment\TestCase3.txt`	Input File$_3$
`C3Assignment\TestCase4.txt`	Input File$_4$
`C3Assignment\TestCase5.txt`	Input File$_5$
`C3Assignment\TestCase6.txt`	Input File$_6$
`C3Assignment\TestCase7.txt`	Input File$_7$
`C3Assignment\TestCase8.txt`	Input File$_8$
`C3Assignment\Documentation`	Documentation (This file)

Solution

Program parameters

Constants:

```
nMax: Integer (1000)
nMin: Integer (1)
Tmin: Integer (0)
Kmin: Integer (20)
Kmax: Integer (50)
```

Variables

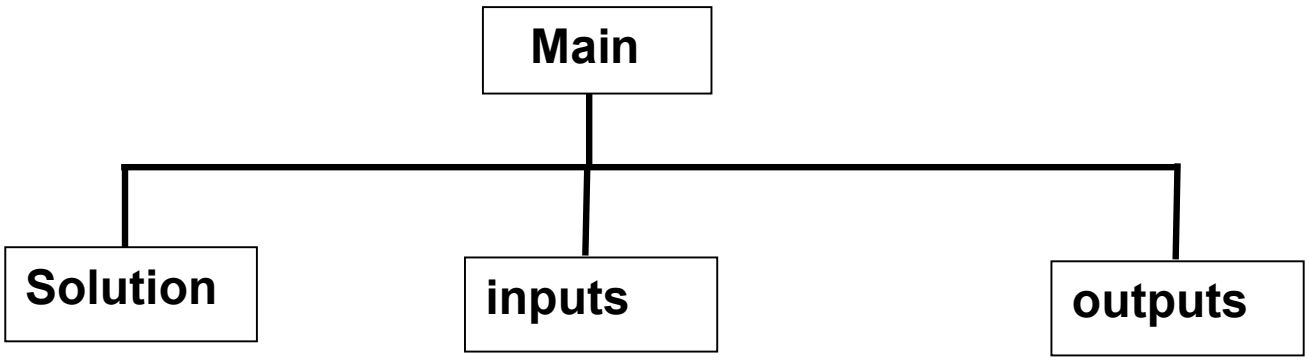
```
N       : Integer
M       : Integer
K       : Integer
Data    : 2D Array (1..N+1,1..M+1 : Integer)
N       : Integer
Index   : Integer
indCnt  : Integer

MaxCnt  : Integer
```

The structure of the program

The modules used by the program, and their locations:

`main.cpp` the program, in the source folder
`iostream` keyboard and console management, part of the C++ system

Structure of FUNCTIONS



Main Program

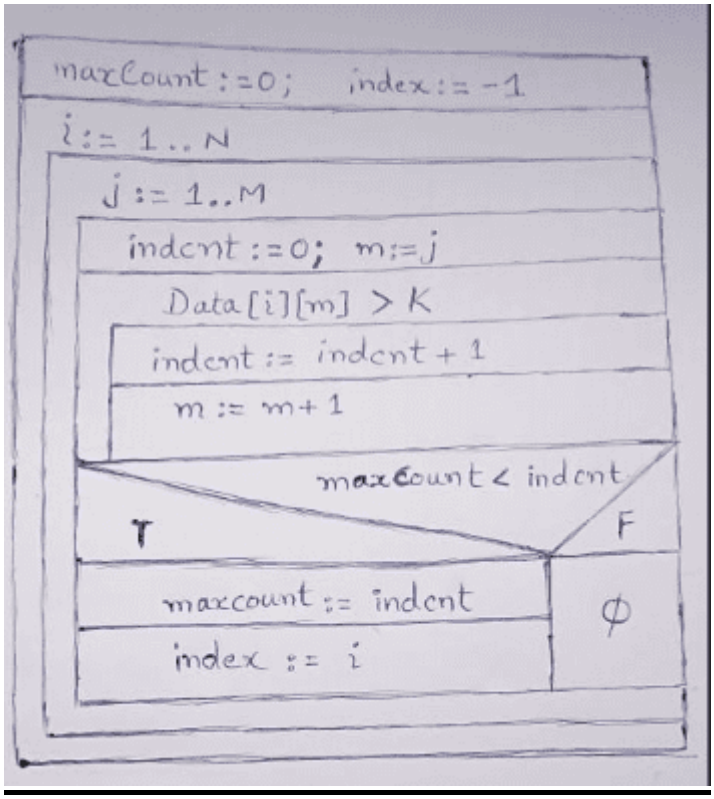
```
intputs(N,M,K,Data[][])
solution(N,M,K,indCnt, MaxCount, index, Data[][])
Outputs(index)
```

Sub-Programs:
inputs:

```
Intputs(N,M,K)
In: N,M,K,Data[1..N][1..M]
```

Solution:

```
solution(N,M,K,indCnt, MaxCount, index, Data[][])
```



outputs:

```
Outputs(index)
```

```
Out: index
```

The code

The content of the `main.cpp` file:

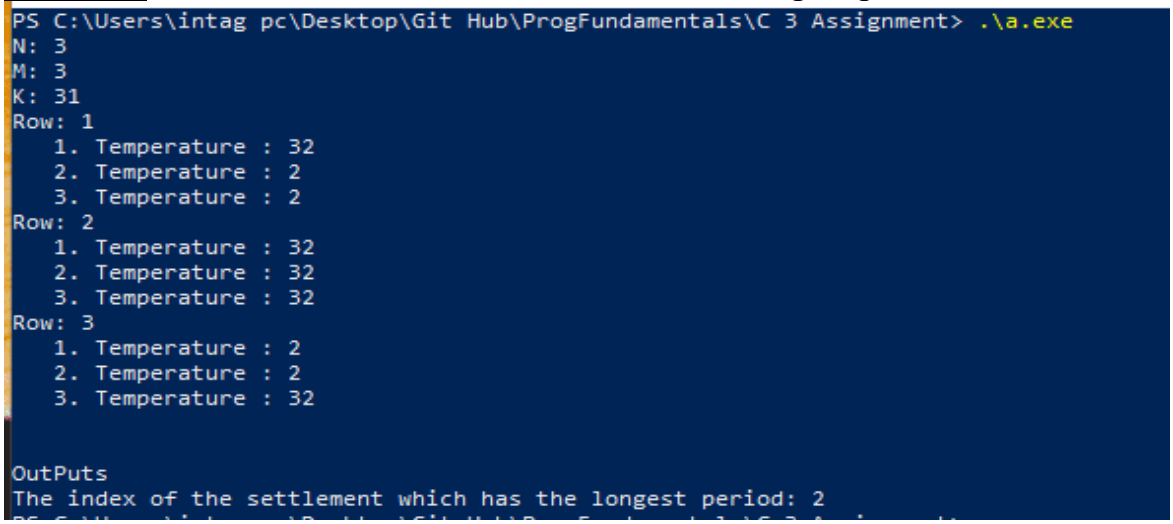
/* Created by: Hussain Izhar; Neptun Code: L9EIKJ E-mail: izhar1022@gmail.com; Task: „Big Project" – Longest period settlement*/

```
#include <iostream>

using namespace std;
const int nMax = 1000;
const int nMin = 1;
const int Kmax = 50;
const int Kmin = 20;
const int Tmin = 0;

//Helper Functions Declarations and definitions

void inputs (int &N, int &M, int &K)
{
	clog << "N: ";
	while(!(cin >> N) || N > nMax || N < nMin)
	{
		cout << "Error! give me a number between " << nMin << " and " << nMax
<< " : ";
		cin.clear();
		cin.ignore(10000,'\n');
	}
	clog << "M: ";
	while(!(cin >> M ) || M > nMax || M < nMin)
	{
		clog << "Error! give me a number between " << nMin << " and " << nMax
<< " : ";
		cin.clear();
		cin.ignore(10000,'\n');
	}
	clog << "K: ";
    while(!(cin >> K ) || K > Kmax || K < Kmin)
	{
		clog << "Error! give me a number between " << Kmin << " and " << Kmax
<< " : ";
		cin.clear();
		cin.ignore(10000,'\n');
	}
}


void Solution(int &N, int &M, int &K, int &indCnt, int &MaxCount, int &index, int
Data[])
{
	for (int i = 1; i <= N; i++)
	{
			clog << i << ". Settlement: " << endl;
		for (int j = 1; j <= M; j++)
		{
				clog << "   " << j << ". Day's Temperature : ";
				while(!(cin >> Data[j]) || Data[j] < Tmin || Data[j] >
Kmax)
				{
					clog << "   Error! give me a number between "<<
Tmin  << " and " << Kmax << " : ";
					cin.clear();
					cin.ignore(1000,'\n');
				}
			}
		for (int k = 1; k<=M; k++)
```

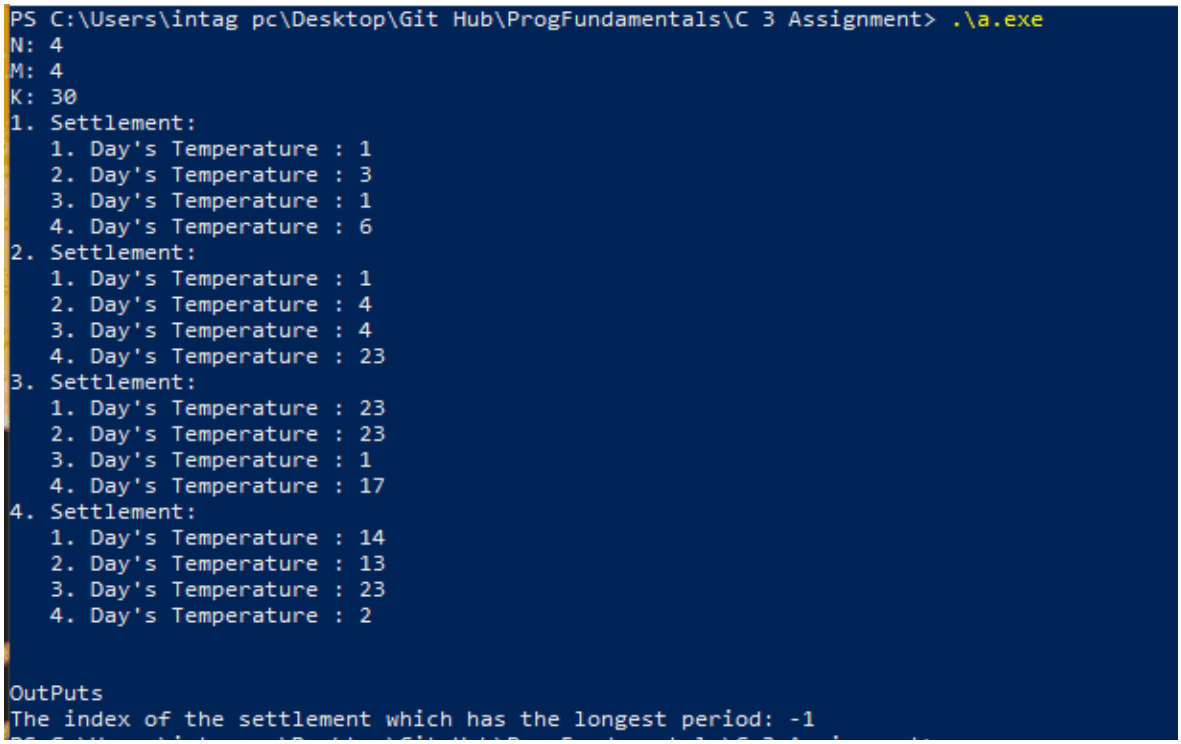
```
		{
			indCnt = 0;
			int m = k;
			while (Data[m] > K)
			{
			indCnt++;
			m++;
			}
			if (MaxCount < indCnt)
			{
				MaxCount = indCnt;
			index = i;
			}

		}
	}
}


outputs(int &index)
{
	clog << endl;
	clog << "OutPuts" << endl;
    cout << "The index of the settlement which has the longest period: " << index <<
endl;
}



int main()
{
	//Declarations
    int N, M, K;
	int indCnt=0;
    int MaxCount = 0;
    int index = -1;

	//Inputs
	inputs(N, M, K);
    int n = ((N+1) * (M+1));
	int Data[n];

	//algorithm implemention
	Solution(N, M, K, indCnt, MaxCount, index, &Data[n]);

	//Outputs
	outputs(index);
    return 0;
}
```

TESTING:

Case 01: When there exists a settlement with a longest period



Case 02: When there exists no settlement with a longest period



Case 03: When there exists two settlements settlement with a longest period

```
PS C:\Users\intag pc\Desktop\Git Hub\ProgFundamentals\C 3 Assignment> .\a.exe
N: 4
M: 4
K: 31
1. Settlement:
   1. Day's Temperature : 31
   2. Day's Temperature : 2
   3. Day's Temperature : 2
   4. Day's Temperature : 31
2. Settlement:
   1. Day's Temperature : 32
   2. Day's Temperature : 32
   3. Day's Temperature : 32
   4. Day's Temperature : 1
3. Settlement:
   1. Day's Temperature : 32
   2. Day's Temperature : 32
   3. Day's Temperature : 32
   4. Day's Temperature : 1
4. Settlement:
   1. Day's Temperature : 32
   2. Day's Temperature : 32
   3. Day's Temperature : 1
   4. Day's Temperature : 1


OutPuts
The index of the settlement which has the longest period: 2
```

Case 04: In case the user gives string instead of numbers

```
PS C:\Users\intag pc\Desktop\Git Hub\ProgFundamentals\C 3 Assignment> .\a.exe
N: one
Error! give me a number between 1 and 1000 : 1
M: two
Error! give me a number between 1 and 1000 : 2
K: thirty
Error! give me a number between 20 and 50 : 30
1. Settlement:
   1. Day's Temperature : twenty one
   Error! give me a number between 0 and 50 : 21
   2. Day's Temperature : ansdjlafkjsdl
   Error! give me a number between 0 and 50 : 21


OutPuts
The index of the settlement which has the longest period: -1
```

Case 05: In case the user gives numbers out of range (specified in pre-condition)

```
PS C:\Users\intag pc\Desktop\Git Hub\ProgFundamentals\C 3 Assignment>
N: -1
Error! give me a number between 1 and 1000 : 10001
Error! give me a number between 1 and 1000 : 2
M: -1
Error! give me a number between 1 and 1000 : 10001
Error! give me a number between 1 and 1000 : 2
K: 19
Error! give me a number between 20 and 50 : 51
Error! give me a number between 20 and 50 : 49
1. Settlement:
   1. Day's Temperature : -1
   Error! give me a number between 0 and 50 : 51
   Error! give me a number between 0 and 50 : 49
   2. Day's Temperature : -1
   Error! give me a number between 0 and 50 : 51
   Error! give me a number between 0 and 50 : 49
2. Settlement:
   1. Day's Temperature : -1
   Error! give me a number between 0 and 50 : 51
   Error! give me a number between 0 and 50 : 1
   2. Day's Temperature : -1
   Error! give me a number between 0 and 50 : 51
   Error! give me a number between 0 and 50 : 1


OutPuts
The index of the settlement which has the longest period: -1
```

Case 06: Random Case: every entry = 0

```
PS C:\Users\intag pc\Desktop\Git Hub\ProgFundamentals\C 3 Assignment> .\a.exe
N: 3
M: 3
K: 30
1. Settlement:
   1. Day's Temperature : 0
   2. Day's Temperature : 0
   3. Day's Temperature : 0
2. Settlement:
   1. Day's Temperature : 0
   2. Day's Temperature : 0
   3. Day's Temperature : 0
3. Settlement:
   1. Day's Temperature : 0
   2. Day's Temperature : 0
   3. Day's Temperature : 0


OutPuts
The index of the settlement which has the longest period: -1
```

Further Development:

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.