„Programming”
Big Project

Made by: Thomas Tiny

Neptun code: A1B2C3

E-mail: thomastiny@example.com

Neptun code: G31R6T

name: lizhipeng

Content

- # User documentation

Task

- Christmas trees are sold on the Christmas market.
- Write a program that gives the cheapest tree among the trees that cost more than K HUF.

- we have 6(N) count of trees price, and have a 5000(K) is mid price.
- find the count of trees price is smallest(5000<price[the indexprice]<10000),and the count number(How many such numbers)Cnt>0.

Create a program that gives two islands that are the closest to each other among islands. If there are no such islands, the result should be 0.

• Runtime environment

An IBM PC that is capable of running exe files, 32-bit operating system (eg. Windows 7). No mouse needed..

• Usage

• Starting the program

The program can be found in the archived file by the name `A1B2C3\bin\Release\A1B2C3.exe`. You can start the program by clicking the `A1B2C3.exe` file.

• Program input

The frst line of the standard input contains the count of trees (1≤N≤100) and a price ((1<=K<=10 000). The next N lines each contain the price of a tress (1<=T<=10 000).

#	Data	Explanation
1.	*N*	The count of trees(1≤N≤100).
2.	$price_1$	The first tree (0≤$price_1$≤10000).
3.	$price_2$	The second tree (0≤$price_2$≤10000).
...	…	
N+1.	$price_N$	The N^{th}height (0≤$price_N$≤10000).

• Program output

The frst line of the standard output should contain the index and price of the cheapest tree among the trees that cost more than K HUF. If there is more than one solution, the output should be the one with the smallest index. If there are no tree that are more

ex -

pensive than K HUF, then the output should be -1

- ## Sample input and output

```
6 5000
2500
5300
1900
2400
8800
5300
2 5300

Process returned 0 (0x0)   execution time : 40.152 s
Press any key to continue.
```

- ## Possible errors

The input should be given according to the sample. If the number of measurements is not a whole number, or it is not in the range 1..100, it will cause a problem. If one of the measurements is not a number, or it is not in the range 0..10000, it also will cause a problem. In the case of an error, the program displays an error message, or asks for the repetition of the input.

- ### *Sample of running in the case of invalid data:*

```
6 100000
1200
522
1444
5220
4111
2333
-1

Process returned 0 (0x0)   execution time : 26.549 s
Press any key to continue.
```

- **Developer documentation**

- **Task B3:**

- Cheapest expensive Christmas tree

- Christmas trees are sold on the Christmas market.

- Write a program that gives the cheapest tree among the trees that cost more than K HUF.

- Input

- The frst line of the standard input contains the count of trees (1≤N≤100) and a

- price (1≤K≤10 000). The next N lines each contain the price of a tress (1≤T≤10 000).

- Output

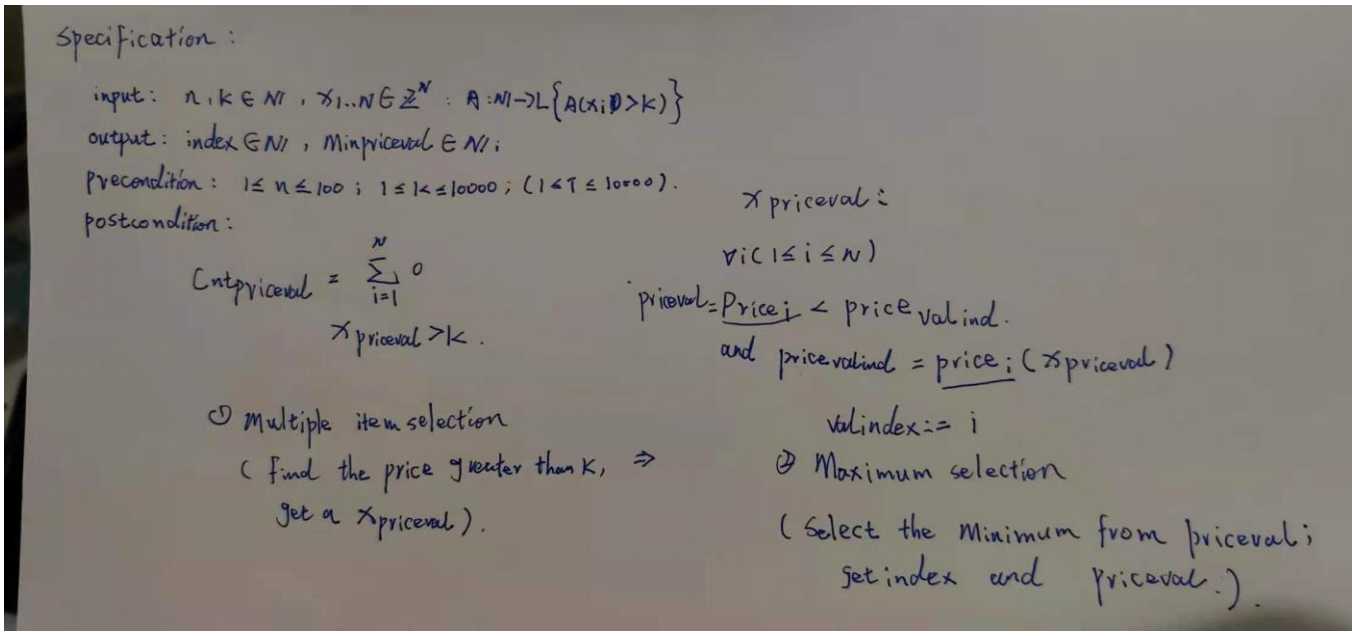
- The frst line of the standard output should contain the index and price of the cheapest tree among the trees that cost more than K HUF. If there is more than one solution,

- the output should be the one with the smallest index. If there are no tree that are more ex -

- pensive than K HUF, then the output should be -1.

• Specification

.



Definitions:

Comment: If there are less than K prices (Exists=False case) the program will write out a -1, and not the logical value (as it was required by the task).

• Developer environment

IBM PC, an operating system capable of running exe files (eg. Windows 7). mingw32-g++.exe c++ compiler (v4.7), Code::Blocks (v13.12) developer tool.

• Source code

All the sources can be found in the `A1B2C3` folder (after extraction). The folder structure used for development: main1.cpp (main5.cpp **FUNCTION)**

File	Explanation
`A1B2C3\bin\Release\A1B2C3.exe`	Executable code
`A1B2C3\obj\Release\main.o`	Semi-compiled code
`A1B2C3\main.cpp`	C++ source code

A1B2C3\test1.txt	input test file$_1$
A1B2C3\test2.txt	input test file$_2$
A1B2C3\test3.txt	input test file$_3$
A1B2C3\test4.txt	input test file$_4$
A1B2C3\test5.txt	input test file$_5$
A1B2C3\doksi\A1B2C3.docx	documentation (this file)

- ## Solution

- ## Program parameters

- ## *Contants*

```
•        MaxN : Integer(10000)  [the max of trees]
•        MAXprice    : Integer(10000)  [the max price]
```

- Types

```
•        prices   = Array(1..MaxN:Integer)
TPrice   = Record(prices:Integer)
```

- ## *Variables*

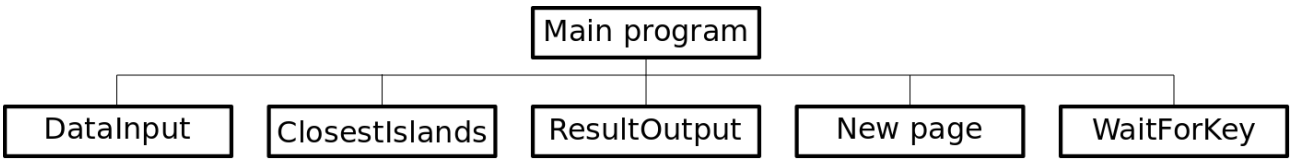
```
N,index        : Integer
cnt,prices     : Integer.
```

- ## The structure of the program

The modules used by the program, and their locations:

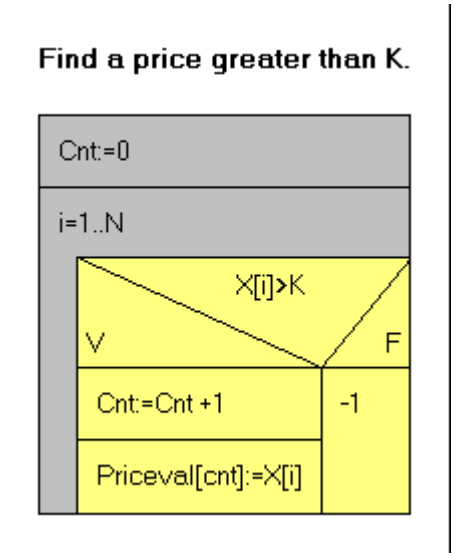
`main.cpp` – the program, in the source folder
`iostream` – keyboard and console management, part of the C++ system
`stdlib.h` – general routines, part of the C++ system

- ## Structure of functions

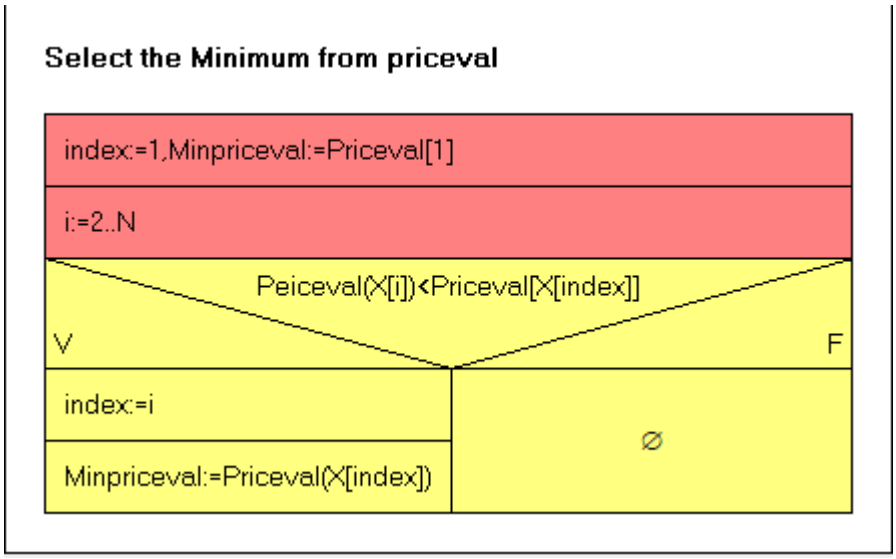
- 

The algorithm of the program

1.FIRST STEP:



2. NEXT



- ## The code

- // G31R6T lizhipeng code.

```
#include <iostream>
using namespace std;
int getCheapest(int *prices, int n, int k)

    int index = -1;
    for (int i = 0; i < n; i++)
    {
    if (prices[i] > k && (index == -1 || prices[i]< prices[index]))
        {
            index = i;
        }
    }
    return index;
}

int main()
{
    int n;
    int k;
    int prices[100];
    cin >> n >> k;
    for(int i = 0;i < n;i++){
        cin >> prices[i];
    }

    int index = getCheapest(prices, n, k);
    if (index < 0)
```

```
        {
            cout << -1 << endl;
        }
        else
        {
            cout << (index+1) << " " << prices[index] << endl;
        }
        return 0;
    }
```

why this is no color code.

Testing

Valid test cases

test case: in1.txt

Input – *the count of trees and price(5000)*
N=6 K=5000 prices[i]1=2500 prices[i]2=5300 prices[i]3=1900 prcise[i]4=2400 prices[i]5=8800 prices[i]6=5300
Output
Cnt=2 prices[index]

Further development options

- Data to be read from file
- Detection of wrong file input, writing out the location and ID# of error
- Capability to run multiple times after each other
- Visual representation of input data, and emphasizing the result islands with different colors