**„Winery Production”**

*Made by: Muhammad Eman Aftab* *Neptun code:* IJE4R1  
 *E-mail:* [*emanaftab2022@gmail.com*](mailto:emanaftab2022@gmail.com)

*Course code: 2* Teacher’s name: Vincze Dorottya

***10/1/2024***

Content

User documentation............................................................................................................................. 4

Task.................................................................................................................................................. 4

Usage................................................................................................................................................ 4

Starting the program.................................................................................................................... 4

Program input............................................................................................................................... 4

Program output............................................................................................................................. 4

Sample input and output.............................................................................................................. 5

Possible errors.............................................................................................................................. 5

Developer documentation.................................................................................................................... 6

Task.................................................................................................................................................. 6

Specification.................................................................................................................................... 6

Developer environment.................................................................................................................... 6

Source code...................................................................................................................................... 6

Solution............................................................................................................................................ 7

Program parameters..................................................................................................................... 7

The structure of the program........................................................................................................ 7

Structure of functions................................................................................................................... 7

The algorithm of the program...................................................................................................... 7

The code....................................................................................................................................... 8

Testing.............................................................................................................................................. 9

Valid test cases............................................................................................................................. 9

Invalid test cases........................................................................................................................ 10

Further development options.......................................................................................................... 11

**User documentation**

**Task**

Multiple wineries register the type and amount of wine sold, and the price the wine was sold at each year. Wineries and products can be listed multiple times.

Write a program that gives the name of the winery, the product, and the amount sold for the highest price by any of the wineries

**Usage**

**Runtime environment**

Hp PC, an operating system capable of running exe files (eg. Windows 7,10). Visual studio (2022) developer tool.

**Starting the program**

The program can be found in the archived file by the name WineryProduction\Program\bin\Release\net6.0\B\_3.exe. You can start the program by clicking the B\_3.exe file.

**Program input**

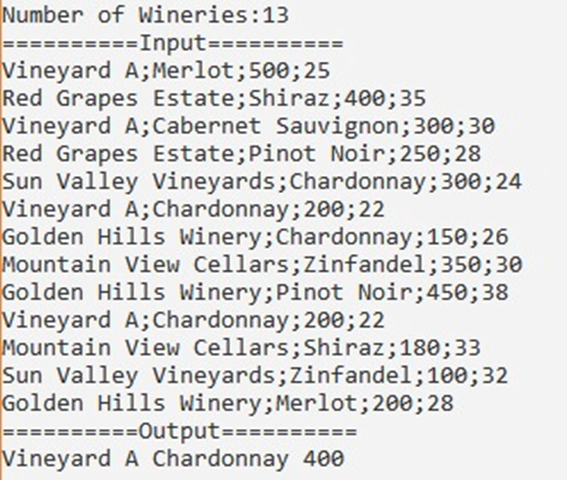
The program reads the input data from the keyboard in the following order:

| **#** | **Data** | **Explanation** |
| --- | --- | --- |
| **1.** | *N* | The count of years (1 ≤ N ≤ 100). |
| **2.** | *winery1.name, winery1.product,*  *winery1.amount,*  *winery1 . price* | The amount of winery1 (1 ≤ amount1 ≤ 10 000).  The price of winery1 (1 ≤ price1 ≤ 10 000). |
| **3.** | *winery2.name,*  *winery2.product,*  *winery2.amount*  *winery2. price* | The amount of winery2 (1 ≤ amount2 ≤ 10 000).  The price of winery2 (1 ≤ price2 ≤ 10 000) |
| **...** | *…* |  |
| **N** | *wineryN.name,*  *wineryN.product, wineryN.amount,*  *wineryN.price* | The amount of Nth winery1 (1 ≤ amountN ≤ 10 000).  The price of Nth winery1 (1 ≤ priceN ≤ 10 000) |
| **N+1** | *wineryN+1.name,*  *wineryN+1.product, wineryN+1.amount,*  *wineryN+1.price* | The amount of N+1 winery1 (1 ≤ amountN+1 ≤ 10 000).  The price of N+1 winery1 (1 ≤ priceN+1 ≤ 10 000) |

**Program output**

The first line of the standard output should contain the name of the winery, the product, and the amount sold for the highest price by any of the wineries. If there are multiple products sold for the same highest price give the first occurrence.

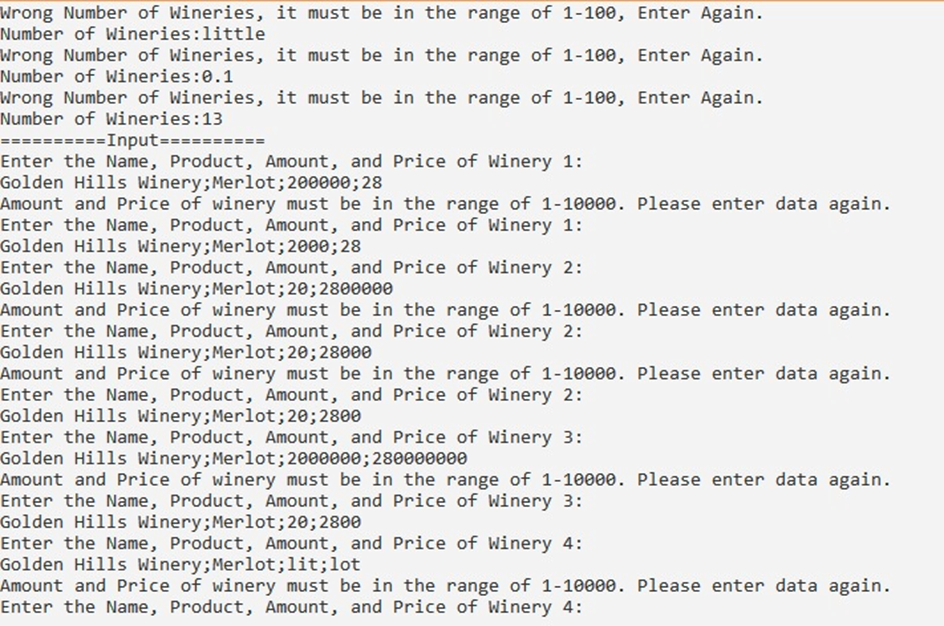
**Sample input and output**



**Possible errors**

The input should be given according to the sample. If the number of wineries is not a whole number, or it is not in the range 1..100, it will cause a problem. If one of the amount or price of winery is not a number, or it is not in the range 1..10,000, it will also 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:***

****

**Developer documentation**

**Task**

Multiple wineries register the type and amount of wine sold, and the price the wine was sold at each year. Wineries and products can be listed multiple times.

Write a program that gives the name of the winery, the product, and the amount sold for the highest price by any of the wineries.

**Specification**

**Input:** N ∈ **ℕ**, wines [1..N] ∈ **wineN**

wine = (name × type × amount × price),

name = T, type = T, amount = N, price = N

**Output:** result ∈ **outputN**,

output = (name × type × amount ),

name = T, type = T, amount = N

**Precondition:** 1 ≤ N ≤ 100

and ∀i (1 ≤ i ≤ N) : 1 ≤ wines*i*.amount ≤ 10,000

and ∀i (1 ≤ i ≤ N) : 1 ≤ wines*i*.price ≤ 10,000

**Postcondition:**

cnt

wines [ ] = ∃ (wines*i*.name = name And wines*i*.type = type)

i=1

cnt

wines[ ] = ∑ (wines*i*.amount + amount And wines*i*.price + price)

i=1

cnt

result = MAX (wines*i*.price)

i=1

**Comment**: If there are less than 2 wineries, the program will write out that winery, and not the logical value (as it was required by the task).

**Developer environment**

Hp PC, an operating system capable of running exe files (eg. Windows 7,10). Visual studio (2022) developer tool.

**Source code**

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

| **File** | **Explanation** |
| --- | --- |
| *WineryProduct\Program\bin\Release\net6.0\B\_3.exe* | Executable code |
| *WineryProduct\Program\obj\Release\net6.0\B\_3.pdb,B\_3.dll* | Semi-compiled code |
| *WineryProduct\Program\Program.cs* | C# source code |
| *WineryProduct\Program\B\_3.sln* | Visual Studio Solution |
|  |  |
| *WineryProduct\Program\TestCases\inp1.txt* | input test file1 |
| *WineryProduct\Program\TestCases\inp2.txt* | input test file2 |
| *WineryProduct\Program\TestCases\outp1.txt* | output test file1 |
| *WineryProduct\Program\TestCases\outp2.txt* | output test file2 |
|  |  |
| *WineryProduct\Documentation\WineryProduction.docx* | documentation (this file) |

**Solution**

**Program parameters**

Types

wine = **Record (name,type : String , amount, price : Integer)**

output = **Record (name,type : String , amount : Integer)**

***Variables***

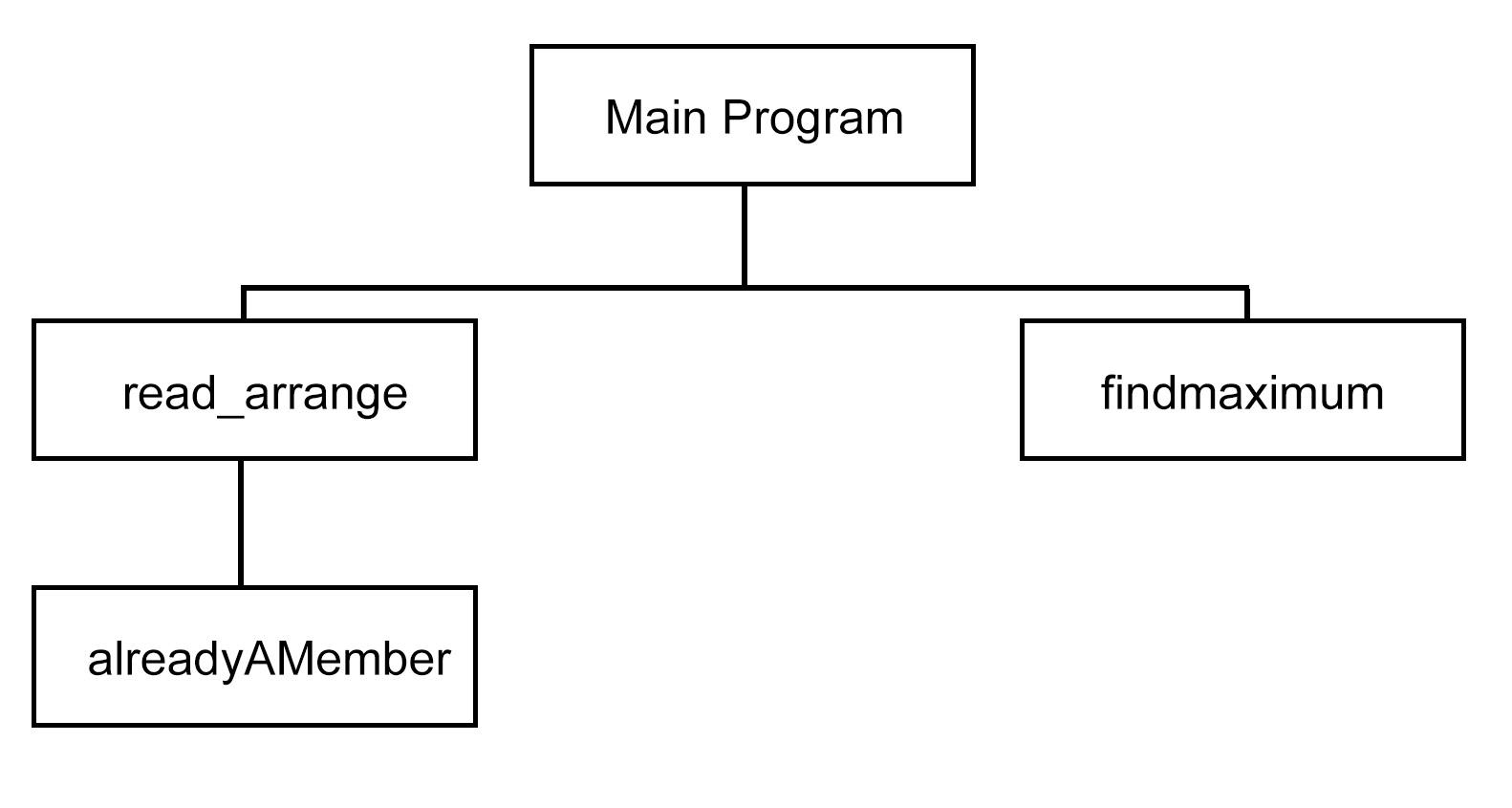
N : Integer  
wines : wine

result : output

**The structure of the program**

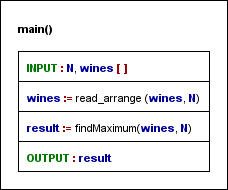
The modules used by the program, and their locations:

Program.cs – the program file, in the Program folder  
B\_3.csproj –C# Project File  
Program.sln – Visual Studio File

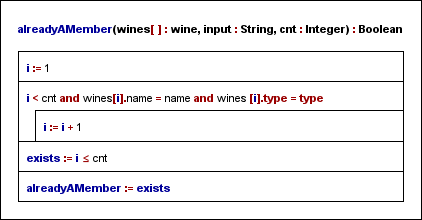
**Structure of functions**

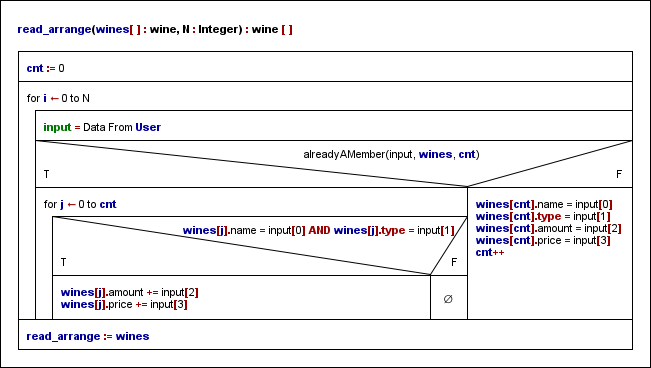
**The algorithm of the program**

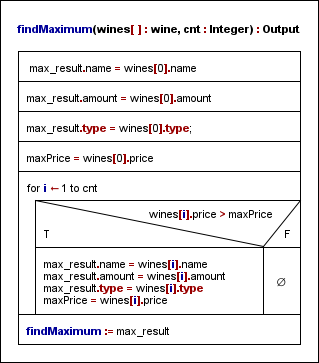
Main program:

****

Subprograms:







**The code**

The content of the program.cs file:

/\*

Created by: Muhammad Eman Aftab

Neptun: IJE4R1

E-mail: emanaftab2022@gmail.com

Task: „Winery Production”

\*/

using System;

namespace B\_3

{

internal class Program

{

public struct wine

{

public string name;

public string type;

public int amount;

public int price;

}

public struct Output

{

public string name;

public string type;

public int amount;

}

static void Main(string[] args)

{

bool good;

int N;

do

{

Console.Write("Number of Wineries:");

string n\_winery = Console.ReadLine();

good = int.TryParse(n\_winery, out N) && N >= 1 && N <= 100;

if (!good)

{

Console.WriteLine("Wrong Number of Wineries, it must be in the range of 1-100, Enter Again.");

}

}

while (!(good));

wine[] wines = new wine[N];

wines = read\_arrange(wines, N);

Output result = findMaximum(wines, N);

Console.WriteLine("==========Output==========");

Console.WriteLine($"{result.name} {result.type} {result.amount}");

//string output = result.name + " " + result.type + " " + result.amount;

Console.ReadLine();

}

public static bool alreadyAMember(string input, wine[] wines, int cnt)

{

int i = 0;

string name = input.Split(";")[0];

string type = input.Split(";")[1];

while (i < cnt && !(wines[i].name == name && wines[i].type == type) )

{

i = i + 1;

}

bool exists = i < cnt;

return exists;

}

public static Output findMaximum(wine[] wines, int cnt)

{

Output max\_result;

max\_result.name = wines[0].name;

max\_result.amount = wines[0].amount;

max\_result.type = wines[0].type;

int maxPrice = wines[0].price;

for (int i = 0; i < cnt; i++)

{

if (wines[i].price > maxPrice)

{

max\_result.name = wines[i].name;

max\_result.amount = wines[i].amount;

max\_result.type = wines[i].type;

maxPrice = wines[i].price;

}

}

return max\_result;

}

public static wine[] read\_arrange(wine[] wines, int N)

{

Console.WriteLine("==========Input==========");

int cnt = 0;

for (int i = 0; i < N; i++)

{

string input;

bool good;

do

{

Console.WriteLine("Enter the Name, Product, Amount, and Price of Winery {0}:",i+1);

input = Console.ReadLine();

string[] inputValues = input.Split(';');

int amount, price = 0;

good = int.TryParse(inputValues[2], out amount) && amount >= 1 && amount <= 10000 &&

int.TryParse(inputValues[3], out price) && price >= 1 && price <= 10000;

if (!good)

{

Console.WriteLine("Amount and Price of winery must be in the range of 1-10000. Please enter data again.");

}

else

{

wines[i].amount = amount;

wines[i].price = price;

}

}

while (!good);

if (alreadyAMember(input, wines, cnt))

{

for (int j = 0; j < cnt; j++)

{

if (wines[j].name == input.Split(";")[0] && wines[j].type == input.Split(";")[1])

{

int amount = Convert.ToInt32(input.Split(";")[2]);

int price = Convert.ToInt32(input.Split(";")[3]);

wines[j].amount += amount;

wines[j].price += price;

}

}

}

else

{

wines[cnt].name = input.Split(";")[0];

wines[cnt].type = input.Split(";")[1];

wines[cnt].amount = Convert.ToInt32(input.Split(";")[2]);

wines[cnt].price = Convert.ToInt32(input.Split(";")[3]);

cnt++;

}

}

return wines;

}

}

}

**Testing**

**Valid test cases**

***1.*** ***test case: in1.txt***

| **Input** |
| --- |
| Number of Wineries:13  Enter the Name, Product, Amount, and Price of Winery 1:  Vineyard A;Merlot;500;25  Enter the Name, Product, Amount, and Price of Winery 2:  Red Grapes Estate;Shiraz;400;35  Enter the Name, Product, Amount, and Price of Winery 3:  Vineyard A;Cabernet Sauvignon;300;30  Enter the Name, Product, Amount, and Price of Winery 4:  Red Grapes Estate;Pinot Noir;250;28  Enter the Name, Product, Amount, and Price of Winery 5:  Sun Valley Vineyards;Chardonnay;300;24  Enter the Name, Product, Amount, and Price of Winery 6:  Vineyard A;Chardonnay;200;22  Enter the Name, Product, Amount, and Price of Winery 7:  Golden Hills Winery;Chardonnay;150;26  Enter the Name, Product, Amount, and Price of Winery 8:  Mountain View Cellars;Zinfandel;350;30  Enter the Name, Product, Amount, and Price of Winery 9:  Golden Hills Winery;Pinot Noir;450;38  Enter the Name, Product, Amount, and Price of Winery 10:  Vineyard A;Chardonnay;200;22  Enter the Name, Product, Amount, and Price of Winery 11:  Mountain View Cellars;Shiraz;180;33  Enter the Name, Product, Amount, and Price of Winery 12:  Sun Valley Vineyards;Zinfandel;100;32  Enter the Name, Product, Amount, and Price of Winery 13:  Golden Hills Winery;Merlot;200;28 |
| **Output** |
| Vineyard A Chardonnay 400 |

***2.*** ***test case: in2.txt***

| **Input –** *starts with continent, there are at least 2 islands* |
| --- |
| Number of Wineries:15  Enter the Name, Product, Amount, and Price of Winery 1:  Vineyard A;Merlot;500;25  Enter the Name, Product, Amount, and Price of Winery 2:  Red Grapes Estate;Shiraz;400;35  Enter the Name, Product, Amount, and Price of Winery 3:  Vineyard A;Cabernet Sauvignon;300;30  Enter the Name, Product, Amount, and Price of Winery 4:  Red Grapes Estate;Pinot Noir;250;28  Enter the Name, Product, Amount, and Price of Winery 5:  Sun Valley Vineyards;Chardonnay;300;24  Enter the Name, Product, Amount, and Price of Winery 6:  Golden Hills Winery;Chardonnay;150;26  Enter the Name, Product, Amount, and Price of Winery 7:  Mountain View Cellars;Zinfandel;350;30  Enter the Name, Product, Amount, and Price of Winery 8:  Sun Valley Vineyards;Sauvignon Blanc;200;18  Enter the Name, Product, Amount, and Price of Winery 9:  Golden Hills Winery;Pinot Noir;450;38  Enter the Name, Product, Amount, and Price of Winery 10:  Vineyard A;Chardonnay;200;22  Enter the Name, Product, Amount, and Price of Winery 11:  Mountain View Cellars;Shiraz;180;33  Enter the Name, Product, Amount, and Price of Winery 12:  Red Grapes Estate;Cabernet Sauvignon;150;40  Enter the Name, Product, Amount, and Price of Winery 13:  Sun Valley Vineyards;Zinfandel;100;32  Enter the Name, Product, Amount, and Price of Winery 14:  Golden Hills Winery;Merlot;200;28  Enter the Name, Product, Amount, and Price of Winery 15:  Mountain View Cellars;Sauvignon Blanc;120;20 |
| **Output** |
| Red Grapes Estate Cabernet Sauvignon 150 |

**Invalid test cases**

***3.*** ***test case***

| **Input –** *wrong number of wineries* |
| --- |
| Number of Wineries:little |
| **Output** |
| Asking again:  N = |

***4.*** ***test case***

| **Intput –** wrong *height* |
| --- |
| Number of Wineries:13  Enter the Name, Product, Amount, and Price of Winery 1:  Vineyard A;Merlot;little;25 |
| **Output** |
| Asking again:  Amount and Price of winery must be in the range of 1-10000. Please enter data again.  Enter the Name, Product, Amount, and Price of Winery 1: |

***5.*** ***test case***

| **Input –** *wrong number of wineries* |
| --- |
| Number of Wineries:13  Enter the Name, Product, Amount, and Price of Winery 1:  Vineyard A;Merlot;500;250000 |
| **Output** |
| Asking again: Amount and Price of winery must be in the range of 1-10000. Please enter data again.  Enter the Name, Product, Amount, and Price of Winery 1 |

…

***6. test case***

…

**Further development options**

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