



BÁO CÁO TIẾN ĐỘ

Nhóm P1_01

- 1, Hán Đức **Toàn**
- 2, Phạm Xuân **Hồng**
- 3, Phạm Khắc Hoài **Nam**

MỤC LỤC

I.	KHÁI QUÁT	3
1.	Thư viện sử dụng:.....	3
2.	Lưu ý kiểm thử	3
II.	THUẬT TOÁN.....	3
1.	Thuật toán sử dụng	3
2.	Thuật toán áp dụng cho trường hợp bộ 2 màu.....	4
3.	Thuật toán áp dụng cho trường hợp bộ 3 màu.....	5
4.	Thuật toán áp dụng cho trường hợp bộ 4 màu.....	6
5.	Thuật toán áp dụng cho trường hợp bộ 5 màu.....	7
III.	MÃ NGUỒN.....	8
A.	Cấu trúc MVC	8
1.	Tổng quan.	8
2.	Các file thuộc Model, nội dung và nhiệm vụ.	8
3.	Các file thuộc View, nội dung và nhiệm vụ.	9
4.	Các file thuộc Controller, nội dung và nhiệm vụ.	10
B.	Tương tác MVC	11
IV.	ĐỊNH DẠNG DỮ LIỆU	12
A.	Dữ liệu đầu vào	12
1.	Định dạng	12
2.	Lọc dữ liệu đầu vào	12
B.	Dữ liệu đầu ra.....	12
1.	Định dạng	12
2.	Lưu trữ.....	12
C.	TỐI ƯU	13
1.	Sắp xếp dữ liệu đầu vào để xử lý.....	13
2.	Đặt điều kiện để dừng trong quá xử lý	13
3.	So sánh hiệu quả	13

I. KHÁI QUÁT

1. Thư viện sử dụng:

thietlaphesoModel.cs: System;

Welcome.xaml: System; System.Windows; System.Windows.Controls;

thietlapHeSo.xaml: System; System.Windows; System.Windows.Controls; Microsoft.Win32;

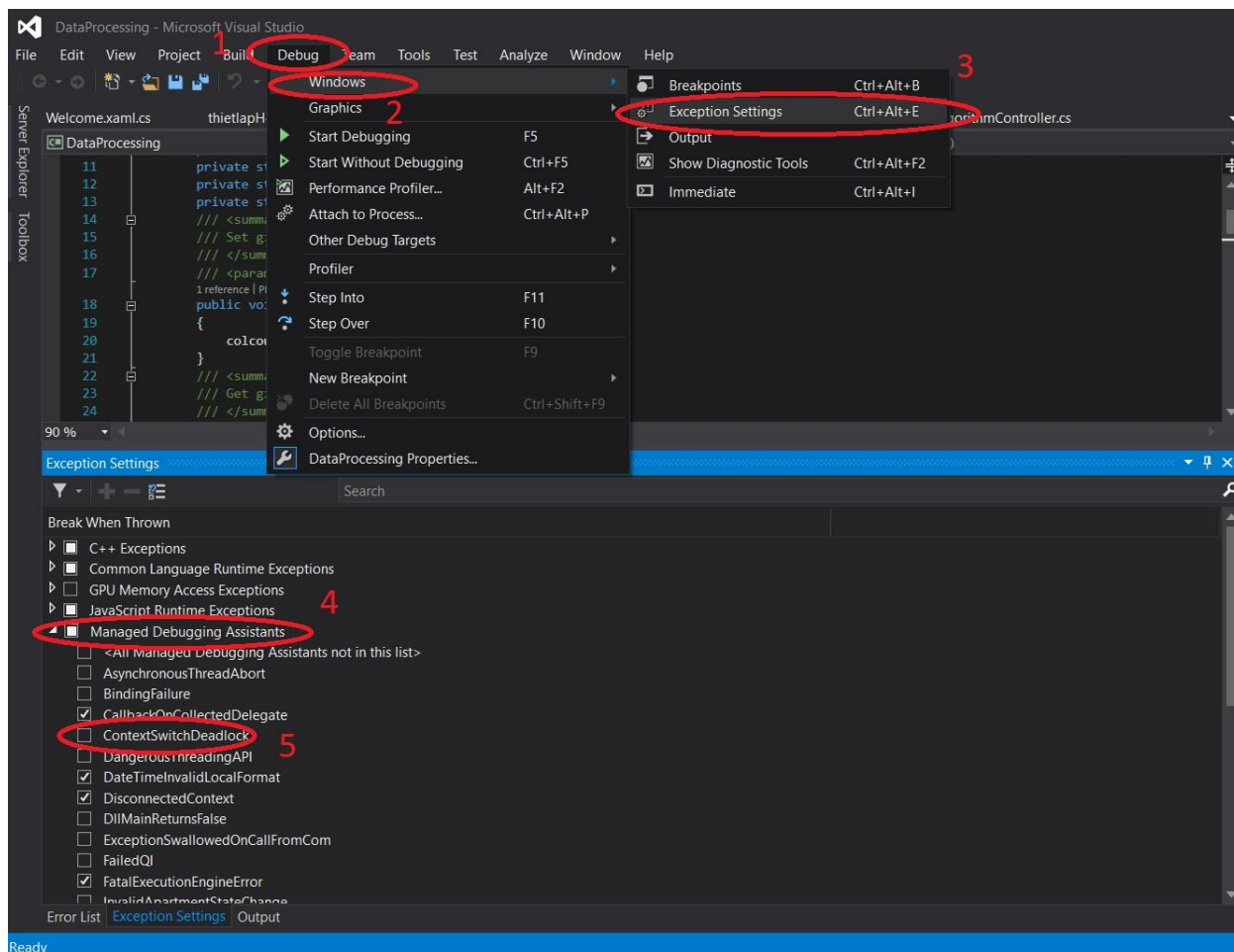
FindingStatus.xaml: System.Windows.Controls;

AlgorithmController: System; System.Collections.Generic; System.Linq;

ExcelController: System; System.Windows; Microsoft.Office.Interop.Excel;

2. Lưu ý kiểm thử

Tính năng **ContextSwitchDeadlock** của phần mềm Visual Studio có tác động ngừng chạy Debug khi quá 60s. Cần chuyển tính năng này qua trạng thái **unchecked** khi kiểm thử.



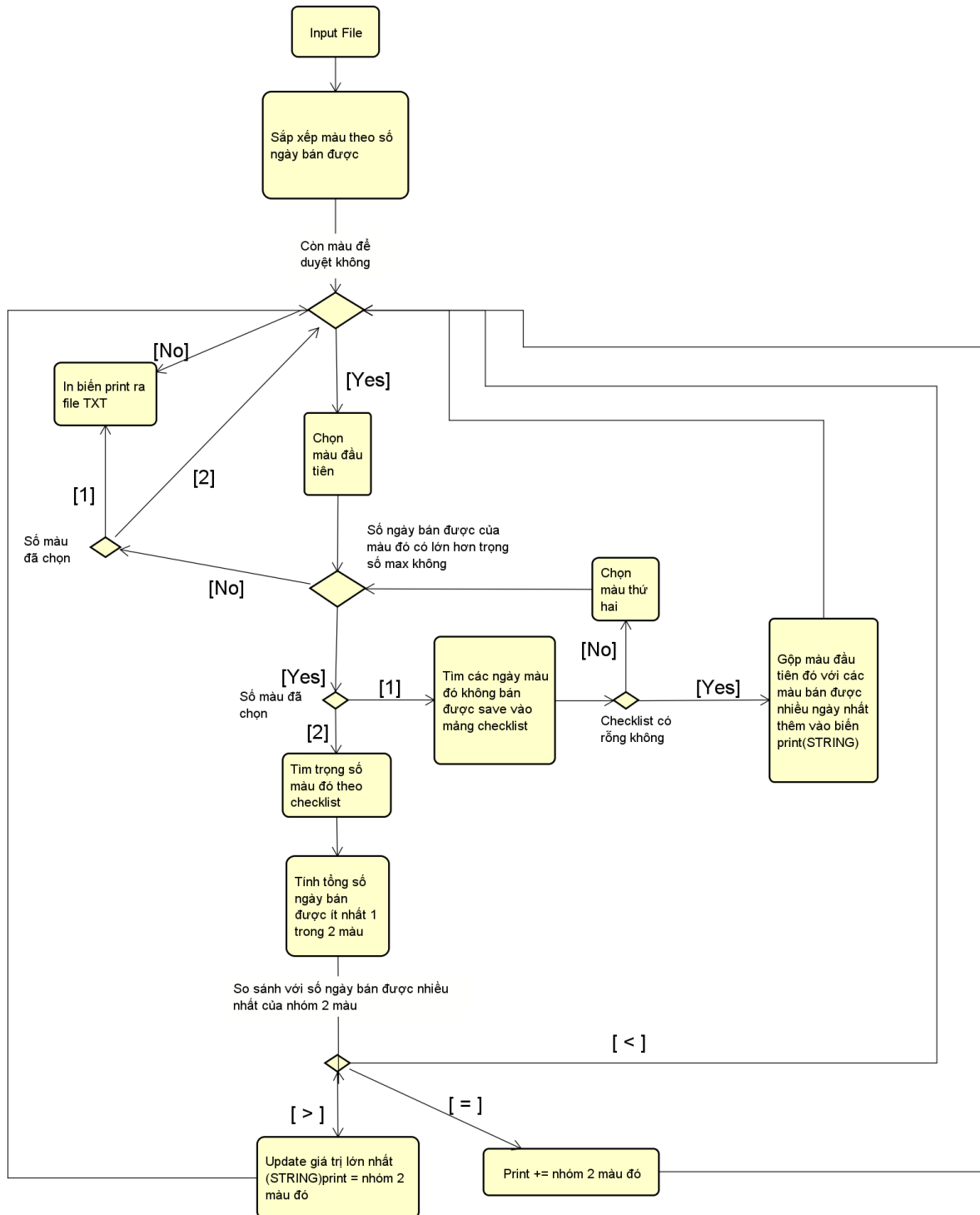
II. THUẬT TOÁN

1. Thuật toán sử dụng

Tính năng	Thuật toán áp dụng	Chi tiết
Sắp xếp dữ liệu đầu vào	Sắp xếp nổi bọt	Tính tổng số ngày bán được của từng màu sau đó sau đó dùng 2 vòng lặp for để sắp xếp mảng giá trị tổng số ngày bán được của từng màu theo chiều giảm dần.
Sắp xếp dữ liệu đầu ra	Sắp xếp nổi bọt	So sánh chỉ số cột của từng màu theo excel index sau đó dùng 2 vòng lặp for để sắp xếp
Xử lý nhóm màu theo yêu cầu	Duyệt mảng tuần tự	Chọn tuần tự từng bộ màu để so sánh tìm ra bộ màu có số ngày bán được là nhiều nhất.

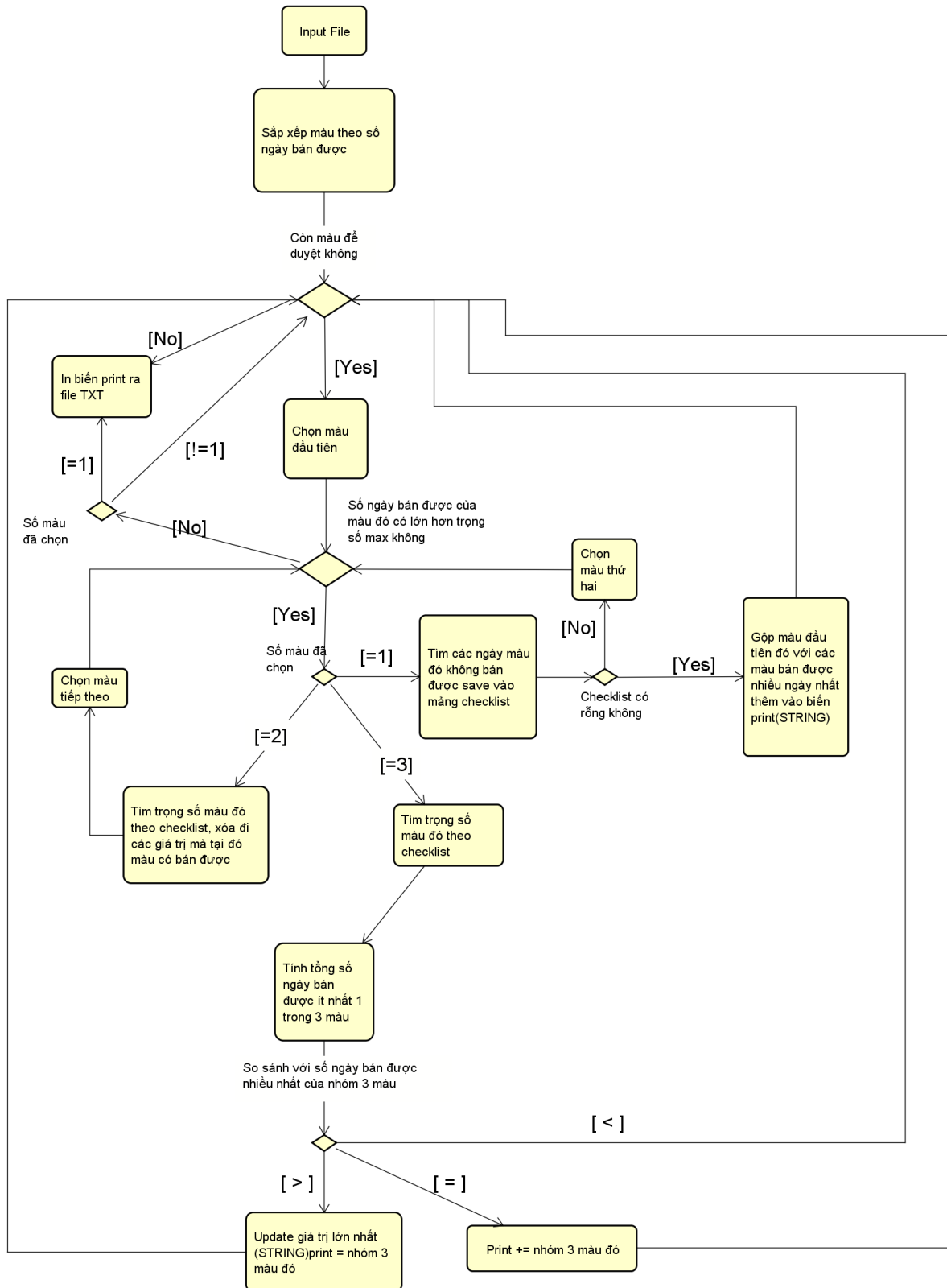
2. Thuật toán áp dụng cho trường hợp bộ 2 màu

act Sơ đồ khối thuật toán trường hợp bộ 2 màu



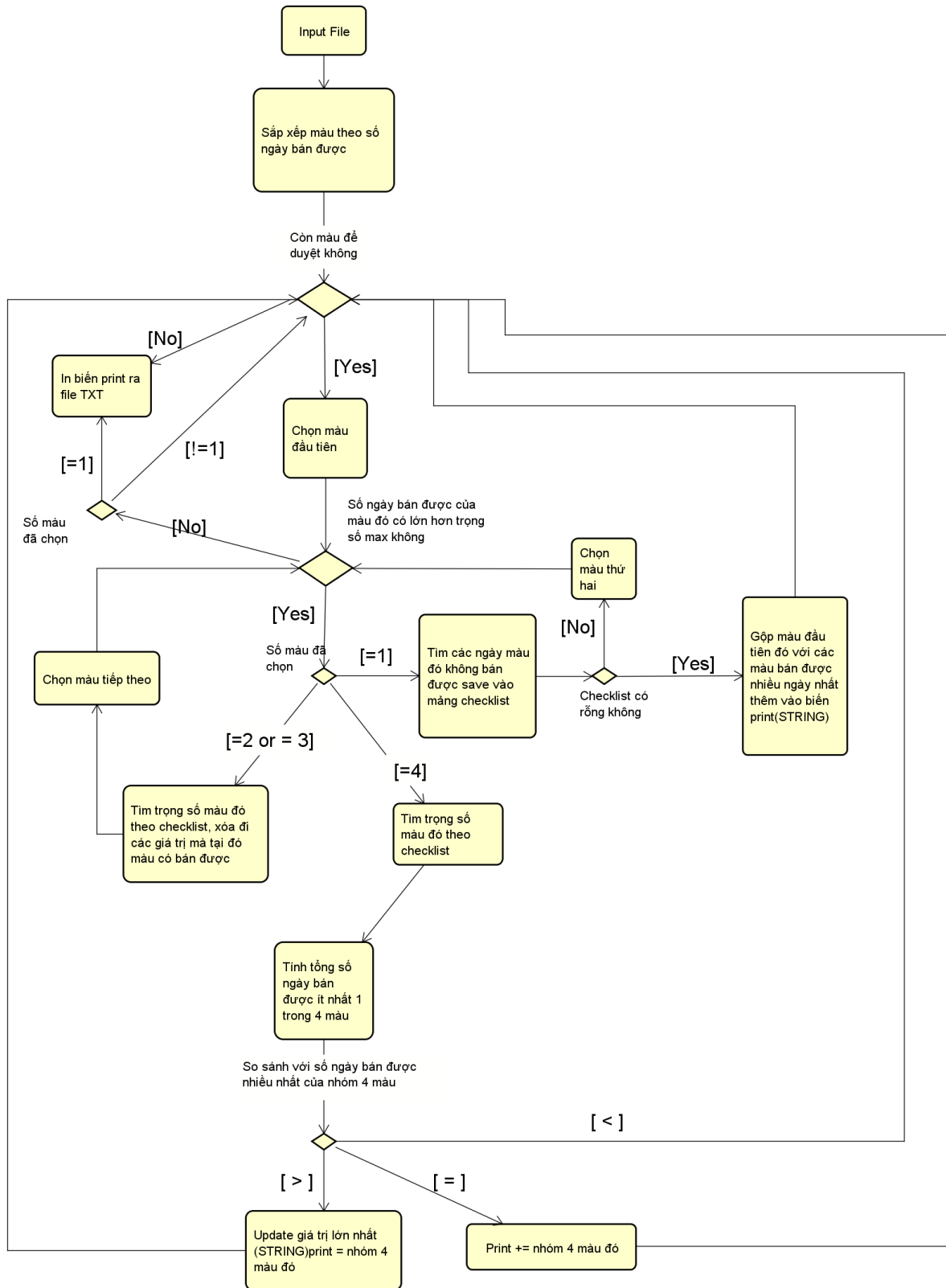
3. Thuật toán áp dụng cho trường hợp bộ 3 màu

act Sơ đồ khối thuật toán trường hợp bộ 3 màu



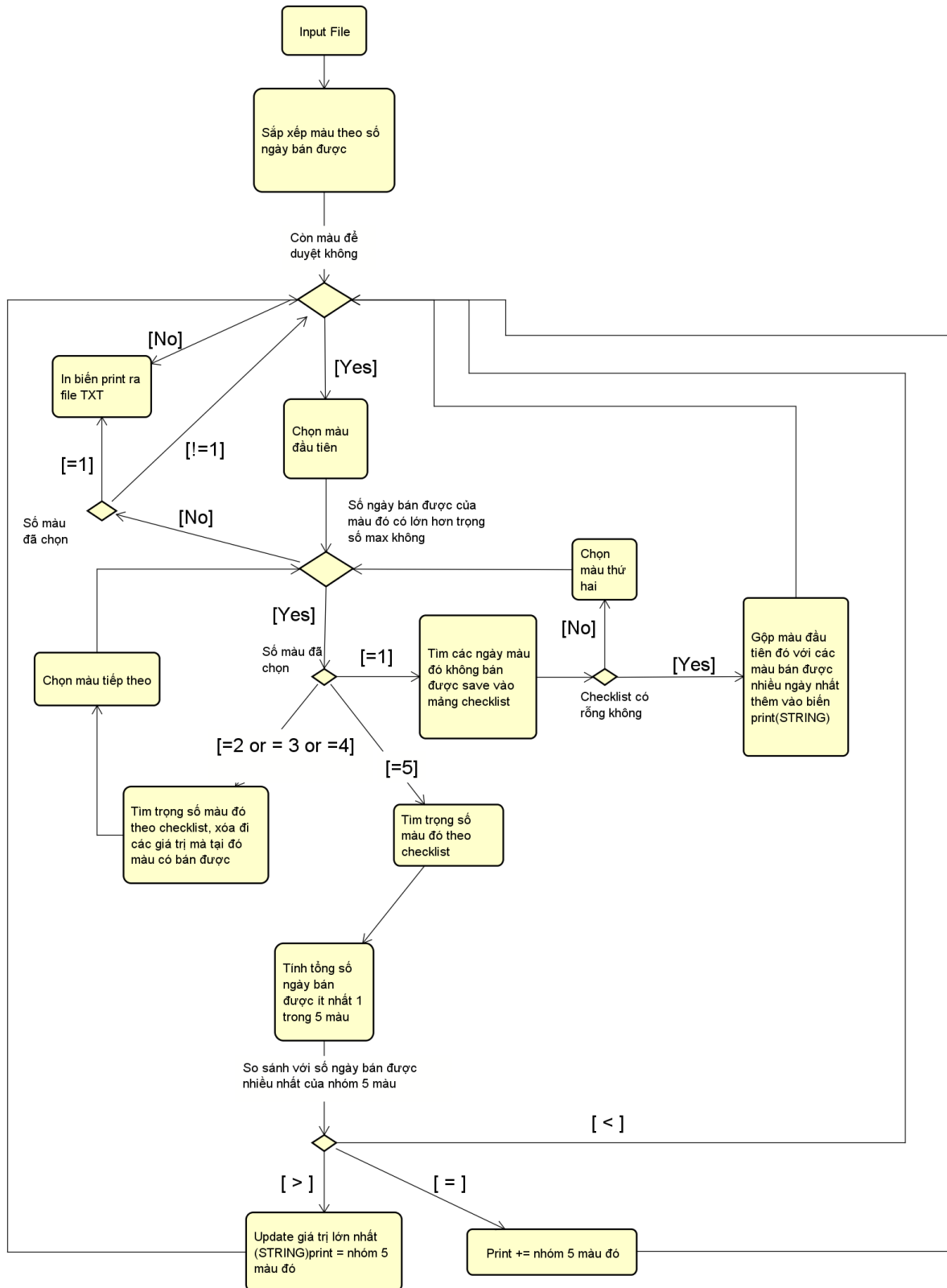
4. Thuật toán áp dụng cho trường hợp bộ 4 màu

act Sơ đồ khối thuật toán trường hợp bộ 4 màu



5. Thuật toán áp dụng cho trường hợp bộ 5 màu

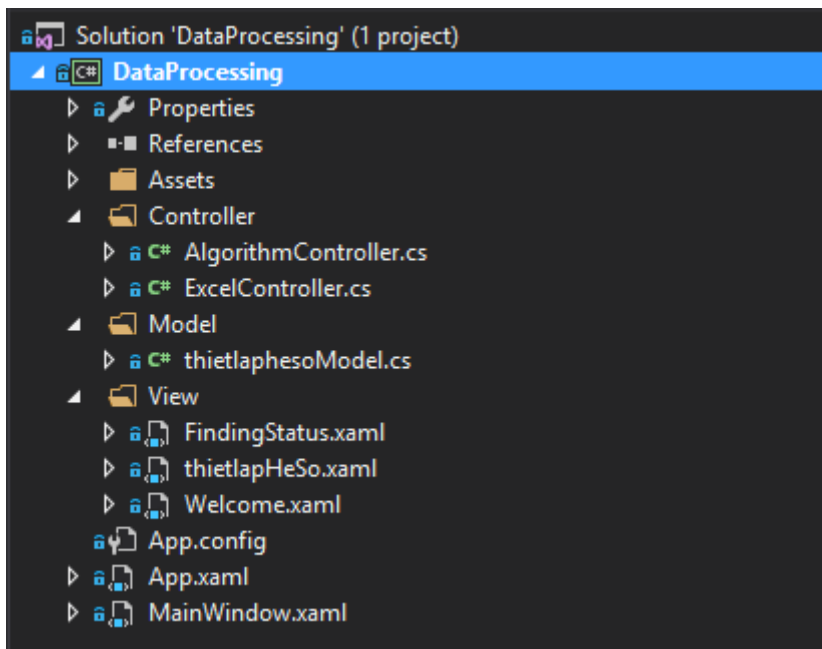
act Sơ đồ khối thuật toán trường hợp bộ 5 màu



III. MÃ NGUỒN

A. Cấu trúc MVC

1. Tổng quan.



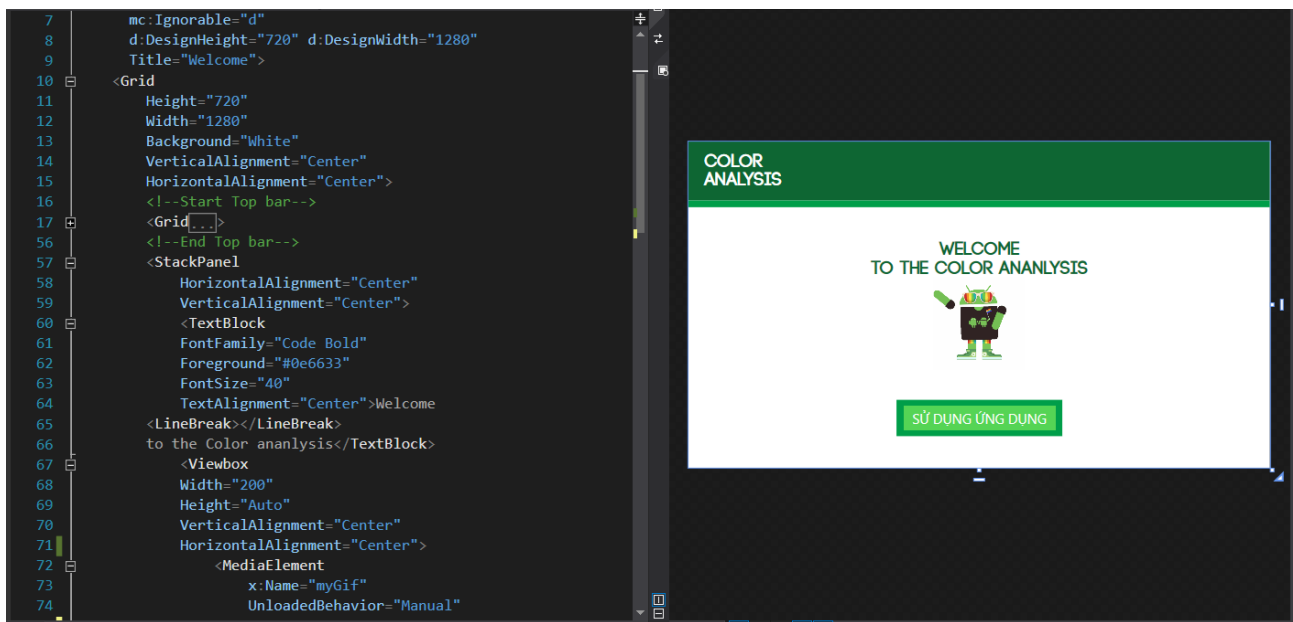
2. Các file thuộc Model, nội dung và nhiệm vụ.

thietlaphesoModel.cs: Sử dụng để khai báo các biến lưu trữ dữ liệu khi đọc từ excel và các hàm get set để cập nhật hoặc lấy dữ liệu cho các biến

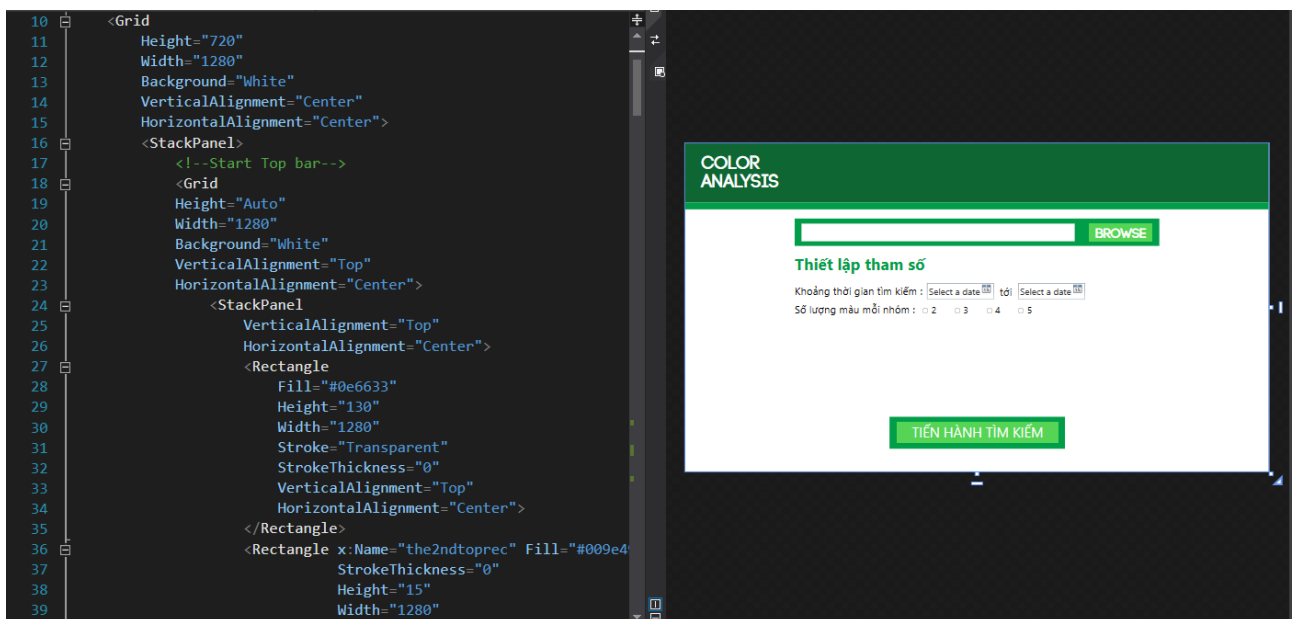
```
3 namespace DataProcessing.Model
4 {
5     public class thietlaphesoModel
6     {
7
8         private static int colcount = 1;
9         private static int rowcount = 1;
10        private static String[] color = new String[colcount - 1];
11        private static int[] value = new int[colcount - 1];
12        private static int[][] zeroOne = new int[colcount - 1][];
13        private static int[] index = new int[colcount - 1];
14
15        /// <summary> Set giá trị mảng cột
16        public void setColCount(int colcount1)...
17
18        /// <summary> Get giá trị mảng cột
19        public int getColCount()...
20
21        /// <summary> Set giá trị mảng hàng
22        public void setRowCount(int rowcount1)...
23
24        /// <summary> Get giá trị mảng hàng
25        public int getRowCount()...
26
27        /// <summary> Set giá trị mảng tên màu
28        public void setColor(String[] color1)...
29
30        /// <summary> Get giá trị mảng tên màu
31        public String[] getColor()...
32
33        /// <summary> Set giá trị mảng tổng màu bán được theo cột
34        public void setValue(int[] value1)...
35
36        /// <summary> Get giá trị tổng màu bán được theo cột
37        public int[] getValue()
```


3. Các file thuộc View, nội dung và nhiệm vụ.

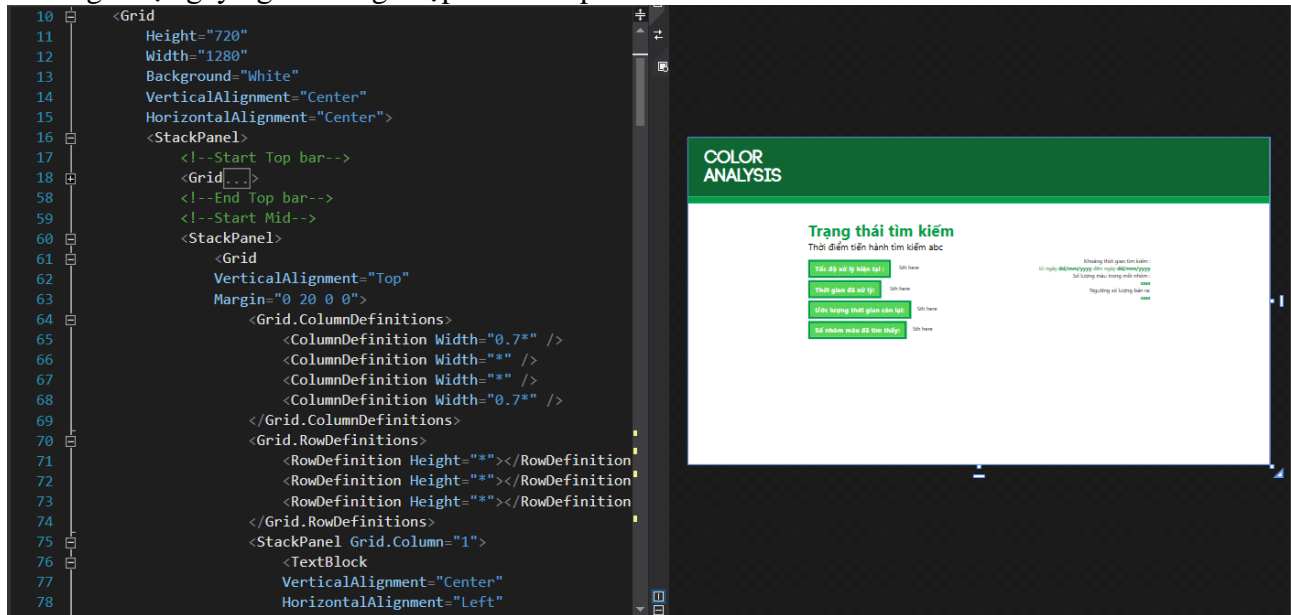
Welcome.xaml: hiển thị giao diện màn hình bắt đầu ứng dụng



thietlapHeSo.xaml: hiển thị giao diện màn hình thiết lập hệ số và xử lý các button trên giao diện



FindingStatus.xaml: hiển thị giao diện màn hình trạng thái tìm kiếm trong quá trình tìm kiếm. Các biến và giá trị tham số thiết lập sẽ được hiển thị (sau này). Hiện tại nhóm em tạm thời chỉ hiển thị demo giá trị ngày người dung nhập từ thietlapHeSo.xaml



4. Các file thuộc Controller, nội dung và nhiệm vụ.

AlgorithmController.cs: gồm các hàm xử lý nhóm 2, 3, 4, 5 màu. Lấy dữ liệu đầu vào từ thietlaphesoModel.cs, được View thietlapHeSo.xaml gọi tới để xử lý các nhóm màu tương ứng.

```

7      {
8      public class AlgorithmController
9      {
10
11
12      /// <summary>
13      /// Hàm xử lý nhóm 2 màu
14      /// </summary>
15      public void processGroup2()...
16
17
18      /// <summary> Hàm xử lý nhóm 3 màu
19      public void processGroup3()...
20
21
22      /// <summary>
23      /// Hàm xử lý nhóm 4 màu
24      /// </summary>
25      public void processGroup4()...
26
27
28      /// <summary>
29      /// Hàm xử lý nhóm 5 màu
30      /// </summary>
31      public void processGroup5()
32      {
33          thietlaphesoModel model = new thietlaphesoModel();
34          string print = "";
35          int currentValue1; // giá trị ở vòng 1
36          int currentValue2; // giá trị ở vòng 2
37          int currentValue3; // giá trị ở vòng 3
38          int currentValue4; // giá trị ở vòng 4
39          int biggestValue = 0;
40          int[] value = model.getValue();
41          int[][] zeroOne = model.getZeroOne();

```

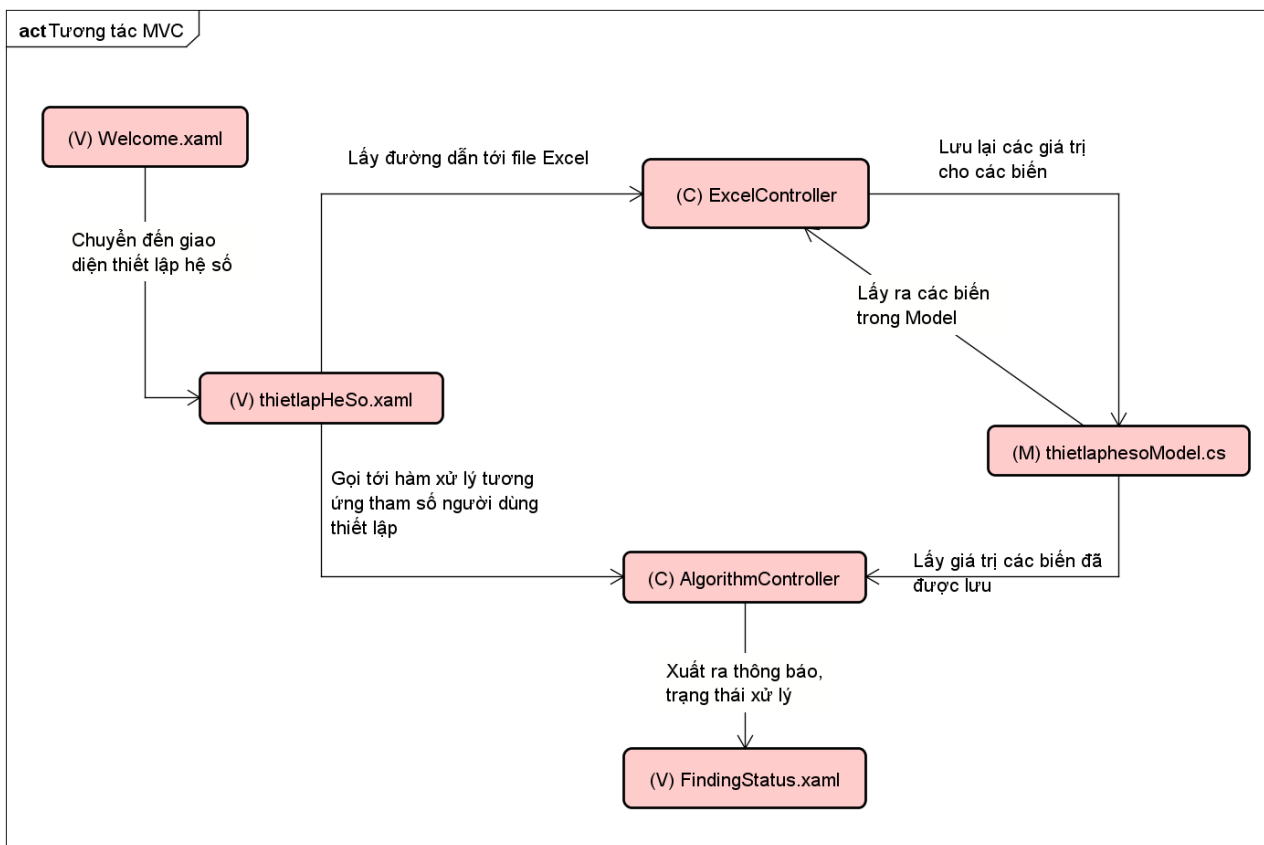
ExcelController.cs: bao gồm hàm readExcel, đọc excel người dung input vào và lưu trữ các giá trị cần thiết của excel vào các biến của thietlaphesoModel.cs

```

10 {
11     thietlaphesoModel model = new thietlaphesoModel();
12     public static int ngaybatdau = 0, ngayketthuc = 0;
13     public void readExcel(String path)
14     {
15
16         Excel.Application excel;
17         excel = new Excel.Application();
18         Excel.Workbook WB = excel.Workbooks.Open(path);
19         WB = excel.ActiveWorkbook;
20         Excel.Worksheet WS;
21         WS = WB.ActiveSheet;
22
23         int start = Environment.TickCount;
24
25         model.setColCount(WS.UsedRange.Columns.Count);
26         model.setRowCount(WS.UsedRange.Rows.Count);
27         String[] color = new String[model.getColCount() - 1];
28         int[] value = new int[model.getColCount() - 1];
29         int[][] zeroOne = new int[model.getColCount() - 1][];
30         int[] index = new int[model.getColCount() - 1];
31         for (int i = 0; i < model.getColCount() - 1; i++)
32         {
33             index[i] = i;
34         }
35         model.setIndex(index);
36         Excel.Range colnumber = WS.get_Range((Excel.Range)WS.Cells[2][1], (Excel.Range)WS.Cells[model.getColCount()][1]);
37         object mamau = colnumber.Value;
38         //Lấy mã màu vào mảng
39         int colorcount = 0;

```

B. Tương tác MVC



IV. ĐỊNH DẠNG DỮ LIỆU

A. Dữ liệu đầu vào

1. Định dạng

Định dạng file **.xls**; **.xlsx**

Thuộc tính	Nội dung	Kiểu dữ liệu source code	Dạng kiểu dữ liệu	Ví dụ
Thời gian	Hiển thị ngày bán hàng tại cột đầu tiên, từ dòng thứ 2	String	mm-dd-yyyy	03/23/2015
Màu	Màu được hiển thị tại dòng đầu tiên, từ cột thứ 2, xác định dựa vào excel index	String		
Trạng thái bán	Trạng thái bán hàng hiển thị tất cả các ô, loại trừ dòng và cột đầu tiên. Bán được: giá trị ô bằng 1 Không bán được: ô không có giá trị	Integer		

2. Loại dữ liệu đầu vào

Sự kiện	Xử lý	Tùy chọn xử lý của người dùng
Multi-sheet	Không	Không
Màu bị trùng	Lấy tất cả và coi là những thông số riêng biệt dựa theo excel index	Không
Giới hạn miền dữ liệu đầu vào	Không	Không

B. Dữ liệu đầu ra

1. Định dạng

Định dạng file	Nội dung	Cấu trúc	Ví dụ
.txt	Những bộ màu được bán với nhau nhiều nhất	[bộ màu]: [tổng số ngày bán được]	H-AU-AW-AX-BG: 180

	Nội dung	Cấu trúc chi tiết	Ví dụ
[bộ màu]	Những màu thuộc bộ màu hiển thị dựa vào excel index Số lượng màu dựa vào thông số đã chọn	[màu 1]- [màu 2]- [màu 3]- [màu 4]- [màu 5]	H-AU-AW-AX-BG
[tổng số ngày bán được]	Tổng số ngày bán được của bộ màu được ghép với nhau		180

2. Lưu trữ

Tên file **write.txt**

Lưu tại **Project1\DataProcessing\DataProcessing\bin\Debug**

C. TỐI ƯU

1. Sắp xếp dữ liệu đầu vào để xử lý

Cách tối ưu: Tính tổng số ngày bán được của từng màu và sắp xếp giảm dần

Hiệu quả: Nhóm màu bán được nhiều nhất có tỷ lệ tìm thấy sớm hơn

2. Đặt điều kiện để dừng trong quá xử lý

Cách tối ưu: Dừng khi *số ngày bán được của màu đang được duyệt nhỏ hơn trọng số lớn nhất của màu đang được xét trong bộ màu tìm thấy*

Hiệu quả: Không cần duyệt những trường hợp thừa, tiết kiệm thời gian tìm kiếm

3. So sánh hiệu quả

Mẫu so sánh: Tiến hành tìm kiếm bộ 5 màu trên tổng 107 màu Thời gian: Khoảng 7 tháng	
Trước khi tối ưu tìm kiếm hết 164.468s	Sau khi tối ưu tìm kiếm hết 60s
