

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**BÁO CÁO ĐỒ ÁN** **1**

**ĐỀ TÀI**: SỬ DỤNG CẤU TRÚC DỮ LIỆU ĐỂ XÂY DỰNG CHƯƠNG TRÌNH QUẢN LÝ BÁN SÁCH

**Giảng viên hướng dẫn : TRẦN TIẾN ĐỨC**

**Sinh viên thực hiện :**

1 : **Bùi Văn Hà**

MSSV:17110129

2 : **Lư Mạnh Quân**

MSSV:17110209

**TP. Hồ Chí Minh, tháng 11 năm 2019**

**MỤC LỤC**

DANH MỤC CÁC BẢNG……………………………………………………………………………..3

[PHẦN MỞ ĐẦU 4](#_Toc24292458)

[1. Lý do chọn đề tài: 4](#_Toc24292459)

[2. Mục đích nghiên cứu đế tài: 4](#_Toc24292460)

[3. Yêu cầu : 4](#_Toc24292461)

[4. Phương pháp nghiên cứu: 4](#_Toc24292462)

[PHẦN NỘI DUNG 5](#_Toc24292463)

[I. Đặc tả 5](#_Toc24292464)

[II. Phân công 5](#_Toc24292465)

[III. Thiết kế 6](#_Toc24292466)

[1) Giải thuật : Cấu túc dữ liệu Danh sách liên kết đơn 6](#_Toc24292467)

2) Phương thức trong Class LinkedList………………………………………………………………………………….6

IV. Chức năng của các Form……………………………………………………………………..13

1. Form Sách……………………………………………………………………………………...13

2. Form Loại Sách………………………………………………………………………………...20

3. Form Hóa Đơn………………………………………………………………………................21

[PHẦN KẾT LUẬN 25](#_Toc24292468)

[1. Kết quả đạt được 25](#_Toc24292469)

[2. Kết luận 25](#_Toc24292470)

[TÀI LIỆU THAM KHẢO 26](#_Toc24292471)

**DANH MỤC CÁC BẢNG**

* **BẢNG PHÂN CÔNG CÔNG VIỆC**
* **BẢNG DANH SÁCH CÁC CLASS TRONG CLASS LINKEDLIST**

# PHẦN MỞ ĐẦU

## Lý do chọn đề tài:

Nhằm củng cố kiến thức đã học ở các môn học trước, lập trình hướng đối tượng, cấu trúc dữ liệu và giải thuật, nhóm chúng em đã xây dựng chương trình quản lý việc bán sách bằng cách sử dụng cấu trúc dữ liệu dạng danh sách liên kết để thực hiện.

## Mục đích nghiên cứu đế tài:

Mục đích của đề tài là tạo ra một chương trình quản lý bán hàng thông qua việc sử dụng các tài nguyên, dữ liệu có sẵn hoặc được thêm hay chỉnh sửa trong các file text có dạng đuôi là.txt

## Yêu cầu :

Gồm 10 yêu cầu như trong file word Yêu cầu nhóm em có kèm theo.

## Phương pháp nghiên cứu:

Nghiên cứu tham khảo tra cứu thông tin và tài liệu.

Hoàn thiện đề tài với sự hướng dẫn của giảng viên.

# PHẦN NỘI DUNG

## Đặc tả

* Chương trình quản lý việc lưu trữ thông tin về sách, hóa đơn bán sách.
* Với việc lưu trữ ta có thể :
* Thêm, xóa, sửa, sắp xếp dữ liệu về Sách, Loại Sách, Lịch sử Hóa Đơn theo ý muốn
* Kiểm tra lịch sử mua hàng có Giá trị Lớn Nhất (Best Sell)
* Tìm kiếm sách theo từng Loại Sách riêng biệt, xuất Lịch sử Hóa Đơn ra Doanh Thu và xuất ra Bill theo dạng đã được yêu cầu
* **Câu hỏi được đặt ra:**
* *Tại sao lại sử dụng danh sách liên kết đơn?*

*Trả lời :* Do chưa biết người dùng sẽ nhập bao nhiêu ký tự và danh sách liên kết đơn giải quyết được vấn đề đó (Danh sách liên kết là một cấu trúc dữ liệu có thể giữ một số lượng phần tử tùy ý và dễ dàng thay đổi kích thước, cũng như dễ dàng bỏ đi các phần tử bên trong nó)

* *Tại sao không sử dụng mảng ?*

*Trả lời:* Khá phức tạp

* **Giao diện:** Chương trình được chạy trên một hệ Windows Form, trong đó chỉ được phép sử dụng các ToolBox : TextBox, Label, Button, MenuStrip, MessageBox.

## Phân công

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên SV | Công việc | Phẩn trăm đóng góp |
| 1 | Lư Mạnh Quân | -Tìm hiểu thuật toán danh sách liên kết đơn trên C#  -Viết hàm thêm đầu, thêm cuối, xóa, sửa tập tin, in lịch sử hóa đơn qua doanh thu, tìm Best Sell, sắp xếp file Sách theo đơn giá. tìm sách theo loại sách, Xuất Bill.  -Viết Báo cáo Word cho Đồ Án 1 | 50% |
| 2 | Bùi Văn Hà | -Tìm hiểu thuật toán danh sách liên kết đơn trên C#  -Viết hàm thêm đầu, thêm cuối, xóa, sửa tập tin, in lịch sử hóa đơn qua doanh thu, tìm Best Sell, sắp xếp file Sách theo đơn giá, tìm sách theo loại sách, Xuất Bill,  -Viết Báo cáo Word cho Đồ Án 1 | 50% |

*Bảng 1: Bảng phân công việc*

## Thiết kế

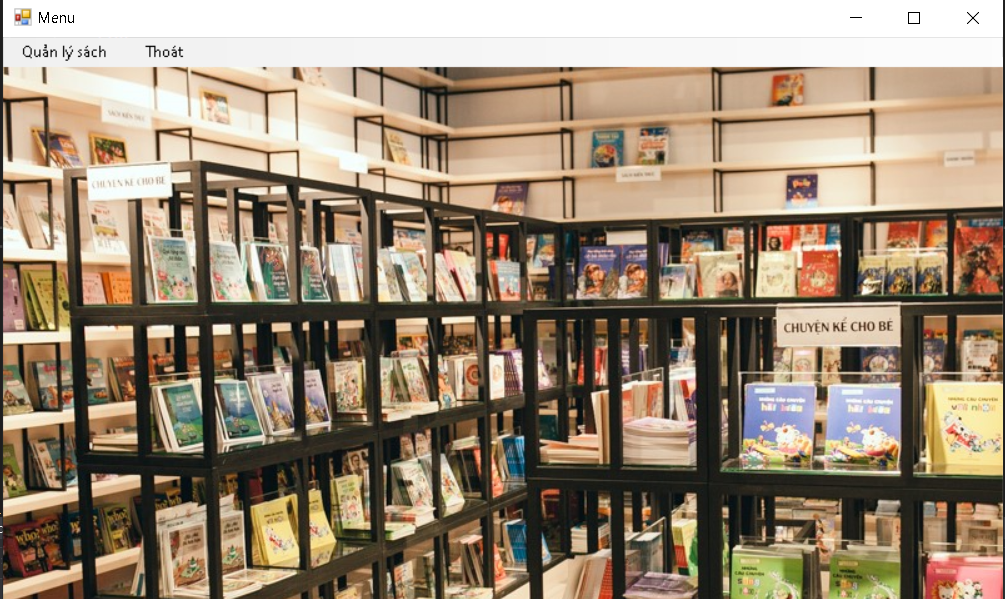
### Giải thuật : Cấu túc dữ liệu Danh sách liên kết đơn

1. Phương thức trong Class **LinkedList** :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Phương thức | Mục đích | Dòng bắt đầu khai báo | SV phụ trách |
| 1 | public SACH info;  public Node Next;  public Node()  {  }  public Node(SACH x, Node next)  {  this.info = x;  this.Next = next;  } | Khởi tạo danh sách liên kết | Dòng 17 | Bùi Văn Hà, Lư Mạnh Quân |
| 2 | public Node CreateNode(SACH x)  {  Node p = new Node();  p.info = new SACH();  p.info.MaSach = x.MaSach;  p.info.TenSach = x.TenSach;  p.info.LoaiSach = x.LoaiSach;  p.info.DonGia = x.DonGia;  p.info.NgayMua = x.NgayMua;  p.info.TacGia = x.TacGia;  p.info.NXB = x.NXB;  p.info.ThongKe = x.ThongKe;  if (p.info == null)  {  return null;  }  return p;  } | Tạo mới Node | Dòng 30 | Bùi Văn Hà, Lư Mạnh Quân |
| 3 | public class SACH  {  public string MaSach;  public string TenSach;  public string LoaiSach;  public float DonGia;  public string NgayMua;  public float SoLuongMua;  public float TongTien;  public float GiamGia;  public float ThanhTien;  public string TacGia;  public string NXB;  public float ThongKe;  // Hàm khởi tạo  public SACH()  {  }    // Hàm nhập  public void Nhap(string masach, string tensach, string loaisach, float dongia)  {  this.MaSach = masach;  this.TenSach = tensach;  this.LoaiSach = loaisach;  this.DonGia = dongia;  }  public void Nhap6(string masach, string tensach, string loaisach, float dongia, string ngaymua, float soluongmua)  {  this.MaSach = masach;  this.TenSach = tensach;  this.LoaiSach = loaisach;  this.DonGia = dongia;  this.NgayMua = ngaymua;  this.SoLuongMua = soluongmua;  }  public void Nhap2(string loaisach, string tacgia, string nxb)  {  this.LoaiSach = loaisach;  this.TacGia = tacgia;  this.NXB = nxb;  }  } | Khai báo class SACH | Dòng51 | Bùi Văn Hà, Lư Mạnh Quân |
| 4 | public class LinkedList  {  // Khai báo  public Node Head;  public Node Tail;  public List<SACH> found = new List<SACH>(); //Danh sách kiểu SACH để chứa những sách thỏa điều kiện tìm    // Hàm khởi tạo  public LinkedList()  {  Head = Tail = null;  }  // Hàm kiểm tra danh sách liên kết có rỗng hay không  public int IsEmptyList()  {  if (this.Head == null)  return 1; // Danh sách rỗng  return 0; // Danh sách không rỗng  }  // Thêm vào cuối danh sách  public void addTail(SACH x)  {  Node p = new Node(x, null);  if (this.Head == null)  this.Head = this.Tail = p;  else  {  this.Tail.Next = p;  this.Tail = p;  }  }  //Thêm vào đầu danh sách  public void addHead(SACH x)  {  Node p = new Node(x, Head);  Head = p;  if (Tail == null)  Tail = Head;  }  public void SearchNode(LinkedList myList, string x)  {  Node p = myList.Head;  while (p != null)  {  SACH temp = new SACH(); //Biến nhớ kiểu sinh viên để đưa vào list  if ((x == p.info.MaSach) || (x == p.info.TenSach) || (x == p.info.LoaiSach) || (x == p.info.DonGia.ToString()))  {  temp.Nhap(p.info.MaSach, p.info.TenSach, p.info.LoaiSach, p.info.DonGia);  found.Add(temp);  }  p = p.Next;  }  }    // Xóa phần tử đầu  public void removeHead(LinkedList myList)  {  if (myList.Head == null)  {  }  else if (myList.Head == myList.Tail)  {  myList.Head = null;  myList.Tail = null;  }  else  {  Node temp = new Node();  temp = myList.Head;  myList.Head = myList.Head.Next;  temp = null;  }  }  // Xóa nút phía sau  public void removeAfter(LinkedList myList, Node node)  {  if (myList.Head == null)  { }  else  {  Node temp = new Node();  temp = node.Next;  if (temp != null)  {  node.Next = temp.Next;  temp = null;  }  }  }  // Xóa nút có giá trị cần xóa  public void removeKey(LinkedList myList, string x)  {  if (myList.Head == null)  { }  else  {  Node result = new Node();  result = myList.Head;  Node before = new Node();  before = null;  while (result.info.MaSach != x)  {  before = result;  result = result.Next;  }  if (result != null)  {  if (result == myList.Head)  removeHead(myList);  else  removeAfter(myList, before);  }  }  }  // Đếm node  public int CountNode(LinkedList myList)  {  int dem = 0;  Node p = new Node();  p = myList.Head;  while (p != null)  {  dem++;  p = p.Next;  }  return dem;  }  // Hàm hủy danh sách liên kết  public void clear(LinkedList myList)  {  Node temp = new Node();  temp = myList.Head;  while (temp != null)  {  myList.removeHead(myList);  temp = temp.Next;  }  }  //Sắp xếp theo MaSach  public void Sort()  {  LinkedList lnv1 = new LinkedList();  LinkedList lnv2 = new LinkedList();  Node pivot, p;  if (Head == Tail)  return;  pivot = Head;  p = Head.Next;  float dongiaPivot = pivot.info.DonGia;  while (p != null)  {  Node q = p;  p = p.Next;  q.Next = null;  float dongiaQ = q.info.DonGia;  if (dongiaQ < dongiaPivot)  lnv1.addTail(q.info);  else  lnv2.addTail(q.info);  };  lnv1.Sort();  lnv2.Sort();  if (lnv1.IsEmptyList() == 0)  {  this.Head = lnv1.Head;  lnv1.Tail.Next = pivot;  }  else  this.Head = pivot;  pivot.Next = lnv2.Head;  if (lnv2.IsEmptyList() == 0)  this.Tail = lnv2.Tail;  else  this.Tail = pivot;  }  public void Sort1()  {  LinkedList lnv1 = new LinkedList();  LinkedList lnv2 = new LinkedList();  Node pivot, p;  if (Head == Tail)  return;  pivot = Head;  p = Head.Next;  float dongiaPivot = pivot.info.DonGia;  while (p != null)  {  Node q = p;  p = p.Next;  q.Next = null;  float dongiaQ = q.info.DonGia;  if (dongiaQ > dongiaPivot)  lnv1.addTail(q.info);  else  lnv2.addTail(q.info);  };  lnv1.Sort1();  lnv2.Sort1();  if (lnv1.IsEmptyList() == 0)  {  this.Head = lnv1.Head;  lnv1.Tail.Next = pivot;  }  else  this.Head = pivot;  pivot.Next = lnv2.Head;  if (lnv2.IsEmptyList() == 0)  this.Tail = lnv2.Tail;  else  this.Tail = pivot;  }  public void Sort2()  {  LinkedList lnv1 = new LinkedList();  LinkedList lnv2 = new LinkedList();  Node pivot, p;  if (Head == Tail)  return;  pivot = Head;  p = Head.Next;  string[] ngaymuaPivot = pivot.info.NgayMua.Split('-');  while (p != null)  {  Node q = p;  p = p.Next;  q.Next = null;  string[] ngaymuaQ = q.info.NgayMua.Split('-');  if (Convert.ToInt32(ngaymuaQ[2]) > Convert.ToInt32(ngaymuaPivot[2]))  lnv1.addTail(q.info);  else if (Convert.ToInt32(ngaymuaQ[2]) < Convert.ToInt32(ngaymuaPivot[2]))  lnv2.addTail(q.info);  else if (Convert.ToInt32(ngaymuaQ[1]) > Convert.ToInt32(ngaymuaPivot[1]))  lnv1.addTail(q.info);  else if (Convert.ToInt32(ngaymuaQ[1]) < Convert.ToInt32(ngaymuaPivot[1]))  lnv2.addTail(q.info);  else if (Convert.ToInt32(ngaymuaQ[0]) > Convert.ToInt32(ngaymuaPivot[0]))  lnv1.addTail(q.info);  else  lnv2.addTail(q.info);  };  lnv1.Sort2();  lnv2.Sort2();  if (lnv1.IsEmptyList() == 0)  {  this.Head = lnv1.Head;  lnv1.Tail.Next = pivot;  }  else  this.Head = pivot;  pivot.Next = lnv2.Head;  if (lnv2.IsEmptyList() == 0)  this.Tail = lnv2.Tail;  else  this.Tail = pivot;  }  } | Tổng hợp các hàm chức năng của danh sách liên kết, bào gồm :  -Kiếm tra rỗng  -Thêm vào cuối  -Thêm váo đầu  -Tìm kiếm 1 Node  -Xóa phần tử đầu  -Xóa phần tử sau 1 phần tử bất kỳ  -Xóa 1 phần tử tai 1 vị trí bất kỳ  -Đếm số Node  -Húy danh sách liên kết hiện hành  -Sắp xếp tăng  -Sắp xếp giảm  -Sắp xếp tìm Best Sell |  | Bùi Văn Hà, Lư Mạnh Quân |
| 5 | **Các form dựa vào class LinkedList.cs đễ thực hiện các chức năng tương ứng** |  |  | Bùi Văn Hà, Lư Mạnh Quân |

* Lưu ý : Khi muốn thêm, xóa, sửa, lưu bất kỳ thông tin nào, ta cần Đọc chúng trong file rồi xuất ra màn hình TextBox trước, rồi mới thực hiện các bước như trên.

1. **Chức năng của các Form:**
2. *Form Menu chính*

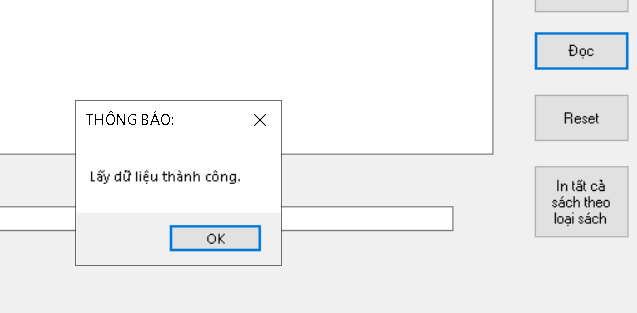
* **Mô hình Form :
* Menu **Quản lý sách** : Truy cập theo file để thực hiện các chức năng tương ứng
* Menu **Thoát** : Thoát hoàn toàn khỏi chương trình khi không có nhu cầu sử dụng

1. *Form Sách*

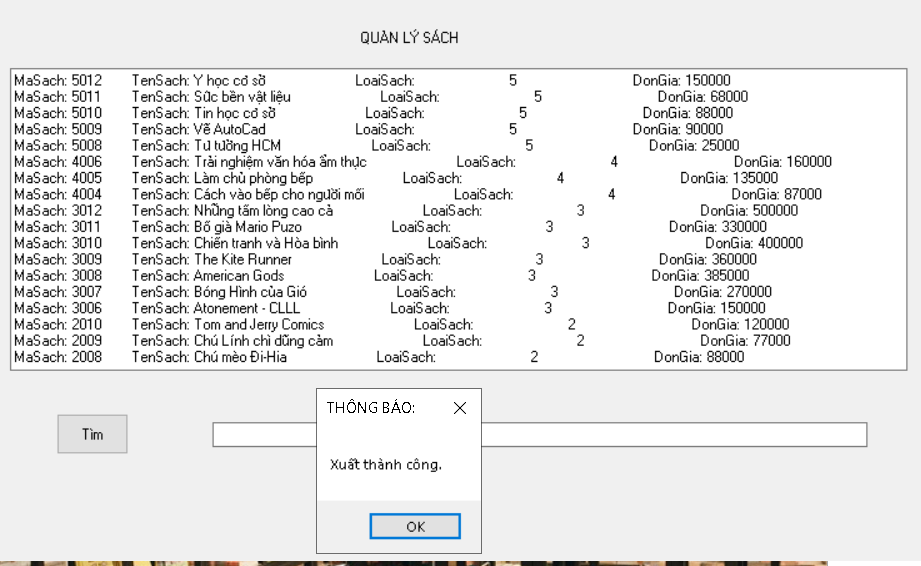
* Mô hình Form:



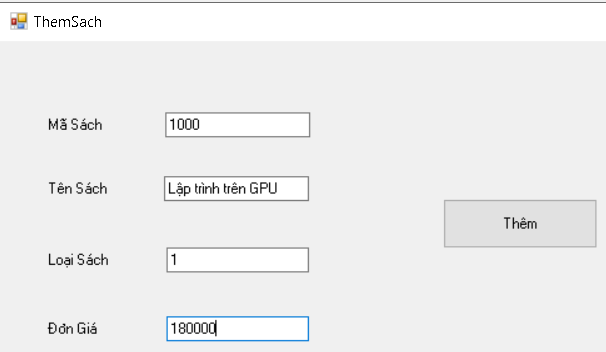
* Button **Đọc**

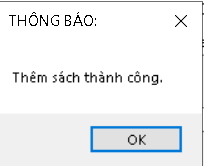
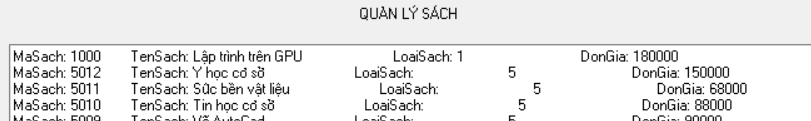


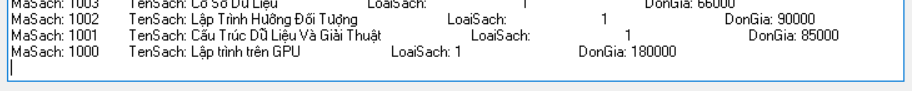
* Button **Xuất**



* Button **Thêm trước** và **Thêm sau:** Khi bấm, 1 form con mới có tên **Themsach.cs** sẽ hiện ra dùng để nhập những thông tin cần thiết cho cuốn sách mới :



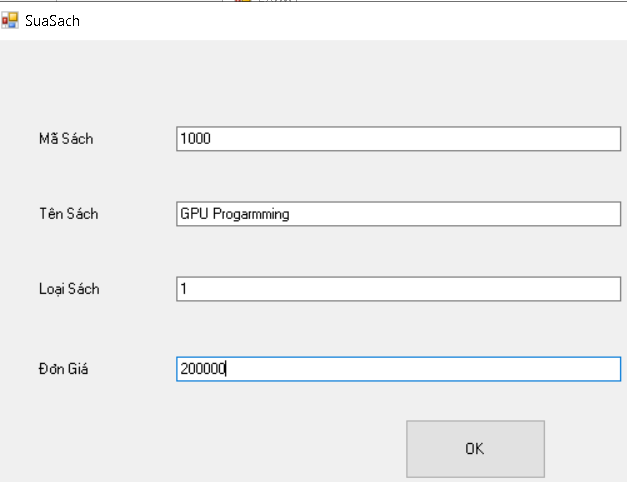
* Nếu thêm thành công sẽ báo :
* Sau đó Refresh lại thông tin bằng nút **xuất** :

-Lúc này trên dòng đầu tiên đã chứa thông tin về cuốn sách được thêm vừa rồi, (nếu là **Thêm sau**, thì nó sẽ được nằm ở dòng cuối cùng như sau :

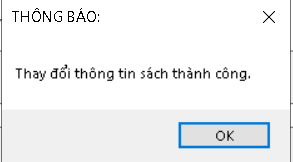
bấm nút **Lưu** nếu muốn lưu giữ lại thông tin, nếu lưu thành công sẽ báo :



* Button **Sửa :** Sửa thông tin tệp tin:
* Ví dụ : Chúng ta muốn sửa tập tin này :
* Khi ta bấm nút **Sửa**, form con **Suasach** sẽ hiện ra để sửa lại thông tin



* Nếu sửa thành công sẽ báo :

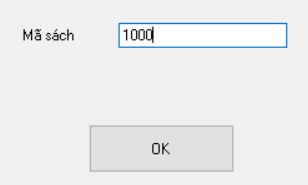


* Sau đó nhần Xuất để check lại. Lúc này, thông tin được sửa thành :
* Nhấn nút **Lưu** để lưu lại những thay đỗi vừa thực hiện.
* Button **Xóa :**
* Tiến hành xóa 1 tập tin đang nằm trong file

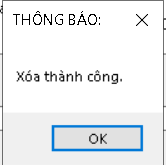
Ví dụ : Ta muốn xóa 1 tập tin như sau:

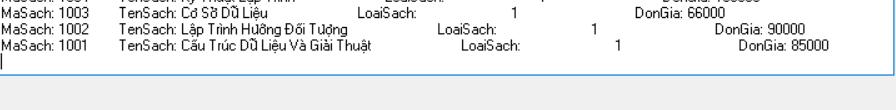


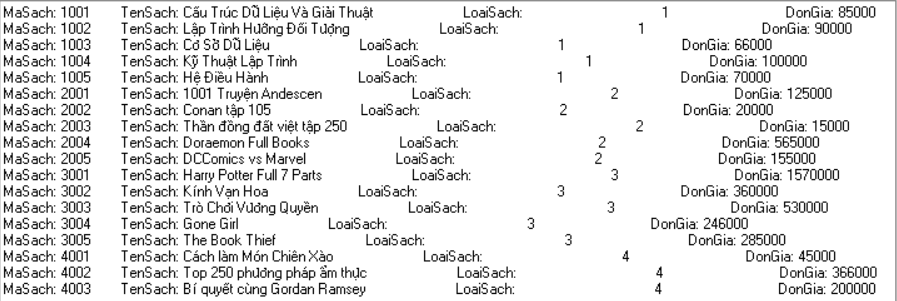
* Chick vào nút **Xóa**, khi đó form con **XoaSach** sẽ hiện ra, ta điền mã cuốn sách cần xóa :

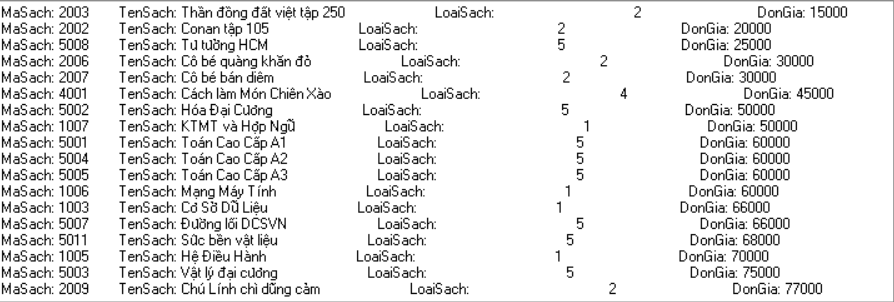


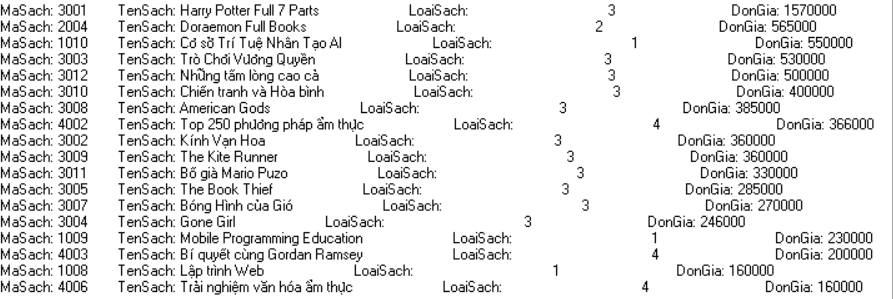
* Nếu xóa thành công sẽ báo :



* Check lại bằng nút **Xuất**, ta thấy sách có mã 1000 không còn nữa
* Nhấn **Lưu** để lưu lại những thay đổi vừa rồi.
* Button **Sắp tăng** và **Sắp giảm** :
* Đầu tiên, ta đọc dữ liệu trong file rồi xuất ra TextBox như sau :

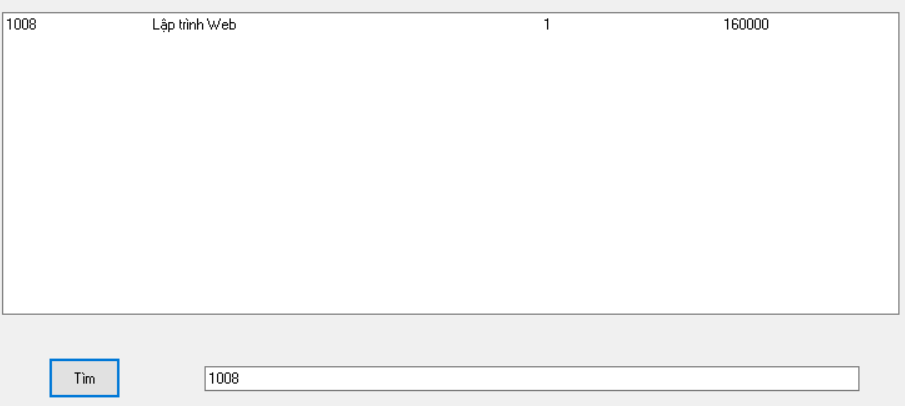


* Ta thấy ở cột đơn giá chưa có sự sắp xếp theo giá trị, từ đây ta sẽ sắp xếp lại sách theo chiều **tăng dần** hoặc **giảm dần** dữa vào cột **Đơn Giá** :
* Sắp **tăng** theo **Đơn giá** :
* Sắp **giảm** theo **Đơn giá** :

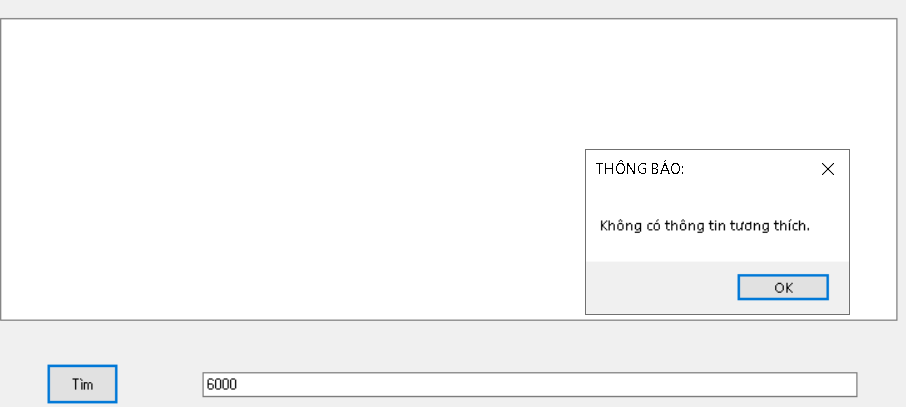


* Button **Tìm 1 cuốn sách dựa theo mã sách có trong file** :
* Dữa vào ô nhập, ta nhập mã sách cần tìm rồi chọn nút **Tìm :**

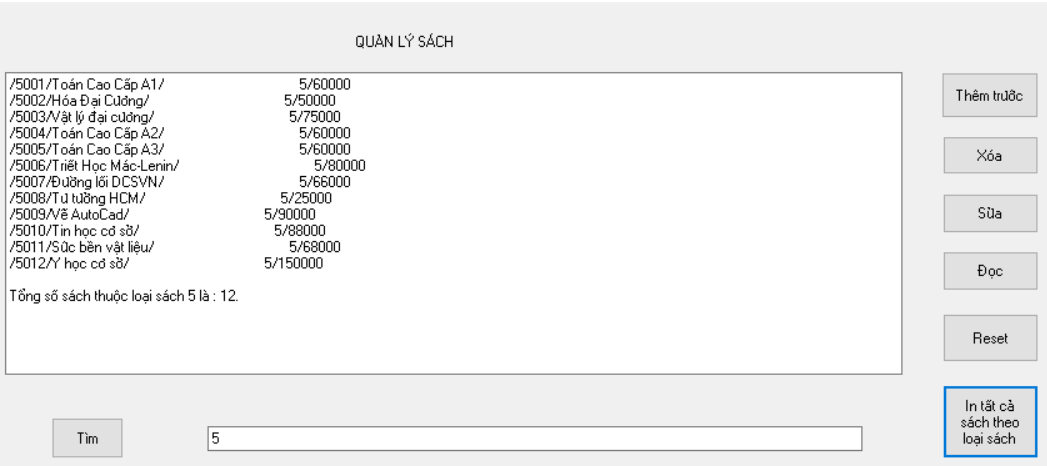
+ Ví dụ ta tìm sách có mã 1008. Nếu thành công sẽ như sau :



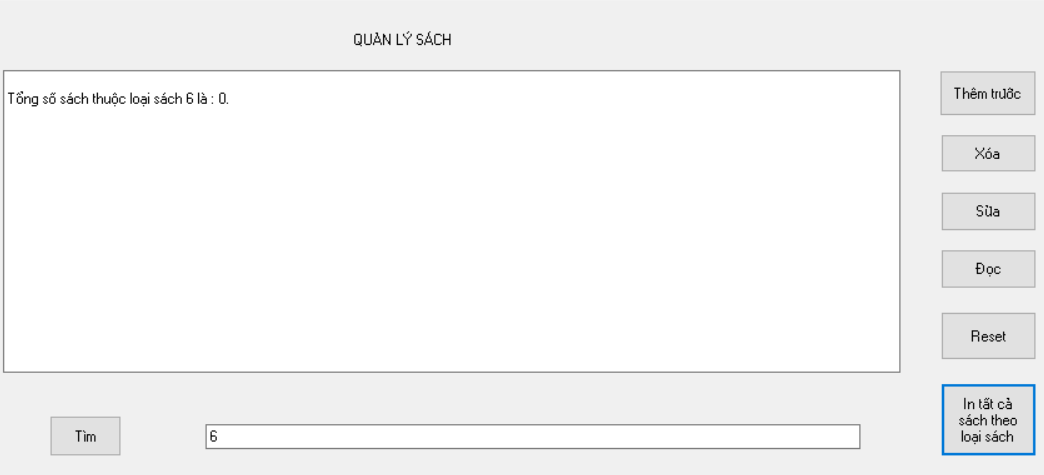
+ Còn nếu thất bại, ví dụ ta tim mã sách 6000 vốn chưa có trong file, sẽ báo :



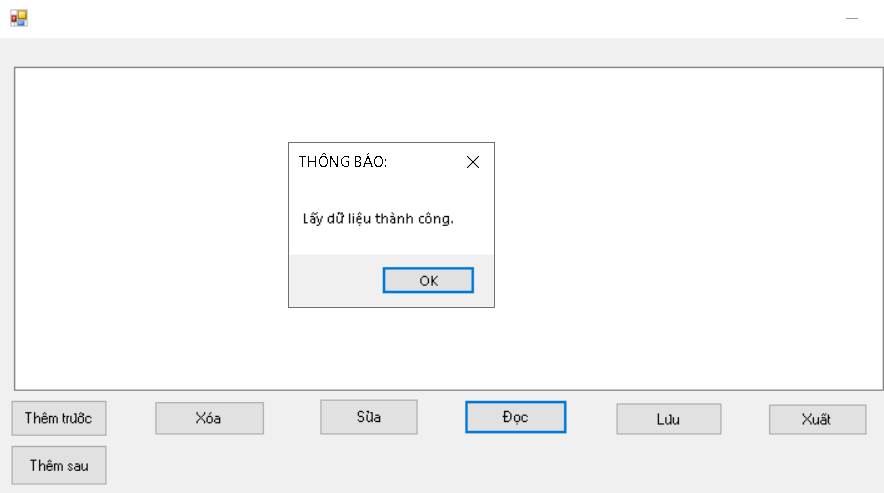
* Button **Tìm tất cả sách có sẵn dựa theo loại sách có trong file** :
* Nhập loại sách trong ô Nhập rồi Tìm, nếu có tồn tại sẽ xuất ra :

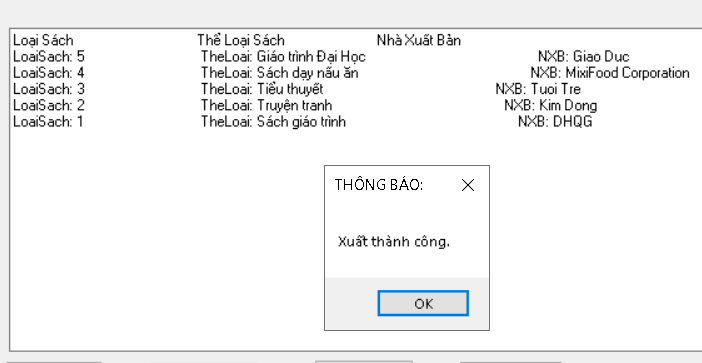


* Còn nếu không có thì TextBox sẽ trống trơn và in ra là 0 có cuốn nào.

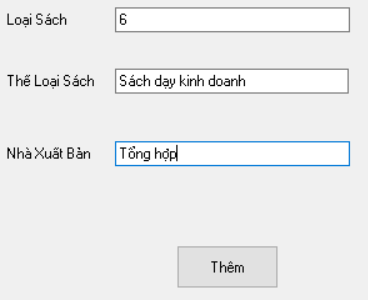


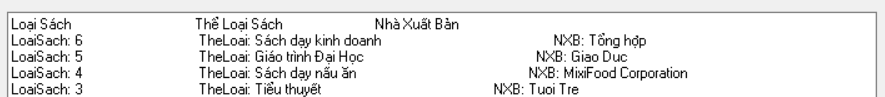
1. *Form Loại Sách*

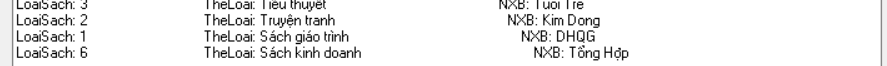
* Button **Đọc** và **Xuất** : chức năng như form **Sách** :
* 



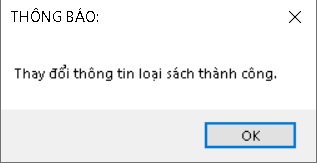
* Button **Thêm trước** và **Thêm sau** : chức năng tương tự như bên form **Sach** :
* Nhập thông tin :

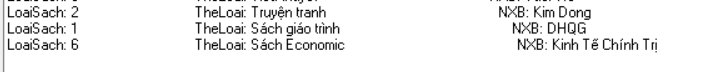


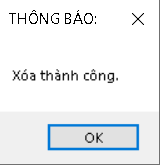
* Kết quả : + **Thêm trước** :

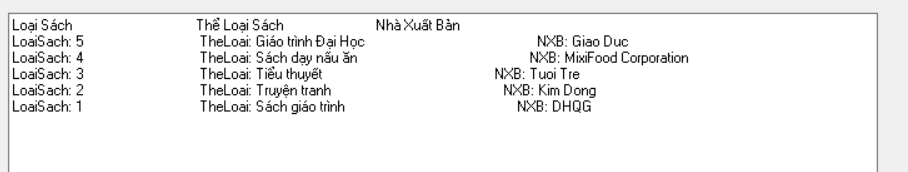
+ **Thêm sau** :

* Button **Sửa và Xóa :**
* Ví dụ : Sửa và xóa dòng Loại sách : 6



* Kết quả :

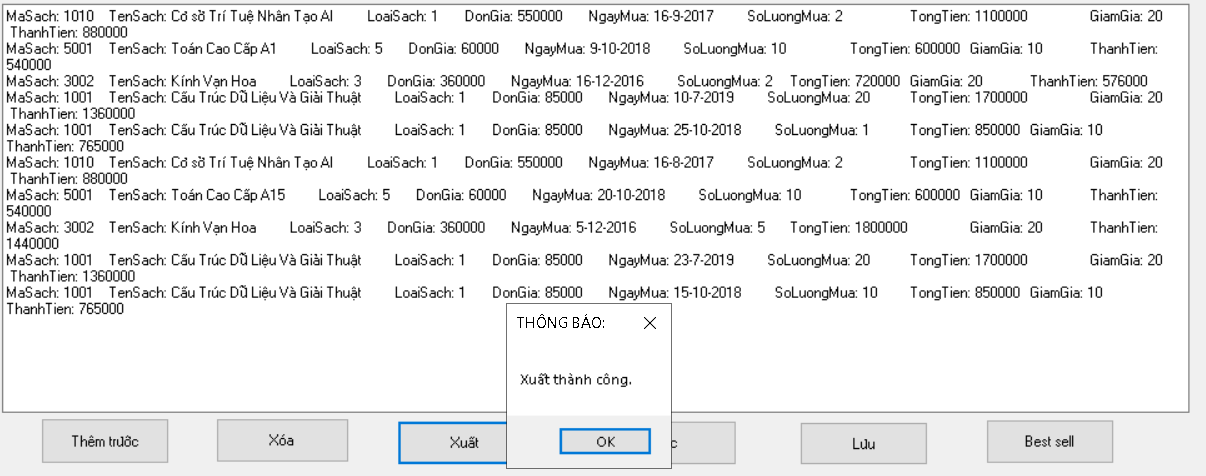


* Kết quả :
* Button **Lưu** : Như form **Sách**.

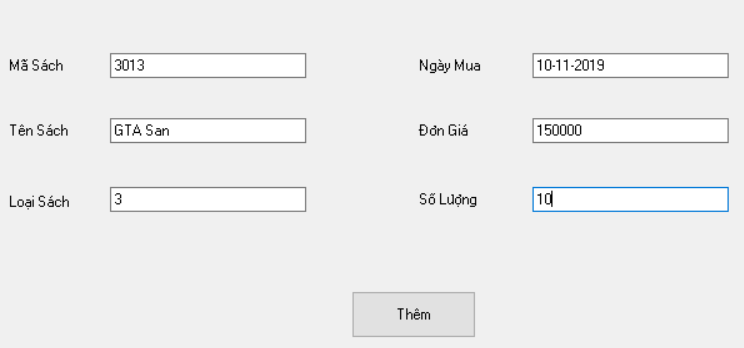
1. *Form Hóa Đơn*

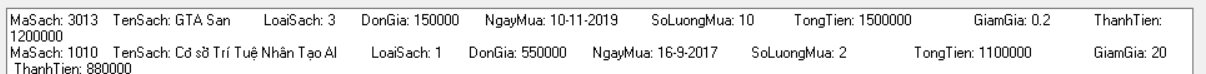
* Button **Đọc, Xuất** : Tương tự **Sach, LoaiSach**

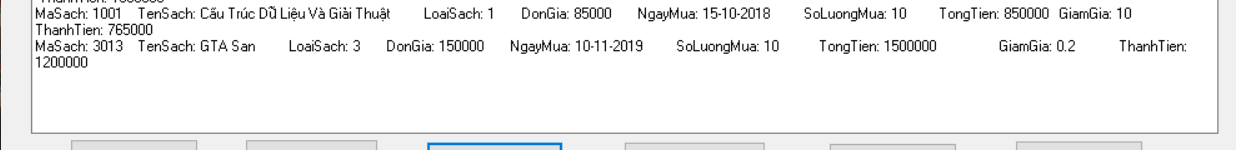


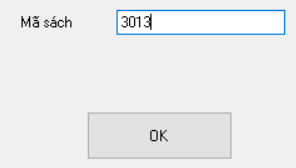


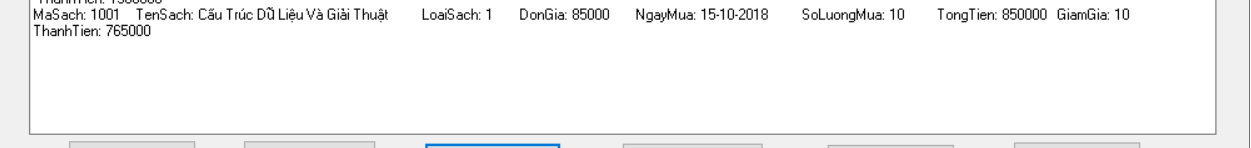
* Button **Thêm trước, Thêm sau, Xóa** : Tương tự **Sach, LoaiSach**
* **Thêm Trước :**



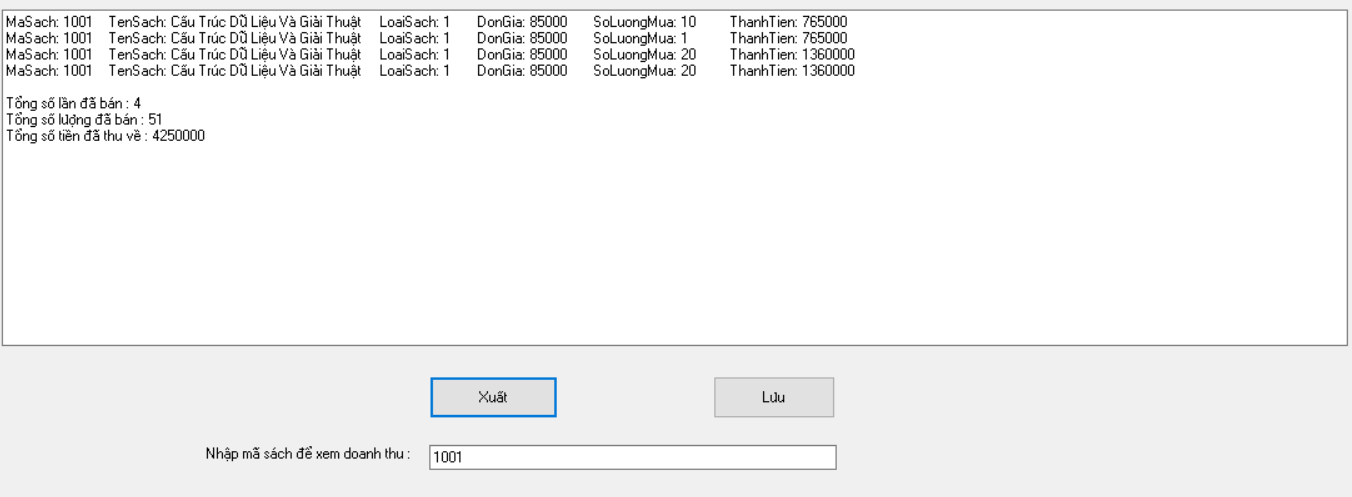


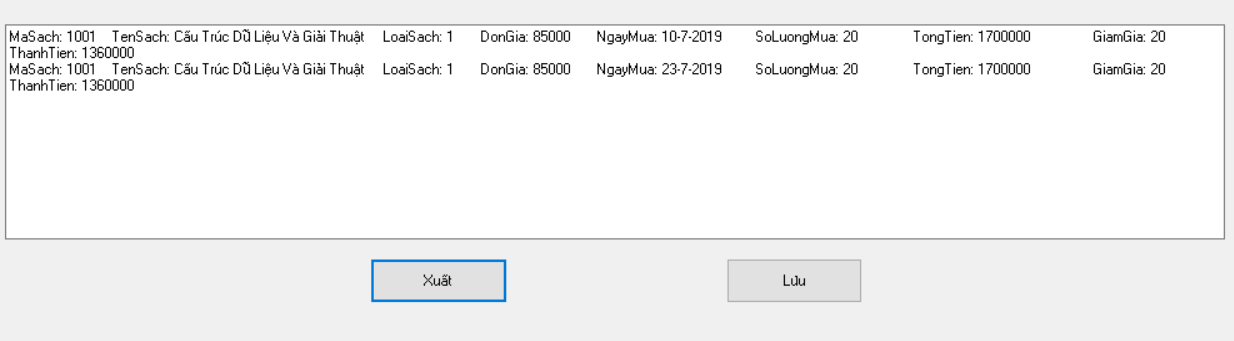
* **Thêm Sau :**
* **Xóa :**

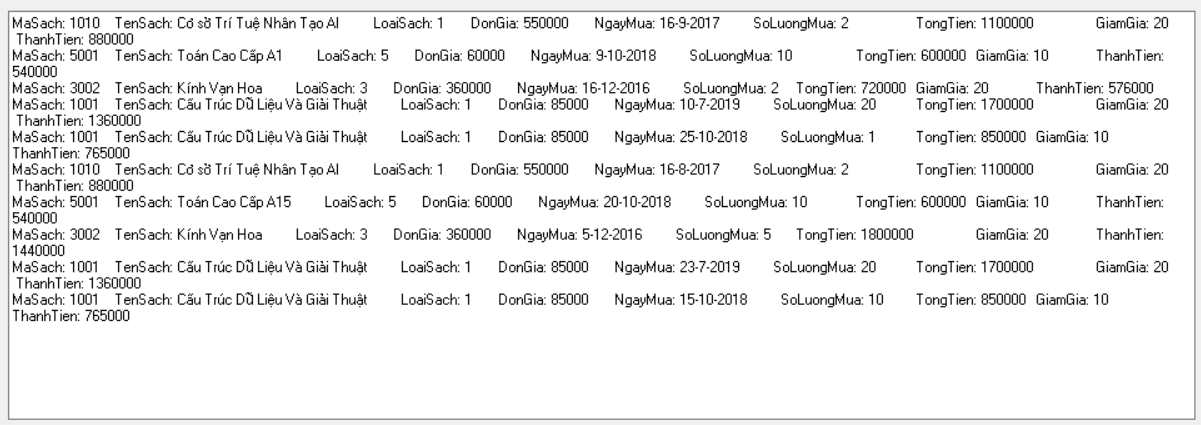


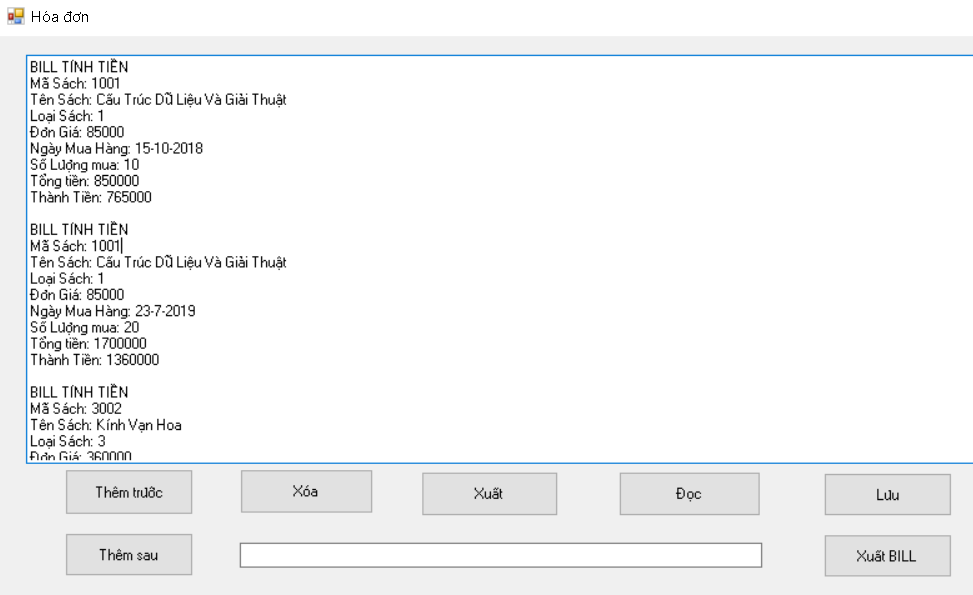
* Kết quả xóa :
* Button **Lưu** : tương tự như **Sach, LoaiSach**.
* Button **Doanh Thu** :
* Từ form Hóa Đơn, ta sẽ kiểm tra xem với từng cuốn sách, ta đã bán được tổng cộng bao nhiêu cuốn, đã thu được bao nhiêu tiền cụ thể như sau :

Ví dụ : Cuốn sách có mã 1001 đã bán được 4 lần, với tổng số lượng bán được là 51 cuốn và thu được 4.250.000 đồng.



* Button **Best Sell** : Từ form Hóa Đơn, ta sẽ tìm những lịch sử Hóa đơn có giá trị cao nhất như sau :
* Ví dụ :
* Như vậy, ta thấy : Hó đơn có giá trị nhất có giá trị là 1.360.000 Đồng.
* Button **Xuất Bill** : Ta sẽ xuất ra các hóa đơn theo dạng Bill với Format đã được cho trước như sau :



* Sau khi xuất ra với format dạng Bill, ta được :

# PHẦN KẾT LUẬN

## Kết quả đạt được

Sau quá trình thực hiện, đồ án “SỬ DỤNG CẤU TRÚC DỮ LIỆU ĐỂ XÂY DỰNG CHƯƠNG TRÌNH QUẢN LÝ BÁN SÁCH ”của nhóm đã hoàn thành được các mục tiêu:

+ Xây dựng một chương trình quản lý việc bán sách với các cách quản lý chuyên biệt tương đối đầy đủ, làm rõ được các yêu cầu đã cho trước.

+ Hiểu rõ được cách thức ứng dụng cấu trúc dữ liệu và giải thuật vào một bài tập cụ thể

+ Tăng cường khả năng tư duy và lập trình cho chúng em

## Kết luận

1. **Ưu điểm**

Sau khi hoàn thành đồ án, nhóm thực hiện đồ án tự nhận thấy đã hoàn thành tốt phần nào những yêu cầu cầu của đề tài đã đặt ra,và cả những mục tiêu mà nhóm chúng em đã tự đặt ra cho bản thân.

Nhờ có những sự phản hồi từ giáo viên hướng dẫn nhóm thực hiện đã nhận ra được lỗi sai nằm ở đâu và tìm cách sửa chữa cho đúng với yêu cầu bài toán đặt ra, cho phù hợp với các cấu trúc dữ liệu đã được áp dụng.

1. **Nhược điểm**

Đồ án còn chưa hoàn hảo. Còn một số nhược điểm về việc nhập xuất, thêm xóa sửa trong 1 số trường hợp đặc biệt khác (mặc dù đa số đều tốt). Chức năng Best Sell chưa được làm rõ hết mức có thể.

# TÀI LIỆU THAM KHẢO

<https://www.geeksforgeeks.org/linked-list-implementation-in-c-sharp/>

<https://www.geeksforgeeks.org/string-split-method-in-c-sharp-with-examples/>

<https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.linkedlist-1?view=netframework-4.8>

<https://www.dotnetperls.com/linkedlist>

<https://vankhangfet.github.io/2018-12-03-LinkList_C/>