

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI PHÂN HIỆU TẠI TP. HỒ CHÍ MINH

## BỘ MÔN CÔNG NGHỆ THÔNG TIN



**BÁO CÁO BÀI TẬP LỚN**

MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

# ĐỀ TÀI: QUẢN LÝ SỰ KIỆN

Giảng viên hướng dẫn: Trần Thị Dung

Sinh viên thực hiện:

-Nguyễn Hồ Hưng

* Dương Văn Hoàng– 6151071050

-Đặng Đình Sang– 6151071092

Lớp: CQ.61.CNTT

TP. Hồ Chí Minh, ngày 17 tháng 11 năm 2021

# 

# LỜI CẢM ƠN

Lời đầu tiên, nhóm xin cảm ơn đến nhà trường, quý thầy cô Bộ môn trường Đại học Giao thông Vận tải – Phân hiệu tại TP. Hồ Chí Minh đã tạo điều kiện cho nhóm có môi trường học tập và nghiên cứu thật tốt. Xin cảm ơn đến quý thầy, cô đã hết lòng quan tâm, dạy dỗ và trang bị cho nhóm những kiến thức và kinh nghiệm quý báu.

Nhóm xin trân trọng cảm ơn cô Trần Thị Dung đã trực tiếp giảng dạy môn Lập trình hướng đối tượng cũng như đã hướng dẫn nhóm trong quá trình nghiên cứu lý thuyết và tiến hành thực hiện đề tài này. Xin cảm ơn thầy đã giúp đỡ nhóm thực hiện tốt đề tài.

Xin gửi lời cảm ơn đến các thành viên trong nhóm, các bạn học đã cố gắng cùng nhau để hoàn thành báo cáo đề tài và góp ý để hoàn thiện bản báo cáo đề tài này.

Trong quá trình nghiên cứu, cũng như là trong quá trình làm bài báo cáo đề tài, nhóm khó tránh khỏi sai sót, rất mong quý thầy bỏ qua. Đồng thời do trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên bài báo cáo về mặt nội dung cũng như trình bày không thể tránh khỏi những thiếu sót, nhóm rất mong nhận được ý kiến đóng góp của thầy để nhóm học thêm được nhiều kinh nghiệm và sẽ hoàn thành tốt hơn bài những bài báo cáo sắp tới.

Cuối cùng em kính chúc quý thầy dồi dào sức khỏe và thành công trong sự nghiệp cao quý. Đồng kính chúc các quý thầy cô trong bộ môn, cùng các thành viên trong nhóm và các bạn học trong lớp luôn dồi dào sức khỏe, đạt được nhiều thành công tốt đẹp trong công việc.

*Nhóm xin chân thành cảm ơn!*

*TP. Hồ Chí Minh, ngày 15 tháng 11 năm 2021*

nhóm sinh viên thực hiện

**MỤC LỤC**

## [LỜI CẢM ƠN](#_bookmark0) 1

## I.Giới thiệu đề tài 3

## II.Nội dung lý thuyết 4

## 1.Tính chất hướng đối tượng 4

## 2.Cấu trúc dữ liệu sử dụng trong bài toán 6

## 3.Đồ họa.............................................................................................................12

## III.Giao diện và chứ năng của chương trình.............................................................18

## IV.Phụ lục.....................................................................................................................24

## 1.Link GitHub....................................................................................................24

## 2.Hướng dẫn cài đặt chương trình...................................................................25

## 3.Hướng dẫn sử dụng chương trình.................................................................25

1. **Giới thiệu đề tài.**

**1.Lí do chọn đề tài**

* Nhóm muốn lập trình một chương trình đơn giản , áp dụng những kiến thức của ngôn ngữ C++ đã được học trên lớp và tìm hiểu tại nhà tạo ra một chương trình quản lí sự kiện giúp người dùng có thể tra cứu tất cả các sự kiện , ngày lễ trong năm và quan trọng hơn là có thể tạo ra những sự kiện riêng của bản thân mình , từ những công việc nhỏ như thời khóa biểu hằng ngày như ( đi gặp đối tác , họp nhóm , hẹn đi chơi , xem phim ...) đến những ngày quan trọng ( sinh nhật , kỉ niệm 100 ngày yêu nhau , kỉ niệm 1 năm ngày cưới ...).
* Mục đích : Giúp người sử dụng tạo ra 1 chuỗi sự kiện hợp lí , không bị quên hay bỏ qua những sự kiện , ngày lễ quan trọng.

**2.Mô tả bài toán**

* Sử dụng lập trình hướng đối tượng , tạo ra 1 chương trình gồm 2 chức năng chính là xem ngày lễ và tạo sự kiện các thao tác của người dùng sẽ dược thiết kế dưới dạng MENU.

**+**  Xem ngày lễ sẽ gồm các chức năng như : Xem thời gian hiện tại ,

xem ngày lễ trong tháng.

+ Tạo sự kiện sẽ gồm các chức năng như : Thêm mới , hiển thị , sắp

xếp , tìm kiếm , xóa , lưu, xem danh sách đã lưu.

**3.Phân công công việc và thực hiện**

* Phân công công việc :

+ Nguyễn Hồ Hưng :lên ý tưởng đề tài, code khung chương trình , upload tài liệu tham khảo ,fix bug chương trình.

+ Dương Văn Hoàng : Code phần giao diện Menu chương trình , code chức năng, fix bug chương trình .

+ Đặng Đình Sang: Làm báo cáo bài tập lớn , code chức năng ,

fix bug chương trình .

* Thưc hiện : Do dich bệnh Covid 19 , nhóm em rất ít khi gặp nhau trực tiếp nên nhóm thảo luận và đưa ra phương án làm việc thông qua kênh trực tuyến là GitHud , Zalo và Discod .

1. **Nội dung cơ sở lý thuyết**

### **Tính chất hướng đối tượng**

* Tính đóng gói(Encapsulation) : là cách dùng để che dấu những tính chất xử lí bên trong của đối tượng , những đối tượng khác không thể tác động trực tiếp làm thay đổi trạng thái chỉ có thể tác động thông qua các method pubic của đối tượng đó.

Ví dụ trong bài :

* using namespace std;
* class DateTime{
* private:
* int day;
* int month;
* int year;
* int hour;
* int minute;
* int second;
* int ngayDaQua=0;
* int thangDaQua=0;
* int timThang;
* public:
* DateTime(){
* time\_t now = time(0);
* tm \*ltm = localtime(&now);
* day = ltm->tm\_mday;
* month = 1 + ltm->tm\_mon;
* year = 1900 + ltm->tm\_year;
* hour = ltm->tm\_hour;
* minute = ltm->tm\_min;
* second = ltm->tm\_sec;
* }
* void thoiGianHienTai();
* void ngayLeTrongThang1();
* void ngayLeTrongThang2();
* void ngayLeTrongThang3();
* void ngayLeTrongThang4();
* void ngayLeTrongThang5();
* void ngayLeTrongThang6();
* void ngayLeTrongThang7();
* void ngayLeTrongThang8();
* void ngayLeTrongThang9();
* void ngayLeTrongThang10();
* void ngayLeTrongThang11();
* void ngayLeTrongThang12();
* void xemNgayLeTrongThang();
* void xemNgayLeTrongNam();
* void xacDinhThuTrongTuan();
* void menuNgayLe();
* void ListMenuNgayLe();
* };
* **Tính kế thừa (Inheritance):** Là kỹ thuật cho phép kế thừa lại những tính năng mà một đối tượng khác đã có, giúp tránh việc code lặp dư thừa mà chỉ xử lý công việc tương tự.

Ví dụ trong bài :

* class HolidayInMonth : public DateTime{
* public:
* void xemNgayLeTrongThang(){
* Frames();
* FramesInput();
* **Tính trừu tượng(Abstraction):** Là phương pháp trừu tượng hóa định nghĩa lên những hành động, tính chất của loại đối tượng nào đó cần phải có

+ Sử dụng các biến có tên liên quan đến sự kiện như tên ngày tháng năm để người tạo chương trình hay người xem code dễ hình dung biến đó đang chứa giá trị gì và làm gì:

Ví dụ trong bài :

* struct Event
* {
* string TieuDe;
* int ngay, thang, nam;
* int gio, phut;
* string NoiDung;
* };
* **Tính đa hình (Polymorphism ):** Là một đối tượng thuộc các lớp khác nhau có thể hiểu cùng một thông điệp theo cách khác nhau.

1. **Cấu trúc dữ liệu và giải thuật dùng trong bài toán**

* **Trong bài sử dụng các cấu trúc trúc dữ liệu và giải thuật là :**

**2.1 Danh sách liên kết đơn:**

+Danh sách liên kết đơn (Single Linked List) là một cấu trúc dữ liệu động, nó là một danh sách mà mỗi phần tử đều liên kết với phần tử đúng sau nó trong danh sách. Mỗi phần tử (được gọi là một node hay nút) trong danh sách liên kết đơn là một cấu trúc có hai thành phần:

+Thành phần dữ liệu: lưu thông tin về bản thân phần tử đó.

+Thành phần liên kết: lưu địa chỉ phần tử đứng sau trong danh sách, nếu phần tử đó là phần tử cuối cùng thì thành phần này bằng NULL.

**Ví dụ trong chương trình:**

* // Tạo cấu trúc danh sách LK đơn
* struct SingleList
* {
* Node \*pHead;
* };
* // Khởi tạo danh sách LK đơn
* void Initialize(SingleList \*&list)
* {
* list=new SingleList;
* list->pHead=NULL;
* }
* // Tạo một Node mới
* Node \*CreateNode(Event \*ev)
* {
* Node \*pNode=new Node;
* if(pNode!=NULL)
* {
* pNode->data=ev;
* pNode->pNext=NULL;
* }
* else
* {
* cout<<"cap phat bo nho that bai!!!";
* }
* return pNode;
* }
* // Thêm một sự kiện vào danh sách LK đơn
* Event \*NhapEvent()
* {
* Event \*ev=new Event;
* Frames();
* FramesInput();
* lineTitle();
* gotoxy(54, 12); cout << "ADD EVENT";
* fflush(stdin);
* gotoxy(45, 14); cout << "Tieu de: ";
* getline(cin, ev->TieuDe);
* settingDay(\*ev);
* settingHour(\*ev);
* gotoxy(58, 24); cout << " ";
* fflush(stdin);
* gotoxy(44, 26); cout << "Noi dung: ";
* getline(cin, ev->NoiDung);
* gotoxy(70, 29); cout << "Nhan enter de hoan tat!";
* getch();
* return ev;
* }
* // Thêm một sự kiện vào cuối danh sách LK đơn
* void InsertLast(SingleList \*&list,Event \*ev)
* {
* Node \*pNode=CreateNode(ev);
* if(list->pHead==NULL)
* {
* list->pHead=pNode;
* }
* else
* {
* Node \*pTmp=list->pHead;
* while(pTmp->pNext!=NULL)
* {
* pTmp=pTmp->pNext;
* }
* pTmp->pNext=pNode;
* }
* }

**2.2 Thuật toán sắp xếp nổi bọt (Bubble Sort) : sẽ tiến hành việc so sánh các cặp phần tử liền kề nhau và tráo đổi vị trí của nó cho đúng thứ tự mà người dùng mong muốn.**

**Ví dụ trong chương trình :**

* // Hàm sắp xếp theo thời gian (sử dụng thuật toán sắp xếp nổi bọt)
* void swap( Event \*&a, Event \*&b ){
* Event \*tmp=a;
* a=b;
* b=tmp;
* }
* void SortList1(SingleList \*&list) {
* Node \*pTmp = list->pHead;
* if(pTmp == NULL) {
* cout << "Danh sach rong" << endl;
* return;
* }
* while(pTmp->pNext != NULL) {
* for(Node \*q = pTmp->pNext ; q != NULL ; q = q->pNext) {
* if(pTmp->data->nam > q->data->nam) {
* swap(pTmp->data, q->data);
* }
* else if(pTmp->data->nam == q->data->nam) {
* if(pTmp->data->thang > q->data->thang) {
* swap(pTmp->data, q->data);
* }
* else if(pTmp->data->thang == q->data->thang) {
* if(pTmp->data->ngay > q->data->ngay) {
* swap(pTmp->data, q->data);
* }
* else if(pTmp->data->ngay == q->data->ngay) {
* if(pTmp->data->gio > q->data->gio) {
* swap(pTmp->data, q->data);
* }
* else if(pTmp->data->gio == q->data->gio) {
* if(pTmp->data->phut > q->data->phut) {
* swap(pTmp->data, q->data);
* }
* else if(pTmp->data->phut == q->data->phut) {
* if(pTmp->data->gio > q->data->gio) {
* swap(pTmp->data, q->data);
* }
* }
* }
* }
* }
* }
* }
* pTmp = pTmp->pNext;
* }
* }

**2.3 Thụât toán tìm kiếm nhị phân(Binary Search):**thực hiện tìm kiếm một mảng đã sắp xếp bằng cách liên tục chia các khoảng tìm kiếm thành 1 nửa. Bắt đầu với một khoảng từ phần tử đầu mảng, tới cuối mảng. Nếu giá trị của phần tử cần tìm nhỏ hơn giá trị của phần từ nằm ở giữa khoảng thì thu hẹp phạm vi tìm kiếm từ đầu mảng tới giửa mảng và nguợc lại. Cứ thế tiếp tục chia phạm vi thành các nửa cho đến khi tìm thấy hoặc đã duyệt hết.

**Ví dụ trong chương trình :**

* // Tìm kiếm theo tiêu đề (sử dụng thuật toán tìm kiếm nhị phân)
* void SearchByTitleEvent(SingleList \*list)
* {
* string title;
* Frames();
* FramesInput();
* gotoxy(54, 12);
* fflush(stdin);
* cout << "Tieu de: ";
* getline(cin, title);
* Node \*pTmp=list->pHead;
* if(pTmp->data->TieuDe == title)
* {
* Event \*ev=pTmp->data;
* gotoxy(44, 14);
* cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";
* gotoxy(44, 16);
* cout << "Tieu de: " << ev->TieuDe;
* gotoxy(44, 17);
* cout << "Thoi gian: " << ev->gio << ":" << ev->phut << ", " << ev->ngay << "/" << ev->thang << "/" << ev->nam;
* gotoxy(44, 18);
* cout << "Noi dung: " << ev->NoiDung;
* gotoxy(44, 19);
* cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";
* }
* else{
* modalLook();
* }
* pTmp=pTmp->pNext;
* }

1. **Đồ họa – Ví dụ**

* **Trong chương trình phần đồ họa và giao diện chủ yếu nhóm dùng những thư viện có sẵn để định dạng màn hình console , thay đổi màu ,chữ kẻ khung , xác định vị trí của các chức năng , viết tiêu đề và tên của chương trình , ngoài ra nhóm có tự code thêm một số chức năng như điều chình mũi tên để chọn chức năng .**
  1. **Thư viện windows.h:**

**Ví dụ trong chương trình:**

**+Định dạng kích cỡ của cmd :**

* // hàm thay đổi kích cỡ của khung cmd với tham số truyền vào là chiều cao, chiều rộng.
* void resizeConsole(int width, int height)
* {
* HWND console = GetConsoleWindow();
* RECT r;
* GetWindowRect(console, &r);
* MoveWindow(console, r.left, r.top, width, height, TRUE);
* }

**+Hàm định dạng màu chữ và fontsize chữ của chương trình:**

* void SetColor(int backgound\_color, int text\_color)
* {
* HANDLE hStdout = GetStdHandle(STD\_OUTPUT\_HANDLE);
* int color\_code = backgound\_color \* 16 + text\_color;
* SetConsoleTextAttribute(hStdout, color\_code);
* }
* void setFontSize(int FontSize)
* {
* CONSOLE\_FONT\_INFOEX info = {0};
* info.cbSize = sizeof(info);
* info.dwFontSize.Y = FontSize; // leave X as zero
* info.FontWeight = FW\_NORMAL;
* SetCurrentConsoleFontEx(GetStdHandle(STD\_OUTPUT\_HANDLE), NULL, &info);
* }

**+Hàm viết tiêu đề EVENT của chương trình:**

* void e1(){
* // hàm viết chữ E
* gotoxy(57, 3);
* cout << char(186) << char(186) << char(186) << char(186) << char(186) << char(186) << char(186) << char(186) << char(186) << "  " ;
* gotoxy(57, 4);
* cout << char(186) << char(186) << "         " ;
* gotoxy(57, 5);
* cout << char(186) << char(186) << "         " ;
* gotoxy(57, 6);
* cout << char(186) << char(186) << char(186) << char(186) << char(186) << char(186) << char(186) ;
* gotoxy(57, 7);
* cout << char(186) << char(186) << "         " ;
* gotoxy(57, 8);
* cout << char(186) << char(186) << "         " ;
* gotoxy(57, 9);
* cout << char(186) << char(186) << char(186) << char(186) << char(186) << char(186) << char(186) << char(186) << char(186) ;
* }
* .
* .
* .
* void t(){
* gotoxy(80, 3);
* cout << char(186) << char(186) << char(186) << char(186) << char(186) << char(186) << char(186) << char(186) << char(186) << char(186) << char(186) << char(186);
* gotoxy(80, 4);
* cout << "     " << char(186) << char(186);
* gotoxy(80, 5);
* cout << "     " << char(186) << char(186);
* gotoxy(80, 6);
* cout << "     " << char(186) << char(186);
* gotoxy(80, 7);
* cout << "     " << char(186) << char(186);
* gotoxy(80, 8);
* cout << "     " << char(186) << char(186);
* gotoxy(80, 9);
* cout << "     " << char(186) << char(186);
* }
* void Text(){
* e(); v(); e1(); n(); t();
* }

**+ Khung chức năng của chương trình:**

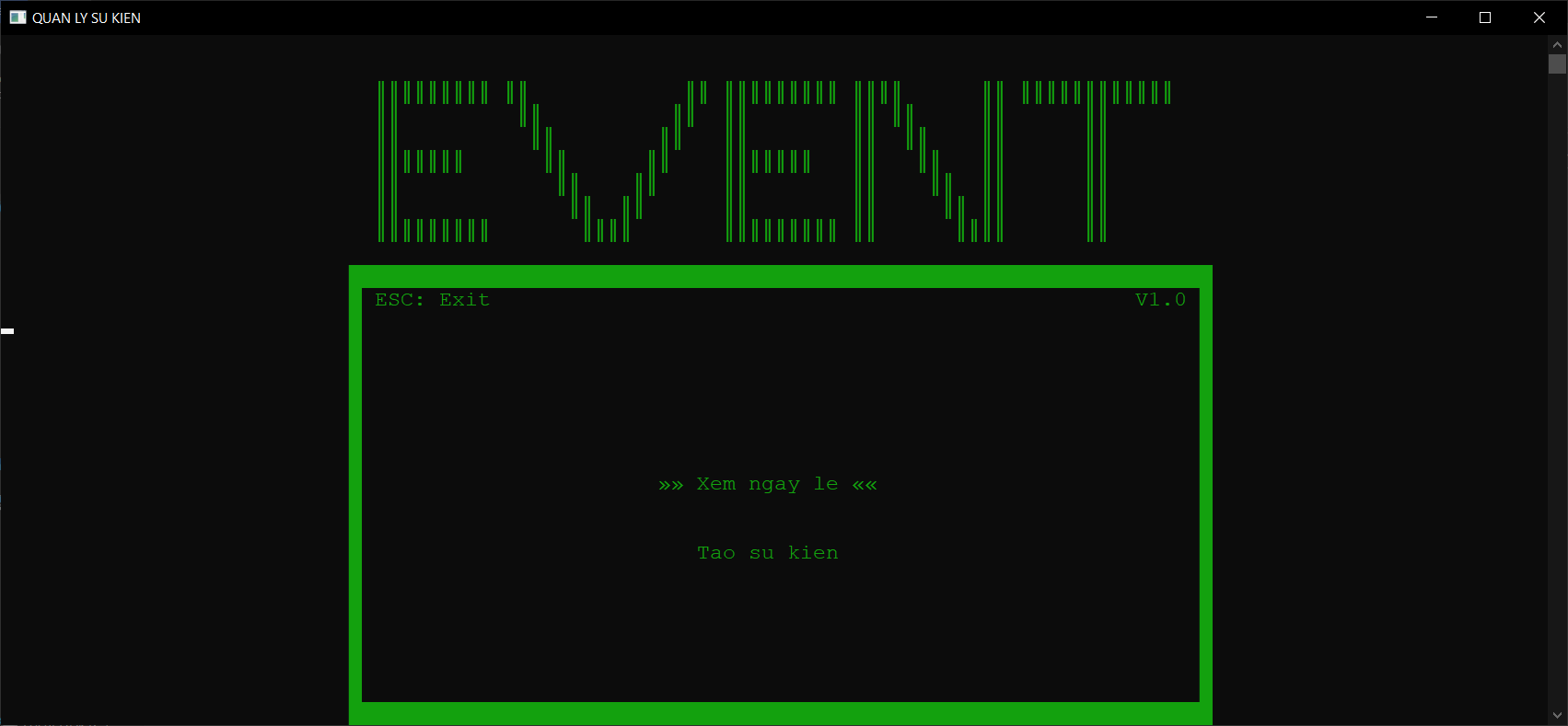
* void FramesInput(){
* // khung input
* gotoxy(45, 11); cout << char(218);
* gotoxy(76, 11); cout << char(191);
* gotoxy(45, 12); cout << char(179);
* gotoxy(76, 12); cout << char(179);
* gotoxy(45, 13); cout << char(192);
* for(int i=46 ; i < 76 ; i++){
* gotoxy(i, 13);
* cout << char(196);
* }
* gotoxy(76, 13); cout << char(217);
* }
  1. **Một số giao diện giao diện tự code :**

**+ Sử dụng mũi tên lên xuống để chọn và xem chức năng của chương trình**

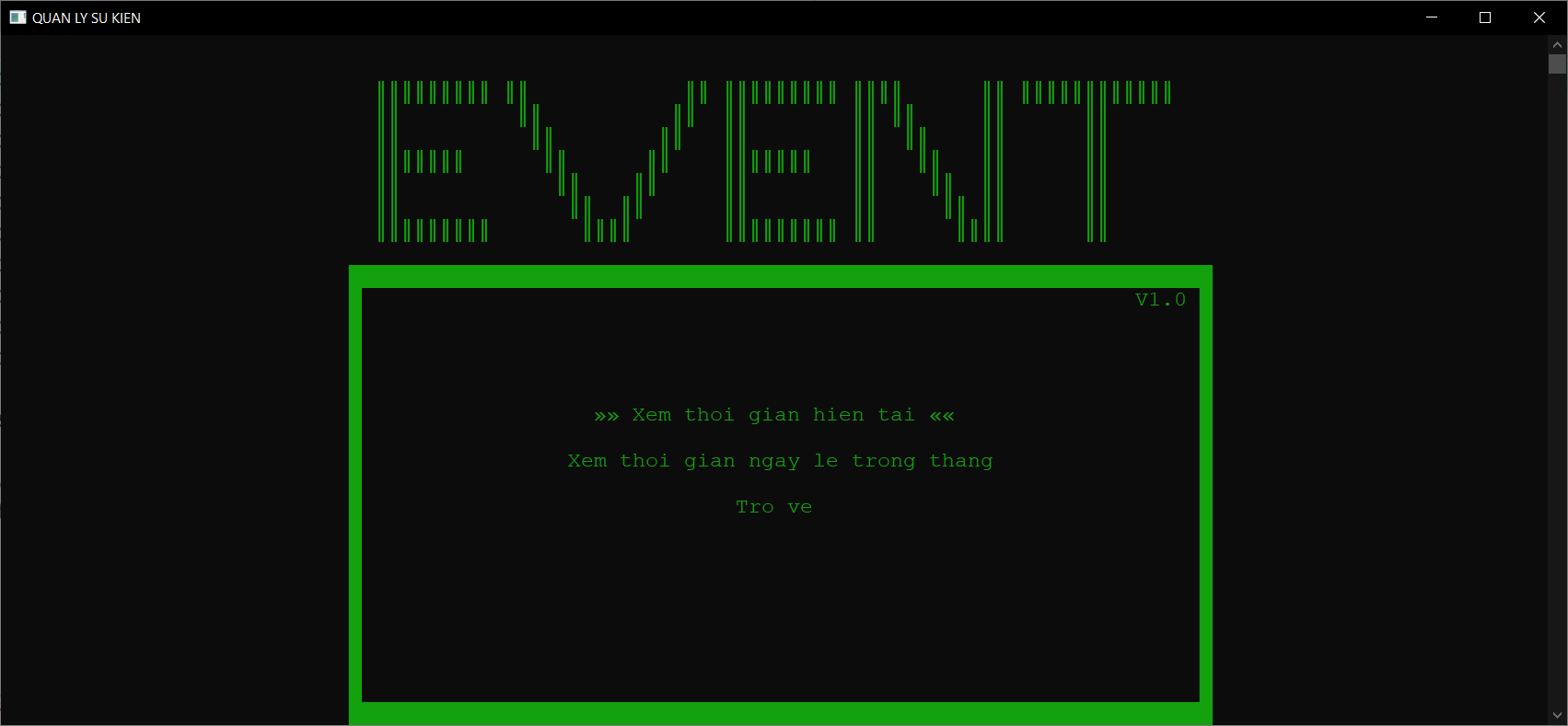
**Ví dụ trong chương trình :**

* int t = \_getch();
* if (t == 13)
* {
* switch (cn)
* {
* case 1:
* {
* time.ListMenuNgayLe();
* break;
* }
* case 2:
* {
* system("cls");
* Frames();
* listMenuEvent();
* break;
* }
* }
* }
* else if (t == 27)
* {
* system("cls");
* exit(0);
* }
* else if (t == 80)
* {
* cn++;
* }
* else if (t == 72)
* {
* cn--;
* }
* if (cn > 2)
* {
* cn = 1;
* }
* else if (cn < 1)
* {
* cn = 2;
* }
* }
* while (1);
* }

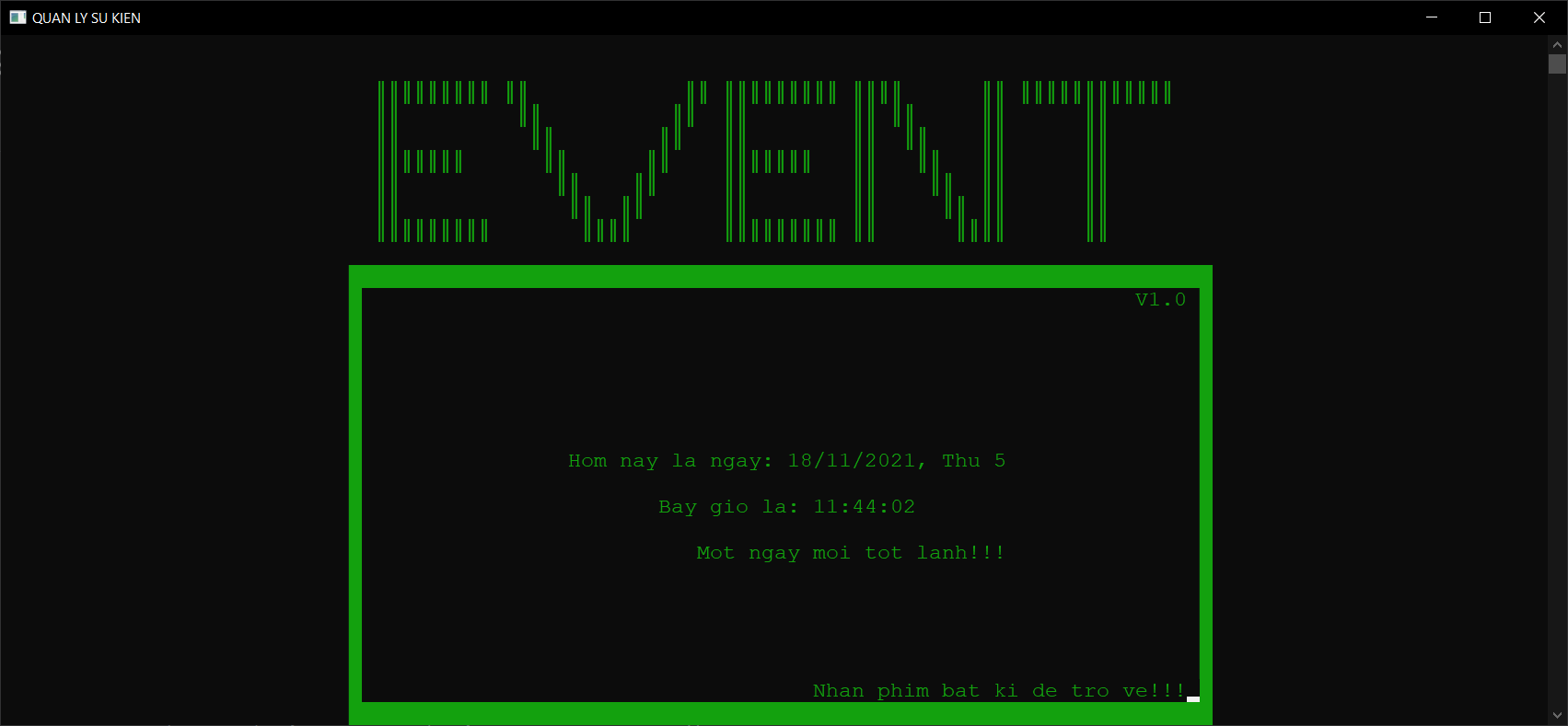
1. **Giao diện chương trình**
2. Giao diện MENU chương trình:



2.Chức năng xem ngày lễ :



* 1. Xem thời gian hiện tại :



* 1. Xem thời gian ngày lễ trong tháng :



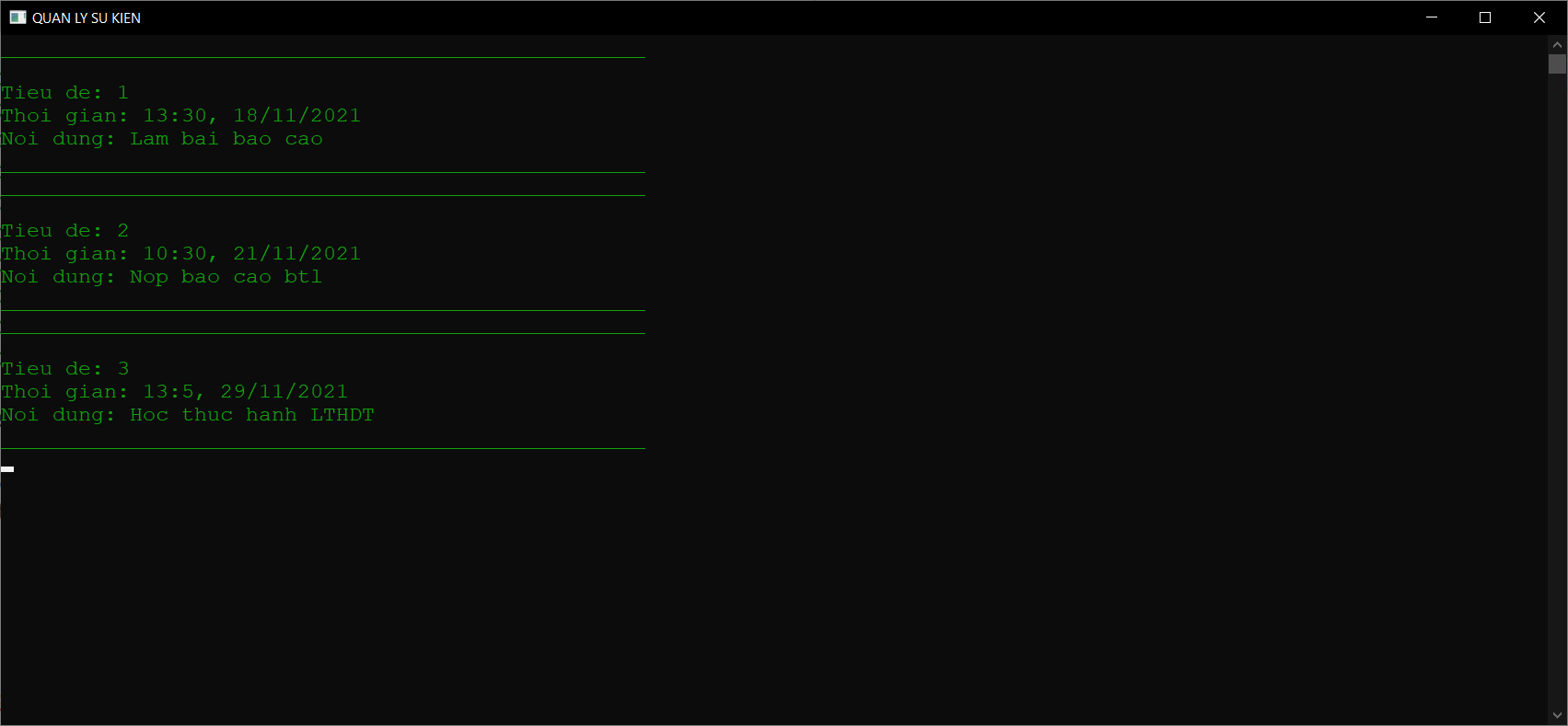
3. Chức năng tạo sự kiện :



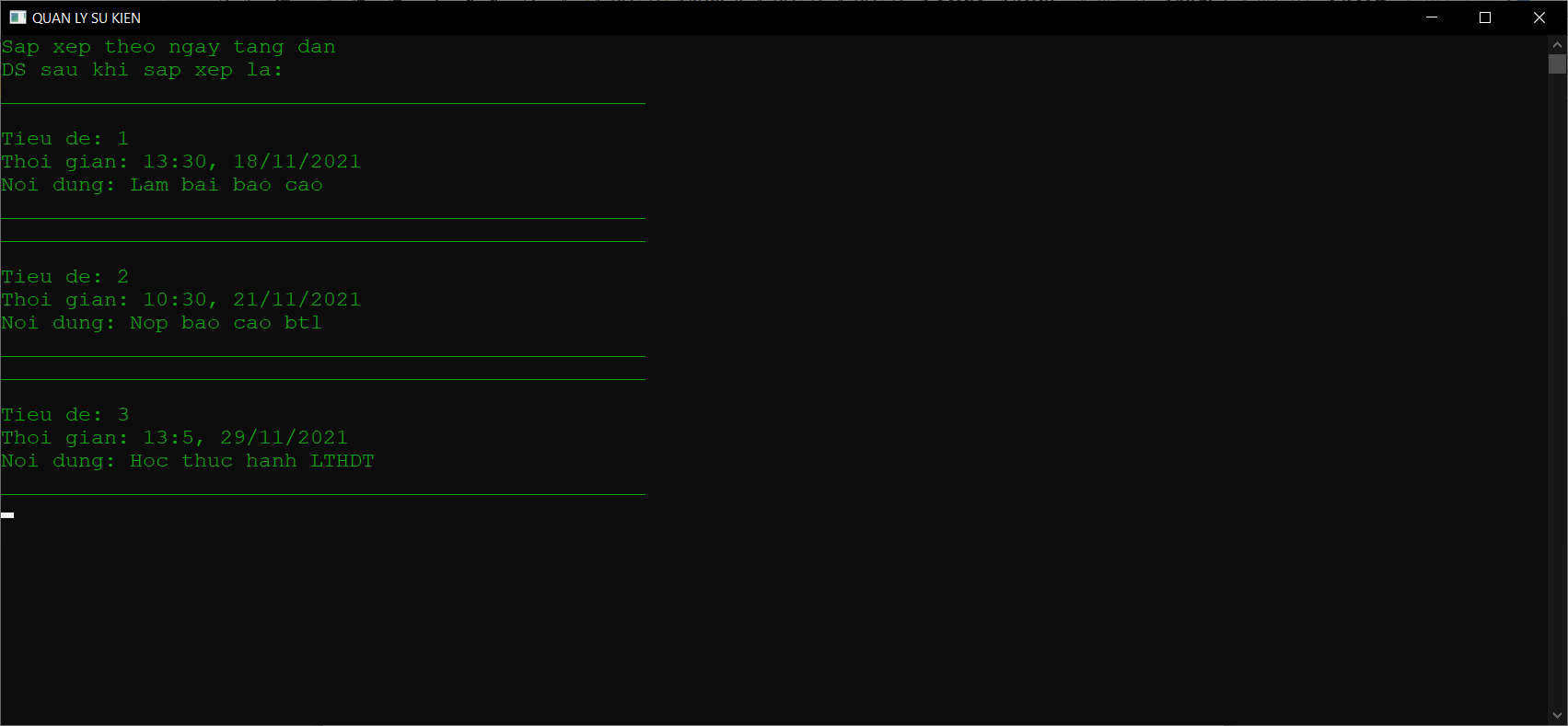
* 1. Thêm mới :



* 1. Hiển thị:



* 1. Sắp xếp theo ngày tăng dần :

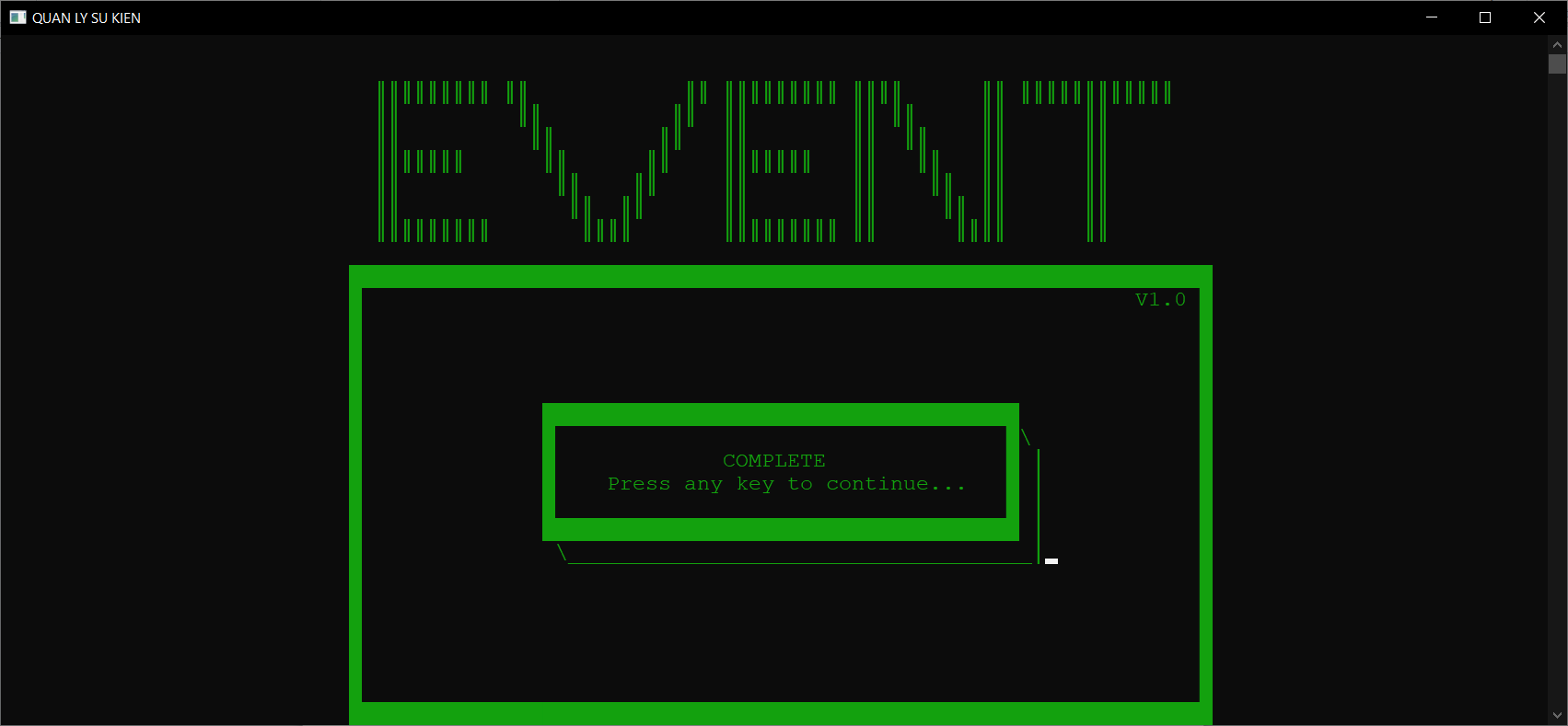


* 1. Tìm kiếm theo tiêu đề :

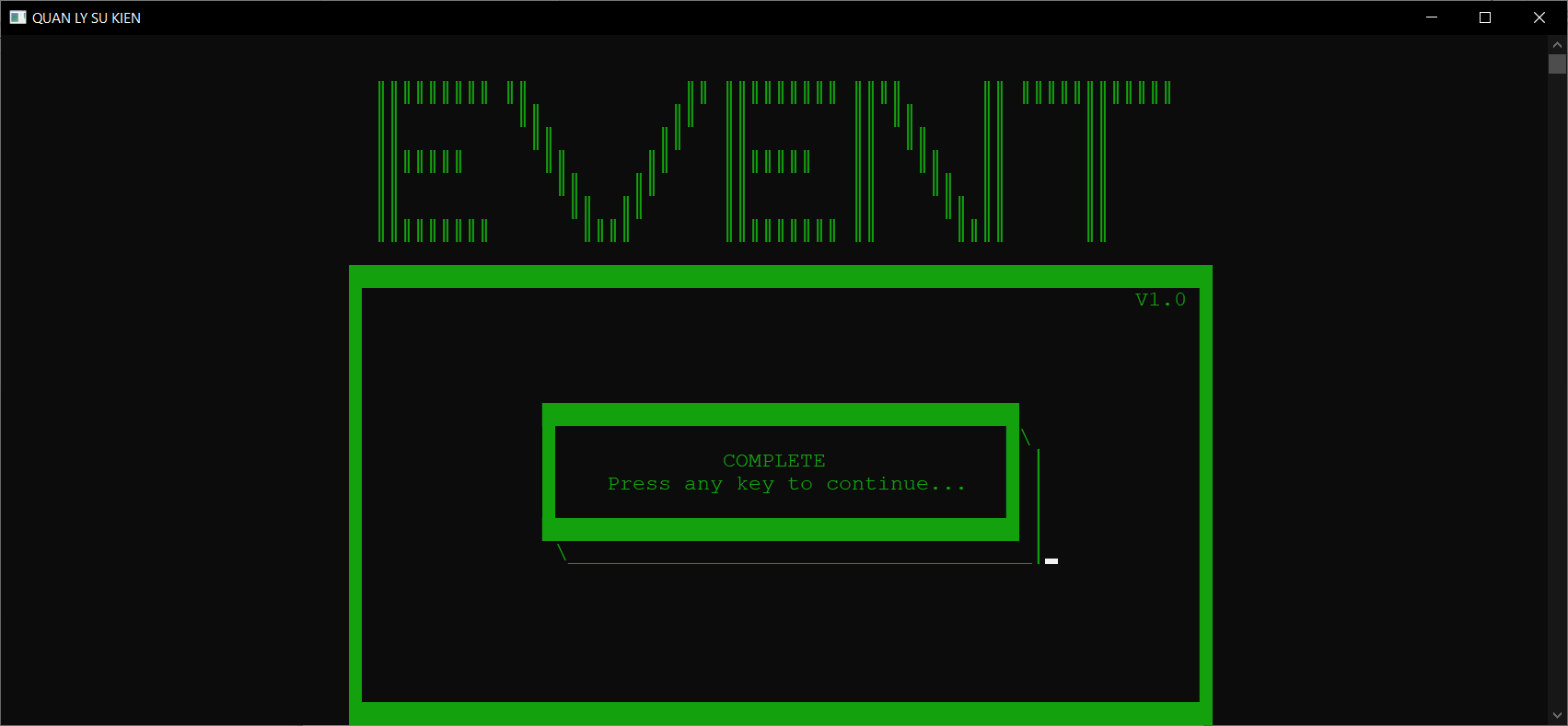


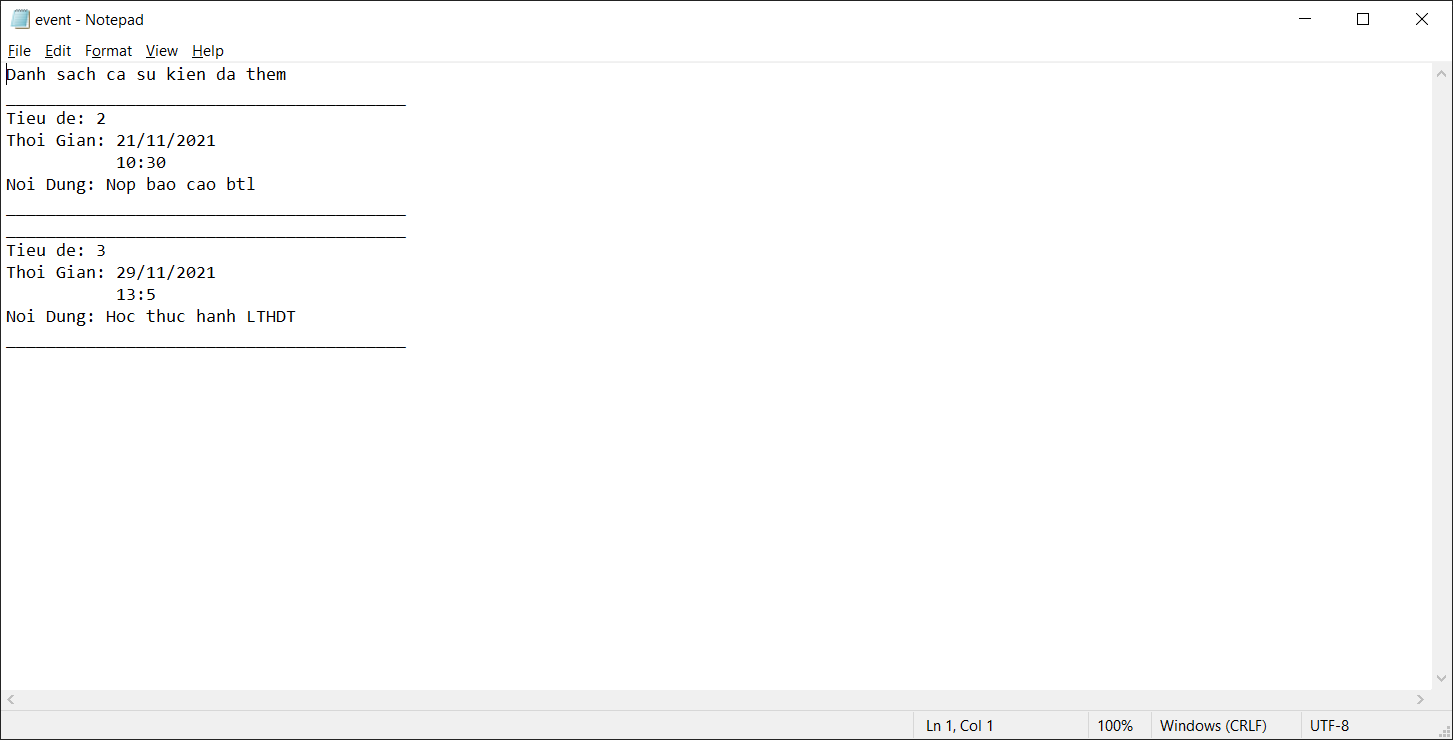
* 1. Xóa sự kiện :



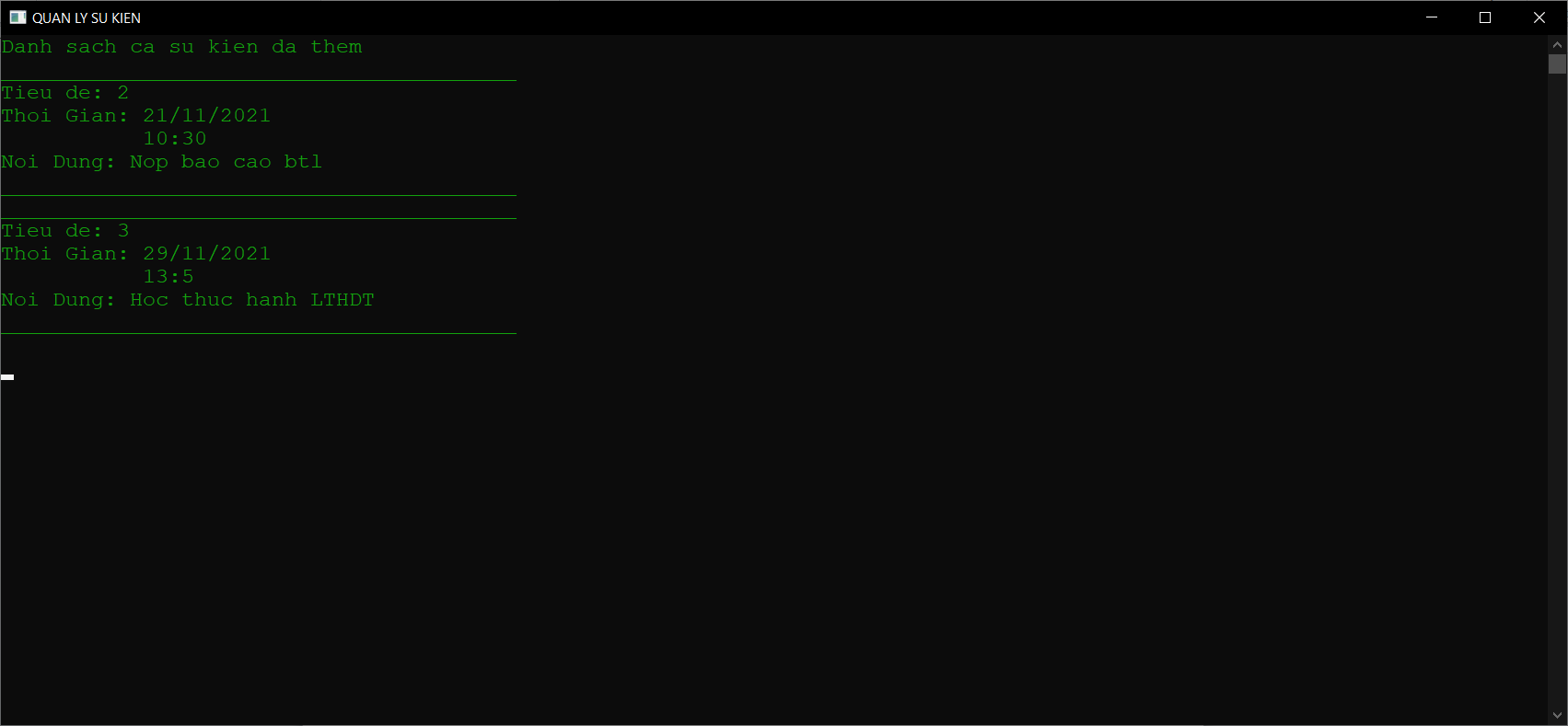


* 1. Lưu file sự kiện đã lưu:





* 1. Xem lại các sự kiện đã lưu:



1. **Phụ lục**

1.Link github : <https://github.com/hung55410/BTL_LapTrinhHuongDoiTuong>

2. Hướng dẫn cài đặt chương trình:

* B1: Cài đặt 1 chương trình Dev C++

+ <https://sourceforge.net/projects/orwelldevcpp/>

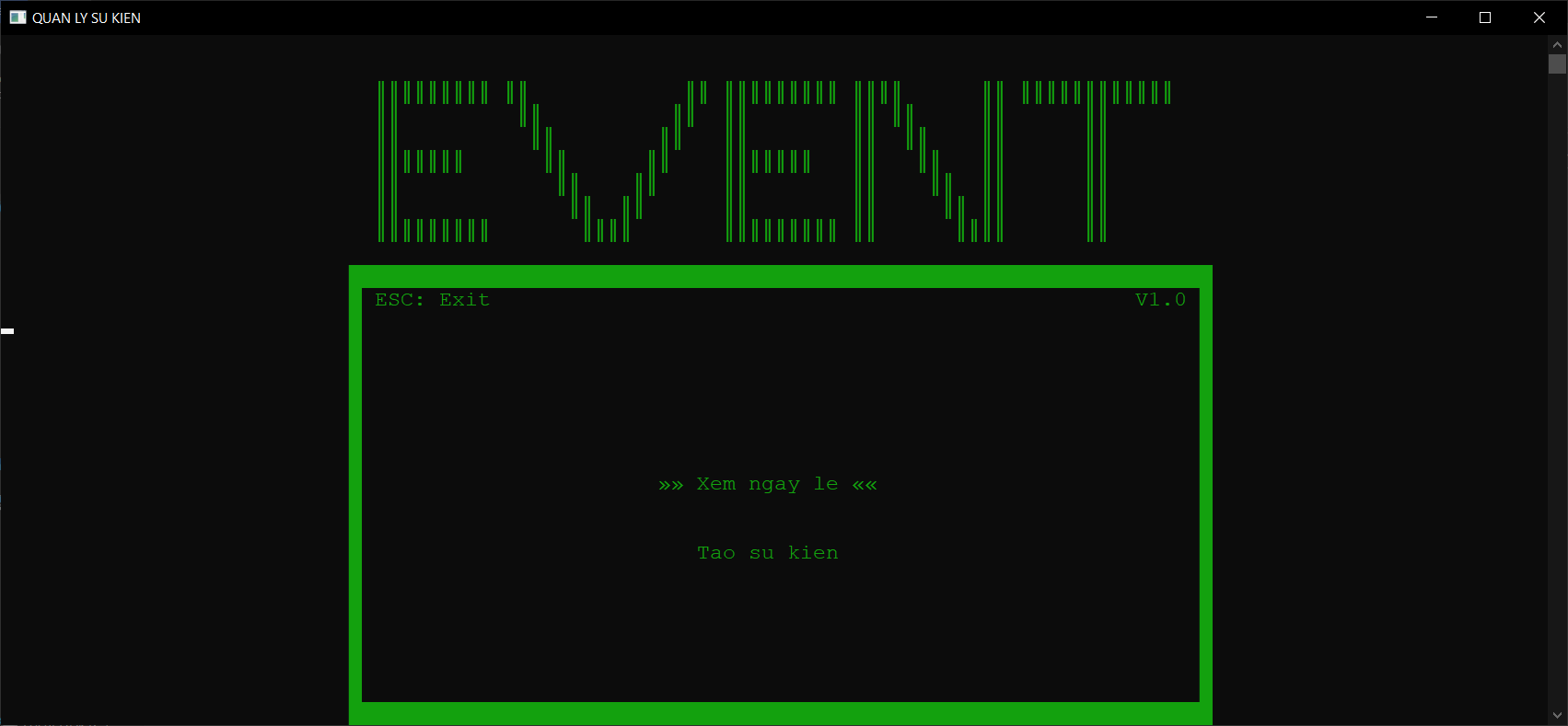
* B2 : Truy cập đường link google drive tải file bai tap lon nhom 16.rar và giải nén

+ <https://drive.google.com/file/d/1yb6rr6AhHRjmK0OV2aY6k9MvES7YBmc1/view?usp=sharing>

* B3: Sau khi giải nén file , vào Dev C++ chạy file main.cpp để chạy chương trình.

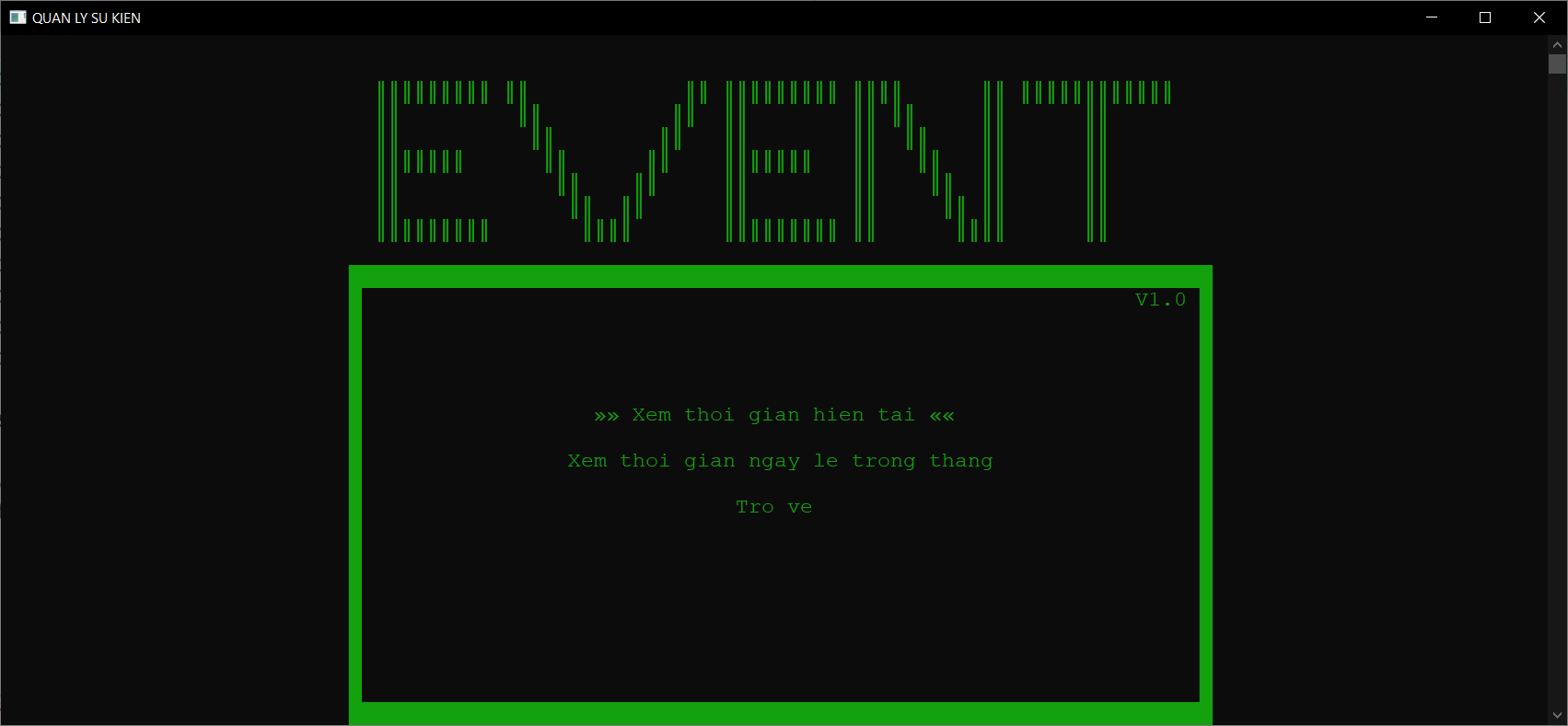
3.Hướng dẫn sử dụng chương trình :

3.1 Khi chạy chương trình sẽ hiện ra MENU và 2 tùy chọn là xem ngày lễ và tạo sự kiện sử dụng mũi tên lên xuống để chọn rồi nhấn enter để chọn :

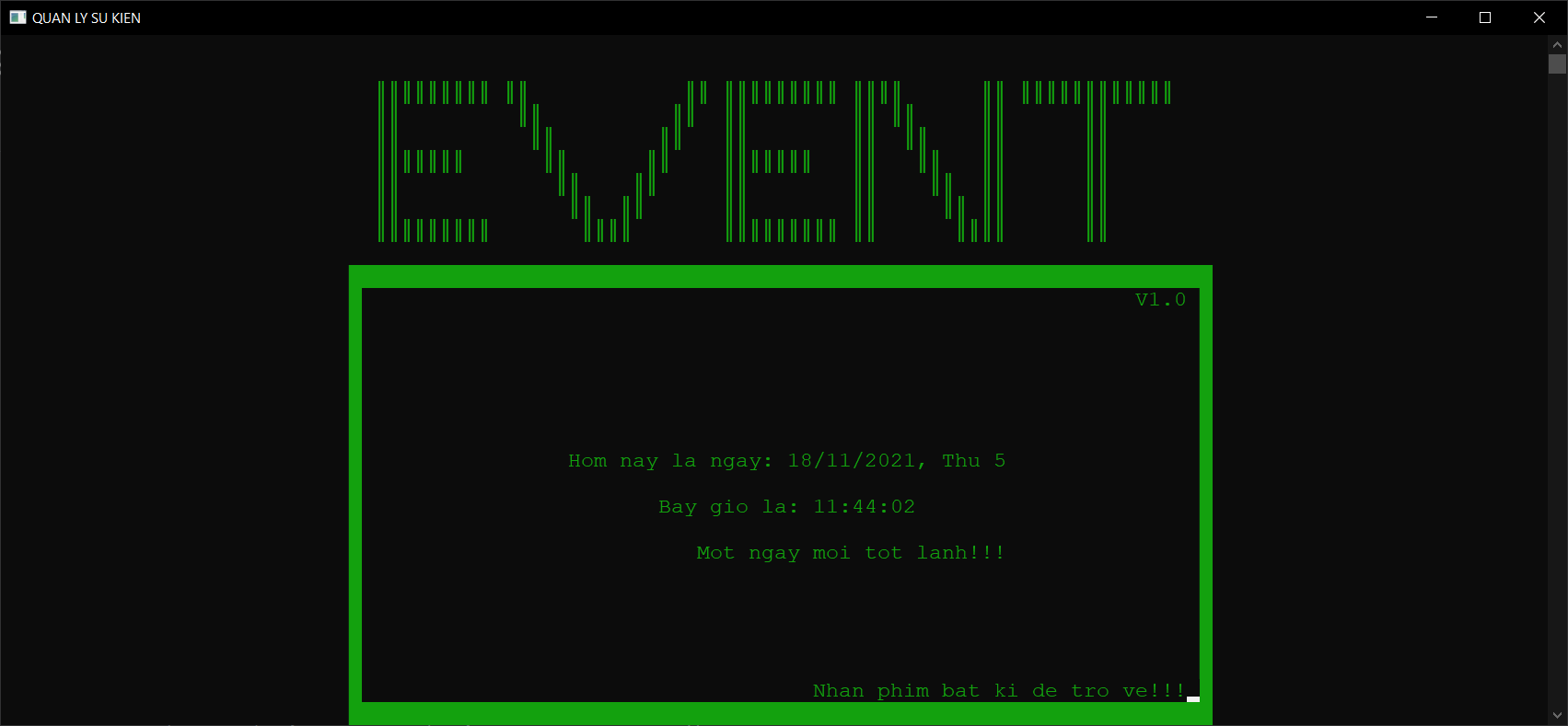


3.2 Nếu chọn chức năng xem ngày lễ màn hình sẽ hiện ra 2 chức năng là xem

thời gian hiện tại , xem thời gian ngày lễ trong tháng và trở về nhấn enter để chọn chức năng :



3.2.1 Chọn xem thời gian ở hiện tại sẽ hiển thị thứ ,ngày, tháng và giờ ở hiện tại :



3.2.2 Nếu chọn Xem thời gian ngày lễ trong tháng thì nhập tháng muốn xem ngày lễ rồi nhấn enter :



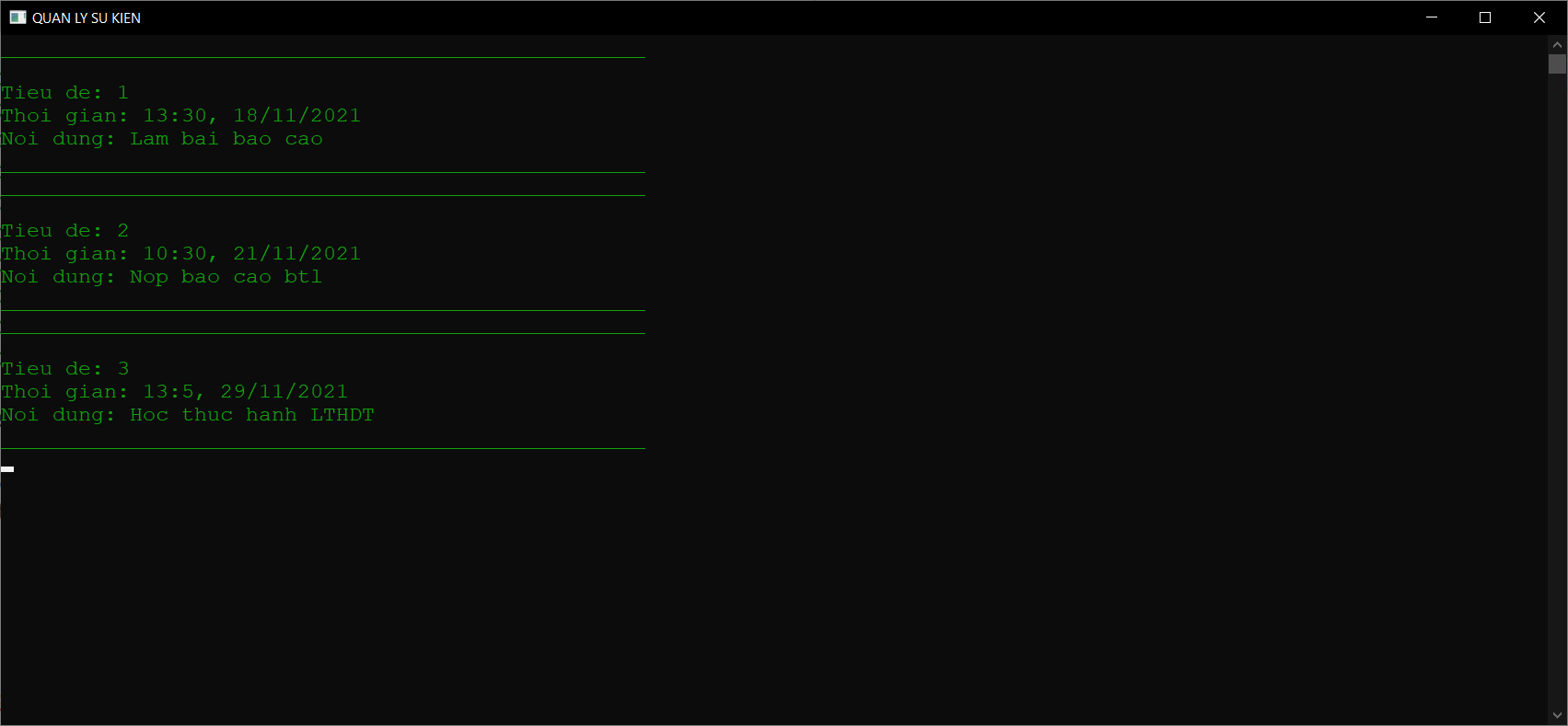
3.3. Khi chọn chức năng tạo sự kiện gồm có 8 phím chức năng sử dụng mũi tên lên xuống và chọn nhấn enter :



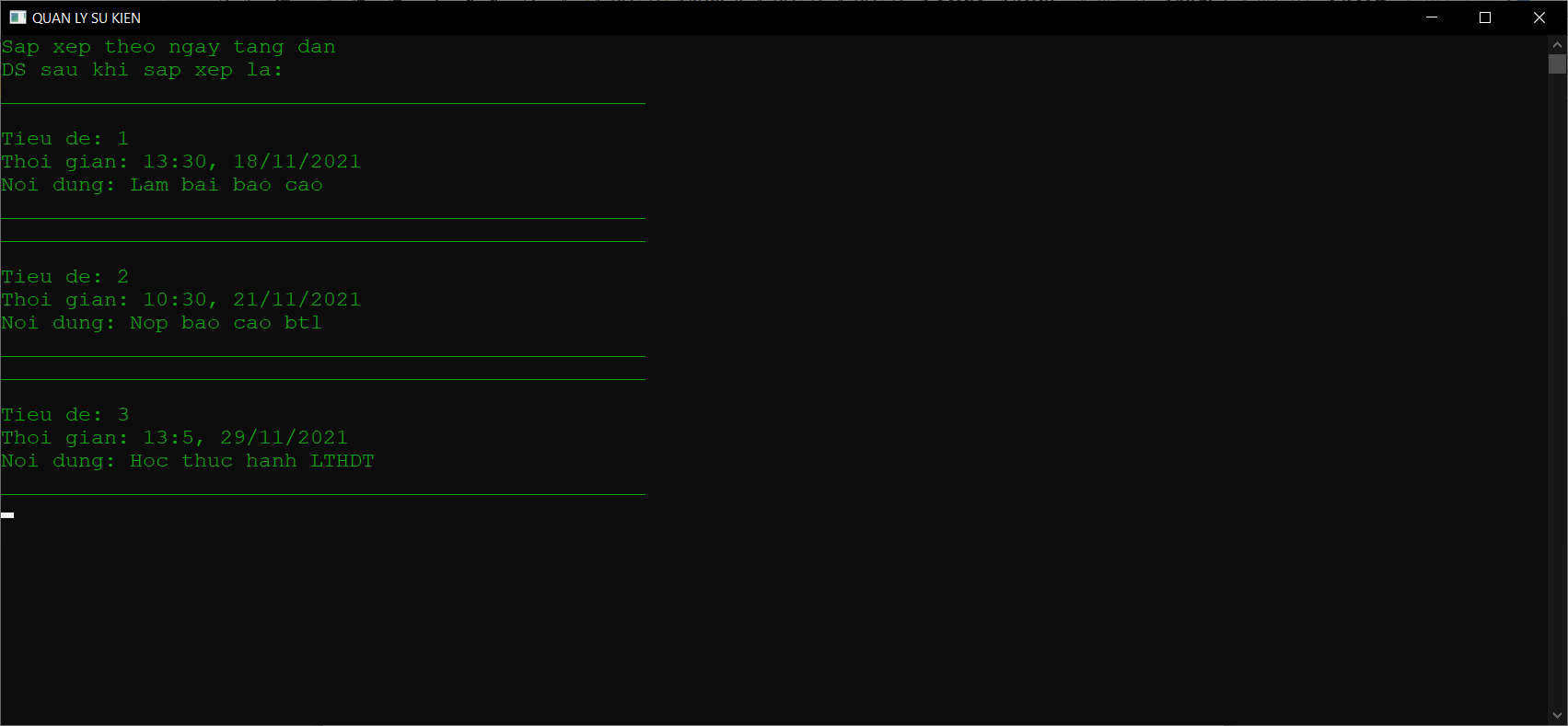
3.3.1 Chức năng thêm mới giúp nhập từ bàm phím 1 sự kiện vào danh sách lưu ý nhập đúng quy tắc của thời gian dể không phải nhập lại nhiều lần :



3.3.2 Khi chọn chức năng hiển thị chương trình sẽ xuất ra danh sách sự kiện đã nhập vào :



3.3.3 Muốn sắp xếp ngày theo tứ tự tăng dần thì chọn chức năng sắp xếp màn hình sẽ hiển thị thứ tự sự kiện tăng dần :

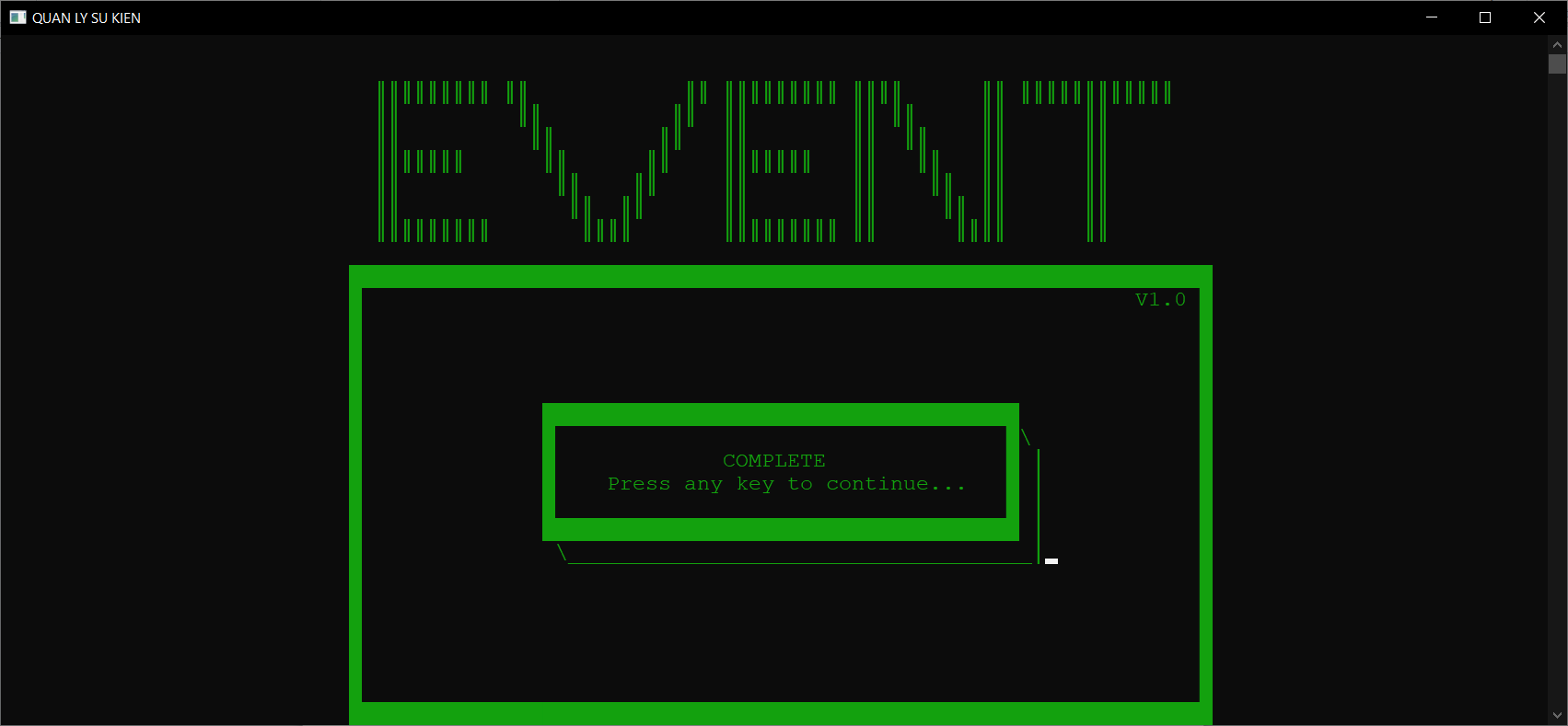


3.3.4 Muốn tìm sự kiện theo tiêu đề chọn chức năng tìm kiếm lưu ý nhập đúng tiêu đề để không phải nhập nhiều lần :

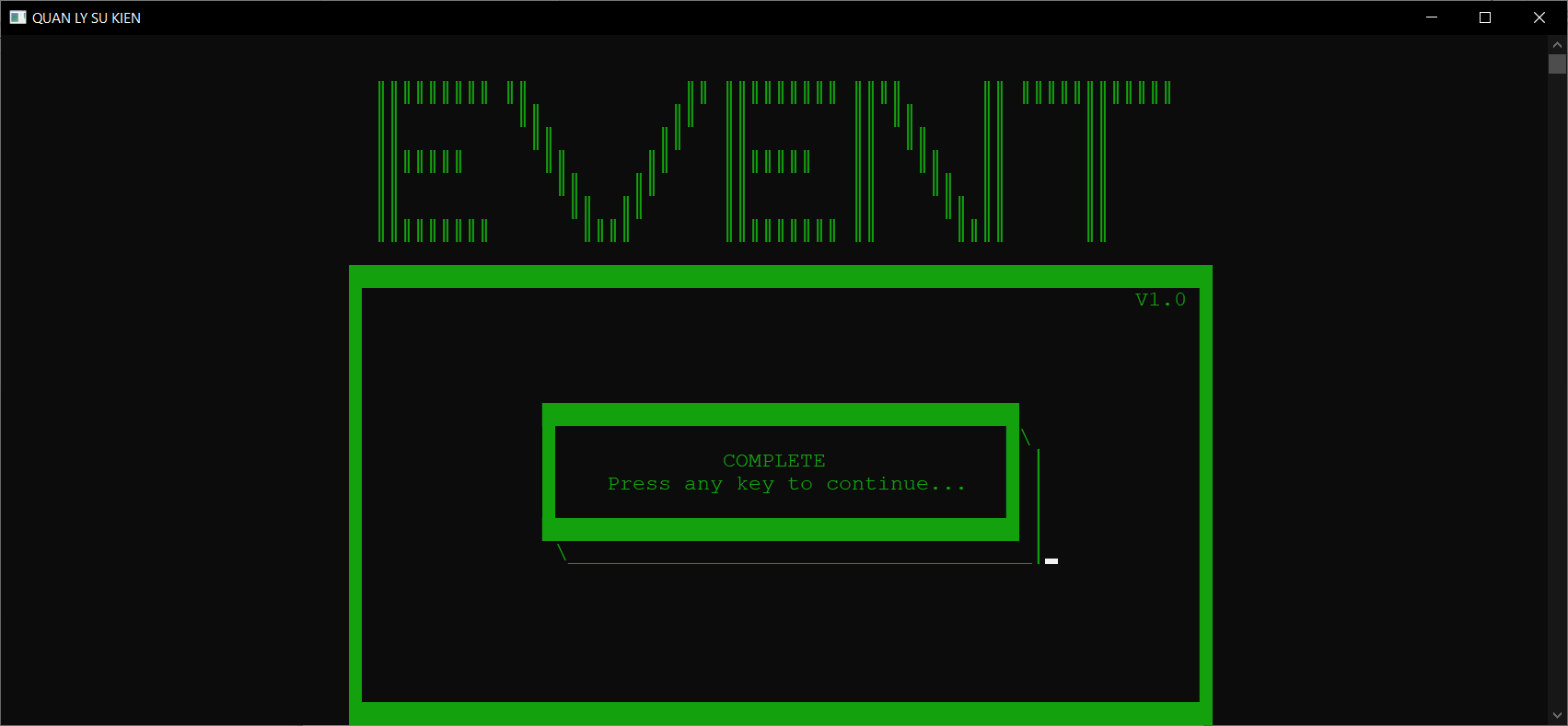


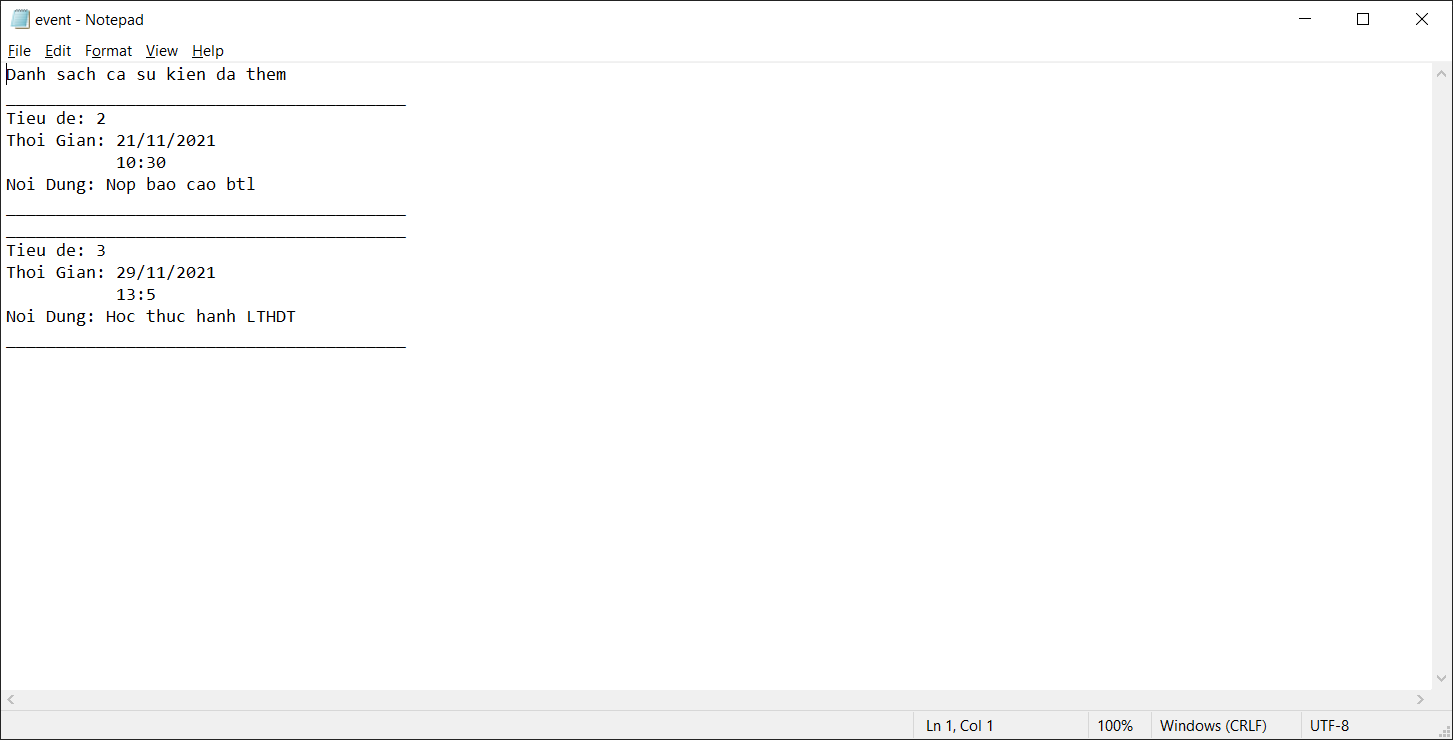
3.3.5 Nếu muốn xóa sự kiện , nhấn vào DELETE xong nhập tiêu đề sự kiện cần xóa rồi enter , lưu ý ghi đúng tiêu đề dể không phải nhập nhiều lần :



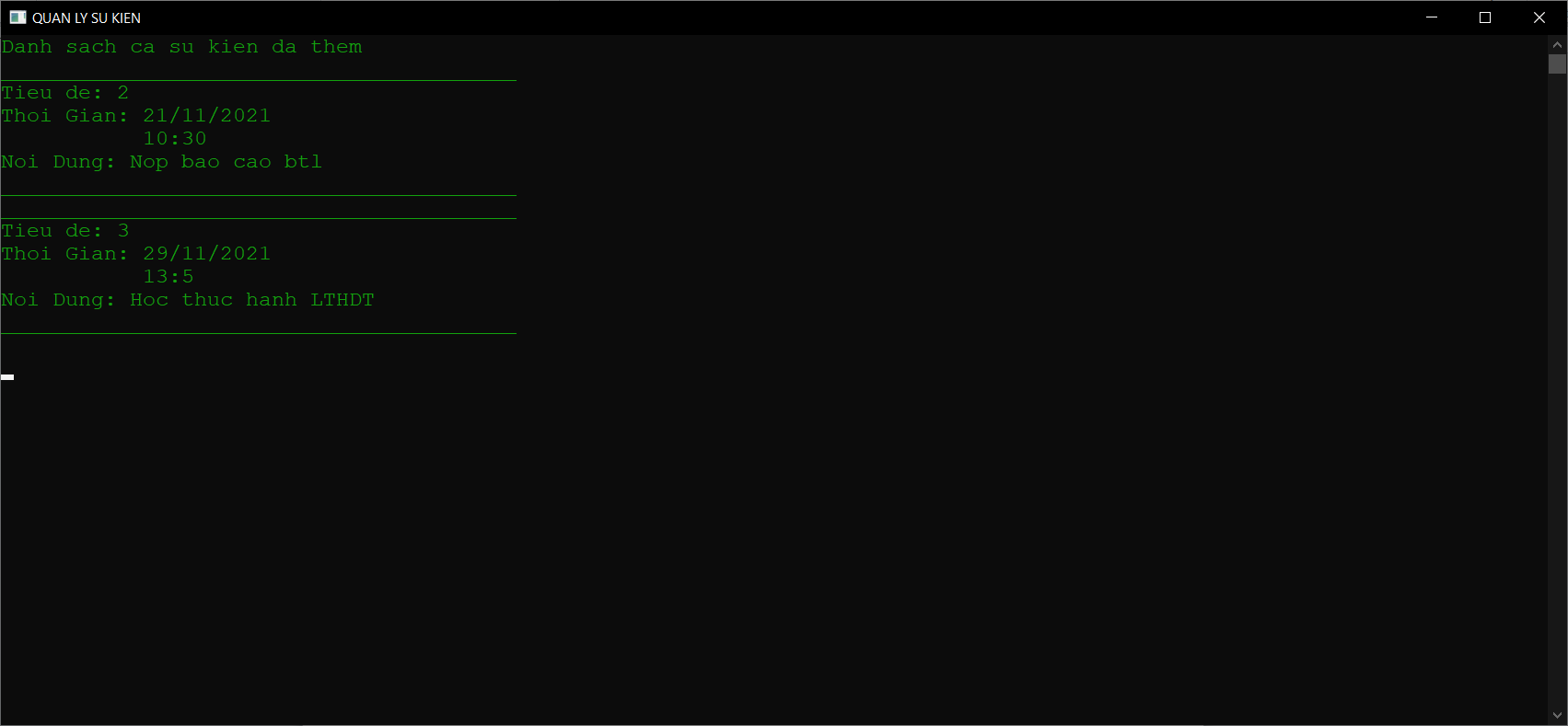


3.3.6 Sau khi nhập , xem , sắp xếp , xóa ,xong sử dụng chức năng SAVE FILE để lưu sự kiện và ghi ra 1 file Notepad ,lưu ý file Notepad sẽ được lưu :





3.3.7 Trước khi thoát khỏi chương trình để chắc chắn file Notepad đã đầy đủ các sự kiện muốn lưu , có thể sử dụng chức năng OPEN FILE để xem lại nhưng sự kiện đã lưu vào file Notepad:



**Hết.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*