# ĐỒ ÁN CUỐI KỲ

# *(Đếm từ trong câu)*

* **THÔNG TIN NHÓM**

## *Hoàng Dân An, 1612001, 16CTT1.1*

## *Nguyễn Phước An, 1612009, 16CTT.2*

## *Nội dung báo cáo: Hoàng Dân An, Nguyễn Phước An*

## *Thiết kế báo cáo: Hoàng Dân An*

# YÊU CẦU CỦA ĐỒ ÁN

# THIẾT KẾ VÀ KIẾN TRÚC CHƯƠNG TRÌNH

## *THIẾT KẾ DỮ LIỆU*

### Từ điển

#### Chia theo chữ cái:

##### a.txt

##### b.txt

##### …

##### y.txt

### NguyenAmDau.txt (File chứa các nguyên âm việt nam)

### PhuAmDau.txt (chứa các phụ âm)

### ThuTuAm.txt (chứa các âm tiết sắp xếp theo alphabet VietNam)

## *PHÂN CHIA HEADER VÀ SOURCE*

### Chương trình sử dụng IDE là *Visual Studio 2013*

### Chương trình được chia thành nhiều project, mỗi project chứa các hàm và hằng liên quan đến project đó:

#### Project TuDon (dùng để kiểm tra từ đơn)

#### Project TuGhep

#### Project TuLay

#### Project DocTuDien (code các hàm đọc dữ liệu)

#### Project ChuongTrinhChinh (chương trình chính và dữ liệu sẽ chạy ở đây, và gọi các hàm từ các project còn lại)

## *CÁC THƯ VIỆN SỬ DỤNG*

### stdio.h (nhập xuất)

#### printf(): ghi thông báo ra màn hình

#### fwprintf(), fwscanf(), fgetws: đọc file và ghi ra file

#### \_wfopen(): mở file widestring.

### conio.h (dừng màn hình)

#### getch();

### stdlib.h

#### exit();

### wchar.h (các thao tác trên widestring)

#### wcslen(): tính độ dài chuỗi

#### wcscpy(): dán chuỗi

#### wcscat(): ghép chuỗi

#### wcstok(): cắt chuỗi

#### wcscmp(): so sánh 2 chuỗi với nhau

### malloc.h (để thao tác với bộ nhớ và mảng động)

#### realloc(): tạo lại vùng nhớ

#### malloc(): khởi tạo vùng nhớ

#### free(): giải phóng vùng nhớ

## *PHÂN CHIA CÔNG VIỆC*

### Nguyễn Phước An:

#### Từ Ghép

#### Từ Đơn

#### Đọc từ điển

### Hoàng Dân An:

#### Từ Láy

#### Chương trình chính

#### Đọc ghi dữ liệu (input, output)

* **MÔ TẢ THUẬT TOÁN VÀ DỮ LIỆU**

## *SƠ ĐỒ CẤU TRÚC CHƯƠNG TRÌNH*

## *CÁC STRUCT VÀ DỮ LIỆU*

## CÁC KIỂU DỮ LIỆU CÓ SẴN

### *wchar\_t*: kiểu kí tự mã hóa unicode

### *int*: kiểu số nguyên

### *bool*: kiểu boolean

### DANH SÁCH CÁC TỪ

struct danhSachTu {

int soTu;

wchar\_t \*\* tu;

};

### KIỂU DỮ LIỆU NGUYÊN ÂM

struct \_nguyenAm {

wchar\_t name; // VD: a

int n; // so chu cai cung nguyen am

wchar\_t \* ds; //danh sach cac chu cai cung nguyen am

// VD: a á à ả ã ạ

};

### DANH SÁCH NGUYÊN ÂM

struct \_danhSachNguyenAm {

int soAm;

\_nguyenAm \* ds; // danh sach cac nguyen am

};

### DANH SÁCH PHỤ ÂM

struct \_danhSachPhuAm {

int soAm;

wchar\_t \*\*phAm;

};

## *KIỂM TRA TỪ ĐƠN VÀ TỪ GHÉP*

* + - **HÀM KIỂM TRA TỪ ĐƠN**

bool KiemTraTuDon(wchar\_t \*s, danhSachTu td[24], \_danhSachNguyenAm &DS\_vowel)

// Kiểm tra từ có nghĩa

* + - * ***Hàm đi kèm:***

wchar\_t mytoLower(wchar\_t a, \_danhSachNguyenAm \* DS)

// chuyển chữ cái thành chữ thường.

int vowelOrder(wchar\_t vowel, wchar\_t &outVowel)

// Lấy số thứ tự của chữ cái trong từ điển.

void DocTuDien(wchar\_t ch, danhSachTu dic[24], \_danhSachNguyenAm \* DS\_vowel)

// Đọc từ điển

* + - * ***Mô tả hàm:*** Tìm vị trí của từ thông qua chữ cái đầu tiên trong bảng 24 chữ, sau đó đọc những chữ bắt đầu bằng chữ cái đó vào mảng. Dùng tìm kiếm tuyến tính để kiểm tra xem có tồn tại từ đó không.
      * ***Biến truyền vào:***
        + wchar\_t \*s : từ đơn cần xét xem có nghĩa hay không.
        + danhSachTu td[24]: 24 danh sách từ điển theo chữ cái.
        + \_danhSachNguyenAm &DS\_vowel: danh sách nguyên âm tiếng Việt.
      * ***Kết quả trả ra:***
        + **TRUE**: nếu từ đơn có nghĩa (nằm trong từ điển)
        + **FALSE**: ngược lại.
    - **HÀM KIỂM TRA TỪ GHÉP**

bool KiemTraTuGhep(wchar\_t \*s1, wchar\_t \*s2, danhSachTu td[24], \_danhSachNguyenAm &dsNgAm);

* + - * ***Hàm đi kèm:***

bool KiemTraTuDon(wchar\_t \*s, danhSachTu td[24], \_danhSachNguyenAm &DS\_vowel)

// Kiểm tra từ có nghĩa

* + - * ***Mô tả hàm:*** dùng 1 biến để ghép 2 từ đơn liên tiếp nhau, sau đó kiểm tra bằng hàm “KiemTraTuDon” để xem có tồn tại.
      * ***Biến truyền vào:***
        + wchar\_t \*s1: từ đơn đứng đầu của từ ghép.
        + wchar\_t \*s2: từ đơn đứng sau của từ ghép.
        + danhSachTu td[24]: mảng 24 danh sách từ điển.
        + \_danhSachNguyenAm &dsNgAm: danh sách nguyên âm tiếng Việt.
      * ***Kết quả trả ra:***
        + **TRUE**: nếu từ ghép có nghĩa (nằm trong từ điển)
        + **FALSE**: ngược lại.

## *KIỂM TRA TỪ LÁY*

### HÀM KIỂM TRA TỪ LÁY

bool KiemTraTuLay(wchar\_t \* s1, wchar\_t \* s2, danhSachTu \* dsTu, \_danhSachNguyenAm \* dsNgAm);

### HÀM ĐI KÈM

wchar\_t \* PhuAmDau(wchar\_t \* s, \_danhSachNguyenAm \* dsNgAm);

wchar\_t \* PhuAmCuoi(wchar\_t \* s, \_danhSachNguyenAm \* dsNgAm);

bool KiemTraPhuAmCuoi(wchar\_t \* s1, wchar\_t \* s2, \_danhSachNguyenAm \* dsNgAm);

// kiem tra xem 2 phu am cuoi co giong nhau khong

bool KiemTraPhuAmDau(wchar\_t \* s1, wchar\_t \* s2, \_danhSachNguyenAm \* dsNgAm);

//kiem tra xem 2 phu am dau co giong nhau khong

#### Mô tả thuật toán:

##### *Hàm PhuAmDau và PhuAmCuoi sẽ tách các kí tự phụ âm đầu hoặc phụ âm cuối ra*

##### *Hàm KiemTraTuLay sẽ kiểm tra xem phụ âm đầu của s1 & s2 có giống nhau không hoặc phụ âm cuối của nó.*

##### *Hàm này chỉ xét tính hợp qui tắc về ngữ pháp của từ láy, chưa xét đến nghĩa của nó.*

### BIẾN TRUYỀN VÀO

#### wchar\_t \* s1: từ đơn thứ nhất của từ ghép cần xét

#### wchar\_t \* s2: từ đơn thứ hai của từ ghép cần xét

#### danhSachTu \* dsTu: danh sách từ điển ứng với chữ cái đầu của từ ghép đó

#### \_danhSachNguyenAm \* dsNgAm: danh sách nguyên âm để kiểm tra và tách các âm trong từ ghép ra.

### KẾT QUẢ TRẢ RA

#### TRUE: Nếu từ ghép đó hợp qui tắc từ láy

#### FALSE: Ngược lại

## *ĐỌC TỪ ĐIỂN VÀ TRUY XUẤT TỪ ĐIỂN*

* + - **HÀM ĐỌC TỪ ĐIỂN**

//Đọc từ điển theo chữ cái

void DocTuDien(wchar\_t ch, danhSachTu dic[24], \_danhSachNguyenAm \* DS\_vowel);

* + - * ***Hàm đi kèm:***

wchar\_t mytoLower(wchar\_t a, \_danhSachNguyenAm \* DS)

// chuyển chữ cái thành chữ thường.

int vowelOrder(wchar\_t vowel, wchar\_t &outVowel)

// Lấy số thứ tự của chữ cái trong từ điển.

* + - * ***Mô tả hàm:*** Có 24 mảng chứa từ, ban đầu kiểm tra xem từ cần đọc đã có chưa (Dùng 2 hàm “mytoLower” và “vowelOrder” để tìm vị trí cần đọc), nếu có thì xuất. Nếu chưa thì đọc vào mảng.
      * ***Biến truyền vào:***
        + **wchar\_t** ch: kí tự của chữ cái mà ta cần đọc từ điển của nó.
        + **danhSachTu** dic[24]: dữ liệu 24 danh sách từ lưu lại từ điển được đọc vào, nếu 1 danh sách từ tương ứng với ***ch*** đã được đọc thì hàm trả ra ngoài.
        + **\_danhSachNguyenAm** \* DS\_vowel: danh sách các nguyên âm tiếng Việt để truy xuất đến file từ điển bên ngoài.

## *ĐẾM TỪ VÀ LOẠI BỎ TRÙNG LẶP (CHƯƠNG TRÌNH CHÍNH)*

### *Khai báo thư viện*

#include "../DocTuDien/DocTuDien.h"

#include "../TuLay/TuLay.h"

#include "../TuLay/NguyenAmPhuAm.h"

#include "../TuDon/TuDon.h"

#include "../TuGhep/TuGhep.h"

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#define fileInput L"../../data/Input/input.txt"

#define fileOutput L"../../data/Output/output.txt"

#### Sử dụng các hàm đã xây dựng ở các project kia.

### *Dữ liệu đọc vào*

danhSachTu my\_input; // du lieu vao

danhSachTu tuDienChu[24]; // tu dien tung chu cai alphabet

\_danhSachNguyenAm dsNgAm; // danh sach cac nguyen am dau

### *Hàm được xây dựng*

bool readInput(danhSachTu &input);

danhSachTu DemTu(danhSachTu &input, danhSachTu tuDienChu[], \_danhSachNguyenAm &dsNgAm);

//dem tu

danhSachTu DemTuDon(danhSachTu &input, danhSachTu tuDienChu[], \_danhSachNguyenAm &dsNgAm);

//dem tu don

danhSachTu DemTuGhep(danhSachTu &input, danhSachTu tuDienChu[], \_danhSachNguyenAm &dsNgAm);

//dem tu ghep

danhSachTu DemTuLay(danhSachTu &dsTuGhep, danhSachTu tuDienChu[], \_danhSachNguyenAm &dsNgAm);

//dem tu lay

bool writeOutput(danhSachTu &input, danhSachTu tuDienChu[], \_danhSachNguyenAm &dsNgAm);

void BaoLoi(char \* mes);

### *Thuật toán chọn từ không bị trùng*

//kiem tra cac chu co trung lap lai khong

bool KiemTraTuTrongDS(wchar\_t \* s, danhSachTu &DS) {

for (int i = 0; i < DS.soTu; ++i) {

if (wcscmp(s, DS.tu[i]) == 0) return true;

}

return false;

}

//them tu vao danh sach

void ThemTuVaoDanhSach(wchar\_t \* s, danhSachTu &DS) {

DS.tu = (wchar\_t\*\*)realloc(DS.tu, (DS.soTu + 1)\*sizeof(wchar\_t\*));

DS.tu[DS.soTu] = new wchar\_t[wcslen(s) + 1];

wcscpy(DS.tu[DS.soTu], s);

DS.soTu++;

}

danhSachTu DemTu(danhSachTu &input, danhSachTu tuDienChu[], \_danhSachNguyenAm &dsNgAm) {

danhSachTu dsTu = { 0, NULL };

for (int i = 0; i < input.soTu; ++i) {

if (KiemTraTuTrongDS(input.tu[i], dsTu) == false) {

ThemTuVaoDanhSach(input.tu[i], dsTu);

}

}

return dsTu;

}

# SỬ DỤNG

## Nhập câu tiếng Việt vào file input.txt trong thư mục *data/Input/*

## Chạy solution DemTu bằng Visual Studio (bản 2013 trở lên)

## Set as StartUp Project cho ChuongTrinhChinh

## Nhấm F5 hoặc click Run để chạy chương trình

## Nếu thành công, chương trình sẽ ghi ra *“Xuat ket qua thanh cong”.*

## Neu khong thanh cong, kiem tra lai cac file trong data.

## Kết quả sẽ nằm trong thư mục *data/Output/output.txt*