| **TRƯỜNG ĐẠI HỌC MỞ HÀ NỘI**  **KHOA CÔNG NGHỆ THÔNG TIN**  ------------------------    **BÁO CÁO BÀI TẬP LỚN**  **MÔN: LẬP TRÌNH HỆ THỐNG**  **ĐỀ TÀI: XÓA THƯ MỤC RỖNG, TÌM SỐ NGUYÊN TỐ VÀ LẤY THÔNG TIN VỀ CỔNG LPT**  **Giảng viên hướng dẫn: TS. Nguyễn Đức Tuấn**  **Nhóm sinh viên thực hiện:** 1. Kiều Hoàng Dương - 1910A03  2. Nguyễn Đình Phước - 1910A02  3. Trần Văn Tiến - 1910A05  **Hà Nội - 2023** |
| --- |

# MỤC LỤC

[**MỤC LỤC 2**](#_heading=h.1fob9te)

[**I. MỞ ĐẦU 3**](#_heading=h.3dy6vkm)

[**II. ĐỀ BÀI 4**](#_heading=h.1t3h5sf)

[**III. MÔ TẢ BÀI TẬP LỚN 5**](#_heading=h.bh4m6n4wba2n)

[1. Chương trình C++ 5](#_heading=h.4d34og8)

[2. Chức năng Xóa thư mục rỗng 7](#_heading=h.2s8eyo1)

[3. Chức năng Tìm số nguyên tố từ 2 đến n 8](#_heading=h.u8u550vhtaeg)

[4. Chức năng Lấy thông tin và hiện địa chỉ của cổng LPT 9](#_heading=h.iexev4f4n3be)

[**IV. LẬP TRÌNH - CÀI ĐẶT 9**](#_heading=h.17dp8vu)

[1. Chương trình chính 9](#_heading=h.3rdcrjn)

[2. Xóa thư mục rỗng 11](#_heading=h.26in1rg)

[3. Tìm các số nguyên tố từ 2 đến n 13](#_heading=h.kflogpgmdjr4)

[4. Số lượng cổng LPT và địa chỉ của cổng 14](#_heading=h.1ksv4uv)

[**V. DEMO 16**](#_heading=h.4i7ojhp)

[1. Menu chương trình 16](#_heading=h.u6k77u48ot3y)

[2. Xóa thư mục rỗng 17](#_heading=h.2p6pwh6nms8o)

[3. Tìm các số nguyên tố từ 2 đến n 18](#_heading=h.91emy56846vm)

[4. Số lượng cổng LPT và địa chỉ của cổng 18](#_heading=h.ulyzlbwjf3z1)

[**VI. KẾT LUẬN 18**](#_heading=h.r2hdg5jz8v9)

[**VII. TÀI LIỆU THAM KHẢO 18**](#_heading=h.2xcytpi)

# 

# MỞ ĐẦU

* Lập trình hệ thống giúp cho sinh viên viết được chương trình bằng ngôn ngữ Assembly trên máy tính PC. Sinh viên có các kỹ năng đơn giản như: Sử dụng trình biên dịch hợp ngữ trong môi trường Window, sửa lỗi, liên kết, khảo sát tập lệnh, các ngắt đơn giản của hệ điều hành DOS. Để vận dụng và nâng cao được kỹ năng lập trình hệ thống bằng hợp ngữ, sinh viên phải nỗ lực rất nhiều trong việc tự học, bổ sung những kiến thức nhất định về phần cứng máy tính cũng như nguyên lý vận hành của các thiết bị ngoại vi có liên quan như: Máy in, hệ vi điều khiển, cổng vào ra nối tiếp/song song, …
* Ngày nay, ứng dụng công nghệ thông tin và việc tin học hóa được xem là một trong những yếu tố mang tính quyết định trong hoạt động của các chính phủ, tổ chức, cũng như các công ty. Nó đóng vai trò hết sức quan trọng, có thể tạo ra những bước đột phá mạnh mẽ. Việc lập trình lên các hệ thống để phục vụ cho các nhu cầu riêng của tổ chức, công ty thậm chí các cá nhân, không lấy gì làm xa lạ. Với một vài thao tác đơn giản, một người bất kì có thể sử dụng một hệ thống đơn giản.

# ĐỀ BÀI

Hãy viết chương trình thực hiện 4 chức năng :

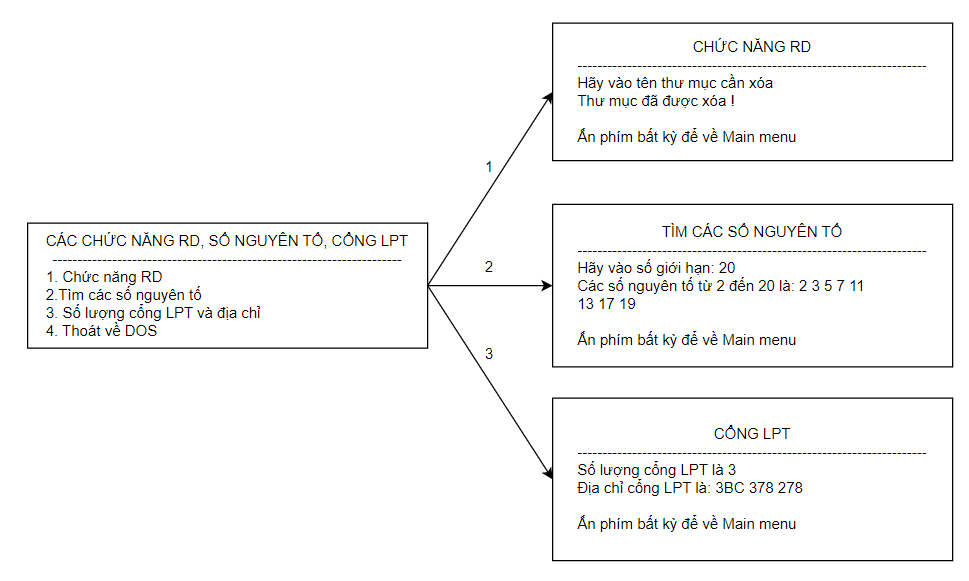
1 .... Chức năng RD

2 .... Tìm các số nguyên tố

3 .... Số lượng cổng LPT và địa chỉ

4 ....Thoát về DOS

Ví dụ khi CT chạy :

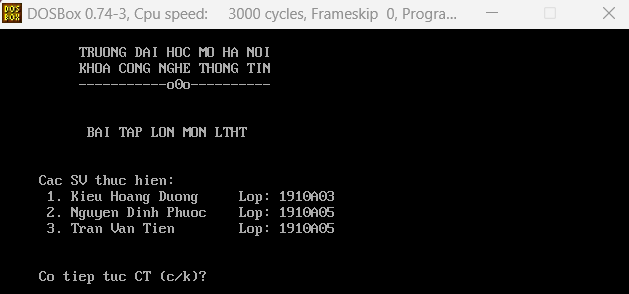


# MÔ TẢ BÀI TẬP LỚN

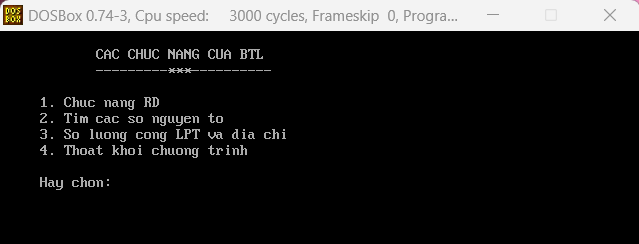
* Đây là chương trình đa tệp, có sự liên kết giữa tệp C++ và tệp ASM. Trong đó, màn hình giới thiệu và màn hình chức năng do chương trình C++ viết, các chức năng cụ thể do chương trình con viết bằng ASM đảm nhiệm. Nhãn dùng chung là tên các chương trình con từ tệp ASM.

## Chương trình C++

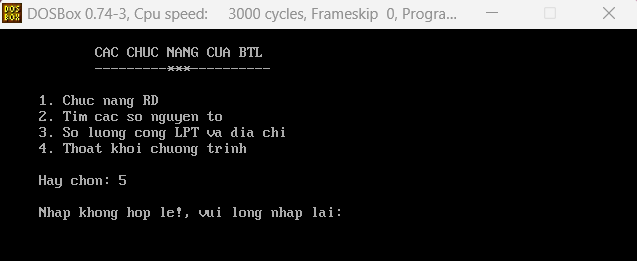
* Cần khai báo extern các nhãn dùng chung là tên các chương trình con bao gồm: XOATM, SONT và CONGLPT.
* Chương trình C++ sẽ hiện màn hình giới thiệu thông tin của sinh viên và lời nhắc ‘Có tiếp tục CT (c/k): ’. Nếu nhấn phím ‘c’ thì màn hình chức năng sẽ hiện ra, ngược lại thì thoát khỏi chương trình. Màn hình giới thiệu sẽ có nội dung như sau:



* Nếu phím ‘c’ được nhấn, màn hình chức năng của chương trình sẽ có dạng:



* Tại đây, sẽ thấy danh sách các chức năng chính của chương trình bao gồm:
* 1. Chức năng RD.
* 2. Tìm các số nguyên tố.
* 3. Số lượng cổng LPT và địa chỉ.
* 4. Thoát khỏi chương trình.
* Muốn sử dụng chức năng nào thì chỉ cần nhấn phím số tương ứng của chức năng đó. Nếu bạn lỡ nhấn sai các giá trị của các phím từ 1 đến 4 sẽ có thông báo ‘Nhập không hợp lệ! Vui lòng nhập lại: ’



* Nhấn phím 4 để thoát khỏi chương trình. Ngược lại nếu chọn các phím chức năng từ 1 đến 3, chương trình C++ sẽ gọi chương trình con tương ứng do tệp ASM viết. Khi sử dụng xong sẽ thoát chương trình con và trở về MENU lựa chọn chính.

## Chức năng Xóa thư mục rỗng

* Chức năng: Xóa thư mục rỗng
* Cách làm:
* Hiển thị thông báo nhập tên thư mục cần tạo (tm).
* Chương trình sử dụng lệnh HienString để hiển thị thông báo yêu cầu nhập tên thư mục.
* Nhập tên thư mục từ người dùng.
* Sử dụng hàm GET\_FILE\_NAME để lấy tên thư mục được nhập từ bàn phím và lưu vào biến buff.
* Gọi hàm DOS để tạo thư mục (int 21h, ah=3Ah).
* Đặt tên thư mục được nhập từ buff vào dir\_name.
* Gọi hàm DOS với int 21h, ah=3Ah để tạo thư mục.
* Kiểm tra kết quả của lệnh DOS.
* Nếu jnc L\_TM1, tức là không có lỗi (CF=0), chương trình hiển thị thông báo tạo thư mục thành công (Suc\_MD).
* Ngược lại, nếu có lỗi (CF=1), chương trình hiển thị thông báo lỗi (Err\_MD).
* Hiển thị thông báo tiếp tục hoặc thoát.
* Hiển thị thông báo yêu cầu tiếp tục hoặc thoát khỏi chương trình (tieptuc).
* Chờ người dùng nhập một ký tự (int 21h, ah=1).
* Nếu ký tự là 'c', chương trình quay lại bước 1 (jmp L\_TM0).
* Nếu ký tự không phải 'c', chương trình kết thúc (jmp Thoat\_TM).
* Kết thúc chương trình.
* Nếu Thoat\_TM được chạy, chương trình sẽ kết thúc và quay về DOS.
* Các MACRO và chương trình con sử dụng:
* CLRSCR: Xóa màn hình.
* HienString: MACRO hiện một xâu ký tự kết thúc bằng '$' ra màn hình.
* GET\_FILE\_NAME: Chương trình con nhận tên tệp vào từ bàn phím.

## Chức năng Tìm số nguyên tố từ 2 đến n

* Chức năng: Tìm các số nguyên tố từ 2 đến n (n được nhập từ bàn phím)
* Cách làm:
* Gọi hàm VAO\_SO\_N để nhập giới hạn từ người dùng và lưu vào thanh ghi AX.
* Gọi hàm HIEN\_SO\_N để hiển thị giới hạn vừa nhập.
* Sử dụng vòng lặp từ 2 đến giới hạn vừa nhập.
* Kiểm tra từng số trong vòng lặp xem có phải số nguyên tố không.
* Nếu là số nguyên tố, hiển thị nó lên màn hình.
* Thuật toán kiểm tra số nguyên tố: kiểm tra xem số đó có thể chia hết cho bất kỳ số nguyên nào từ 2 đến căn bậc hai của nó không. Nếu không, nó là số nguyên tố.
* Các MACRO và chương trình con sử dụng:
* CLRSCR: Xóa màn hình.
* HienString: MACRO hiện một xâu ký tự kết thúc bằng '$' ra màn hình.
* VAO\_SO\_N: Nhận một số nguyên từ bàn phím, kết thúc số bằng ENTER (0dh). Số nhận được nằm trong AX (không được đánh sai và sửa).
* HIEN\_SO\_N: Hiện nội dung thanh ghi AX ra màn hình dạng cơ số 10.

## Chức năng Lấy thông tin và hiện địa chỉ của cổng LPT

* Chức năng: Lấy thông tin và hiện địa chỉ của cổng LPT
* Cách làm:
* Sử dụng ngắt int 11h để lấy số lượng cổng LPT (kết quả được chứa trong thanh ghi AH).
* Chia số lượng cổng cho 26 (dịch phải 6 bit) để kiểm tra xem có cổng LPT hay không. Nếu khác 0, tức là có ít nhất một cổng LPT, thì nhảy tới bước tiếp theo.
* Hiển thị thông báo "Co" để thông báo rằng có cổng LPT.
* Thiết lập CL bằng số lượng cổng LPT.
* Hiển thị thông điệp "So luong cong LPT ma may tinh co la:" kèm theo số lượng cổng LPT.
* Hiển thị địa chỉ của từng cổng LPT bằng cách sử dụng int 10h để in mã ASCII tương ứng với địa chỉ HEXA của cổng LPT.
* Các MACRO và chương trình con sử dụng:
* CLRSCR: Xóa màn hình.
* HienString: MACRO hiện một xâu ký tự kết thúc bằng '$' ra màn hình.
* HIEN\_HEXA: Chương trình con hiện 1 số nguyên trong thanh ghi AX ra màn hình dạng cơ số HEXA.

# LẬP TRÌNH - CÀI ĐẶT

## Chương trình chính

#include <iostream.h>

#include <conio.h>

extern XOATM();

extern SONT();

extern CONGLPT();

void main()

{

char tl;

int cn;

clrscr();

cout << "\n Truong Dai hoc Mo Ha Noi";

cout << "\n KHOA CONG NGHE THONG TIN";

cout << "\n -----------o0o----------";

cout << "\n\n\n BAI TAP LON MON LTHT";

cout << "\n\n\n Cac SV thuc hien:";

cout << "\n 1. Kieu Hoang Duong Lop: 1910A03";

cout << "\n 2. Nguyen Dinh Phuoc Lop: 1910A05";

cout << "\n 3. Tran Van Tien Lop: 1910A05";

cout << "\n\n\n Co tiep tuc CT (c/k)?";

cin >> tl;

if (tl == 'c')

{

while (1)

{

clrscr();

cout << "\n CAC CHUC NANG CUA BTL";

cout << "\n ---------\*\*\*----------";

cout << "\n\n 1. Chuc nang RD";

cout << "\n 2. Tim cac so nguyen to";

cout << "\n 3. So luong cong LPT va dia chi";

cout << "\n 4. Thoat khoi chuong trinh";

cout << "\n\n Hay chon: ";

cin >> cn;

while (cn <= 0 || cn >= 5)

{

cout << "\n Nhap khong hop le!, vui long nhap lai: ";

cin >> cn;

}

switch (cn)

{

case 1:

XOATM();

break;

case 2:

SONT();

break;

case 3:

CONGLPT();

break;

case 4:

cout << " Cam on da su dung he thong!\n";

return;

default:

cout << "\n Vui long nhap lai!";

}

}

}

}

## Xóa thư mục rỗng

INCLUDE lib1.asm

.MODEL small

.STACK 100h

.DATA

tm db 13, 10, ' CHUC NANG RD'

db 13, 10, ' ------------'

db 13, 10,'Hay vao ten thu muc can xoa: $'

Err\_MD db 13, 10,'Khong xoa duoc thu muc!$'

Suc\_MD db 13, 10,'Thu muc da duoc xoa!$'

prompt db 13, 10, 'An phim bat ky de ve Main Menu$'

buff db 30

db ?

dir\_name db 30 dup(?)

.CODE

PUBLIC @XOATM$qv

@XOATM$qv PROC

PS:

mov ax, @data

mov ds, ax

L0:

CLRSCR

HienString tm

lea dx, buff

call GET\_FILE\_NAME

lea dx, dir\_name

mov ah, 3Ah

int 21h

jnc L1

HienString Err\_MD

jmp EXIT

L1:

HienString Suc\_MD

EXIT:

HienString prompt

mov ah, 1

int 21h

ret

@XOATM$qv ENDP

INCLUDE lib3.asm

END PS

## Tìm các số nguyên tố từ 2 đến n

INCLUDE lib1.asm

.MODEL small

.STACK 100h

.DATA

snt1 db 13, 10, ' TIM CAC SO NGUYEN TO'

db 13, 10, ' --------------------'

db 13, 10, 'Hay vao so gioi han: $'

snt2 db 'Cac so nguyen to tu 2 den $'

snt3 db ' la: $'

space db ' $'

prompt db 13, 10, 'An phim bat ky de ve Main Menu$'

so dw ?

.CODE

PUBLIC @SONT$qv

@SONT$qv PROC

PS:

mov ax,@data

mov ds,ax

L0:

CLRSCR

HienString snt1

call VAO\_SO\_N

HienString snt2

call HIEN\_SO\_N

HienString snt3

mov bx,ax

mov so,1

L1:

inc so

mov ax,so

cmp ax,bx

jg EXIT

mov cx,ax

shr cx,1

L2:

cmp cx,1

jle L3

xor dx,dx

div cx

and dx,dx

jz L1

mov ax,so

loop L2

L3:

call HIEN\_SO\_N

HienString space

jmp L1

EXIT:

HienString prompt

mov ah,1

int 21h

ret

@SONT$qv ENDP

INCLUDE lib2.asm

END PS

## Số lượng cổng LPT và địa chỉ của cổng

INCLUDE lib1.asm

.MODEL small

.STACK 100h

.DATA

M1 db 13, 10, ' CONG LPT'

db 13, 10, ' --------'

db 13, 10, 'May tinh dang dung co cong LPT khong? $'

co db 'Co $'

khong db 'Khong'

M2 db 13, 10,'So luong cong LPT ma may tinh co la: $'

M3 db 13, 10,'Dia chi cac cong LPT la: $'

space db ' $'

prompt db 13, 10, 'An phim bat ky de ve Main Menu$'

.CODE

PUBLIC @CONGLPT$qv

@CONGLPT$qv PROC

PS:

mov ax,@data

mov ds,ax

CLRSCR

HienString M1

int 11h

mov al,ah

mov cl,6

shr al,cl

jnz L1

HienString khong

jmp EXIT

L1:

HienString co

mov cl,al

xor ch,ch

HienString M2

add al,30h

mov ah,0eh

int 10h

HienString M3

xor ax,ax

mov es,ax

mov bx,408h

L2:

mov ax,es:[bx]

call HIEN\_HEXA

HienString space

add bx,2

loop L2

EXIT:

HienString prompt

mov ah,1

int 21h

ret

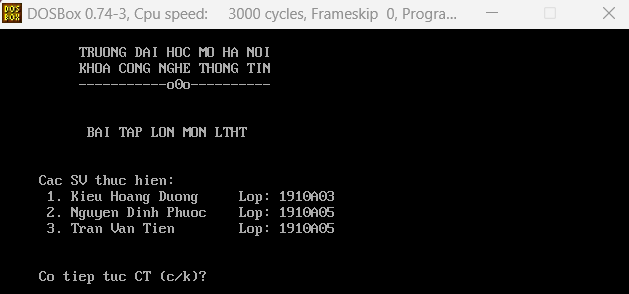
@CONGLPT$qv ENDP

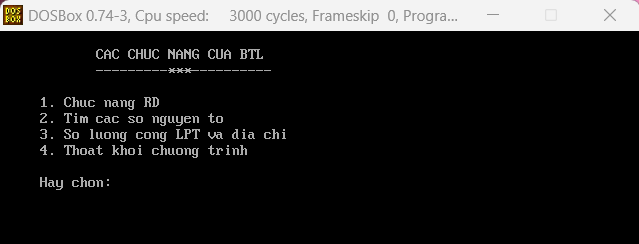
INCLUDE lib2.asm

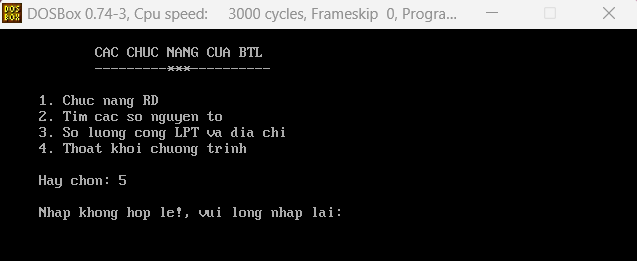
END PS

# DEMO

## Menu chương trình





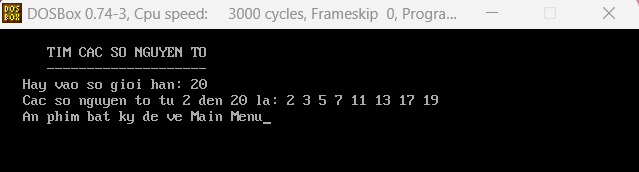


## Xóa thư mục rỗng

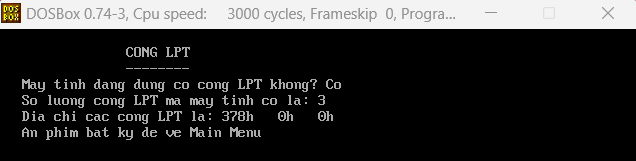




## Tìm các số nguyên tố từ 2 đến n



## Số lượng cổng LPT và địa chỉ của cổng



# KẾT LUẬN

* Kết quả bài tập thực hiện đúng, đủ các yêu cầu được nêu. Mã nguồn ngắn gọn, dễ hiểu, thực hành đúng theo những kiến thức đã được dạy và truyền đạt. Chi tiết, tỉ mỉ, có kiểm tra dữ liệu đầu vào và cảnh báo người dùng. Giao diện đơn giản, hiệu quả, không cầu kỳ, rườm rà gây rối mắt.

# TÀI LIỆU THAM KHẢO

* Đặng Thành Phu, **Các bài tập lập trình bằng ngôn ngữ Assembly,** Nhà xuất bản Khoa học và Kỹ thuật, 2010.
* Turbo Assembler Ver 4.0 (User’s Guide). Borland International, INC. 1880 GREEN HILLS ROAD, 1993.