**Bài 1: Lập trình hợp ngữ Assembly**

*Câu 1: Viết chương trình hợp ngữ in ra lời chào Tiếng Anh và Tiếng Việt*

**Flow chart**

A diagram of a computer

Description automatically generated

**Hình 1.** Flowchart câu 1

**Mã nguồn**

A screenshot of a computer

Description automatically generated

**Hình 2.** Mã nguồn câu 1

A screenshot of a phone

Description automatically generated**Giao diện hiển thị**

**Hình 3.** Giao diện hiển thị câu 1

*Câu 6. Viết chương trình hợp ngữ Assembly cho phép nhập một chuỗi các ký*

*tự kết thúc bởi "# "và yêu cầu in ra màn hình chuỗi ký tự đó theo*

*thứ tự ngược lại.*

**Flowchart**

**A diagram of a flowchart

Description automatically generated**

**Hình 1.** Flowchart câu 6

**Mã nguồn**

A screenshot of a computer

Description automatically generated

**Hình 2.** Mã nguồn câu 6

**Giao diện hiển thị**

A screenshot of a computer screen

Description automatically generated **Hình 3.** Giao diện hiển thị câu 6

*Câu 12. Viết chương trình hợp ngữ Assembly cho phép nhập vào một số và in*

*ra màn hình giai thừa của số đó.*

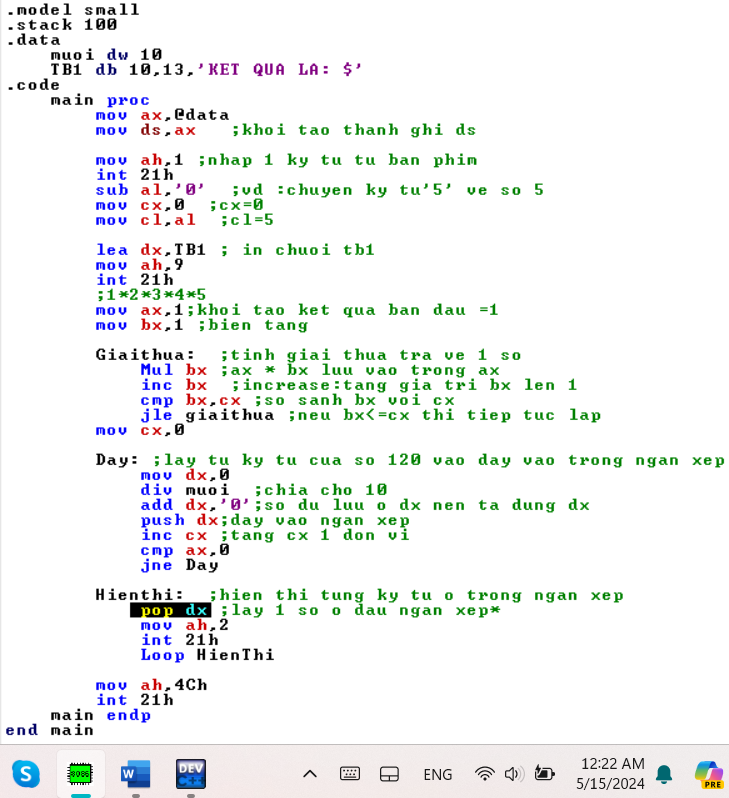
**Flowchart**

A diagram of a computer program

Description automatically generated with medium confidence

**Hình 1.** Flowchart câu 12

**Mã nguồn**



**Hình 2.** Mã nguồn câu 12

**Giao diện chương trình**

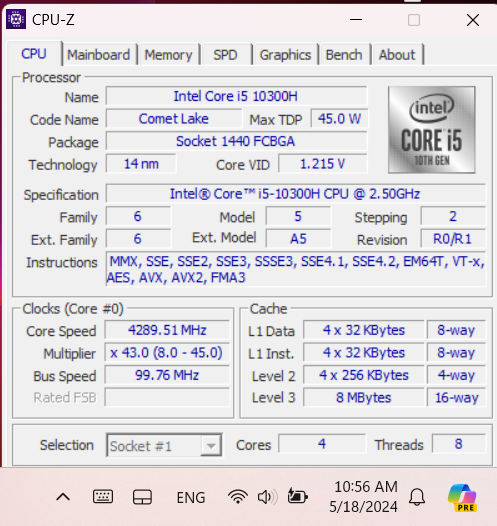
**A screenshot of a computer

Description automatically generated**

**Hình 3.** Giao diện chương trình câu 12

**Bài số 2: Thực hành phân tích khảo sát hệ thống bộ nhớ**

*Phần 1: Khảo sát cấu hình của máy và hệ thống bộ nhớ của máy đang sử dụng (Bộ nhớ trong: ROM, RAM, Cache System, Bộ nhớ ngoài: ổ đĩa cứng, CD, Thiết bị vào ra)*

**

CPU và cache

A screenshot of a computer

Description automatically generated

Ram

A screenshot of a computer

Description automatically generated

Rom

A screenshot of a computer

Description automatically generated

Thiết bị vào ra

*Phần 2: Dùng công cụ Debug khảo sát nội dung các thanh ghi IP, DS, ES,SS, CS, BP, SP*

*A screenshot of a computer

Description automatically generated*

*Phần 3: Giải thích nội dung các thanh ghi, trên cơ sở đó giải thích cơ chế quản lý bộ nhớ của hệ thống trong trường hợp cụ thể này.*

Câu 1.

Khi chương trình bắt đầu chạy, hệ điều hành tự động khởi tạo các thanh

ghi, vùng nhớ và cấp phát không gian địa chỉ cho chương trình.

A screenshot of a computer

Description automatically generated

Tương ứng với các câu lệnh trong mã nguồn, nội dung các thanh ghi có thể thay đổi hoặc không (1 lần single step).

1. **IP (Instruction Pointer)**: Là thanh ghi trỏ đến địa chỉ bộ nhớ của lệnh hiện tại được thực thi. IP trỏ đến địa chỉ của từng dùng lệnh nên luôn luôn thay đổi 1 lượng sau mỗi dòng lệnh được thực hiện. Địa chỉ chính xác của mỗi dòng lệnh trong bộ nhớ là CS\*10+IP

A screenshot of a computer

Description automatically generated

1. **DS (Data Segment)**: Là thanh ghi giữ địa chỉ của phân đoạn dữ liệu hiện tại trong bộ nhớ. Mặc định là DS = 700h nhưng khi chạy xong câu lệnh mov thì nội dung thanh ghi đã đổi thành địa chỉ bắt đầu thực tế của đoạn dữ liệu, tức 717h.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **ES (Extra segment)**: Thanh ghi đoạn dữ liệu mở rộng. ES chứa địa chỉ bắt đầu đoạn dữ liệu mở rộng. Mặc định là 700h. Trong cả chương trình không có sự thay đổi.

A screenshot of a computer

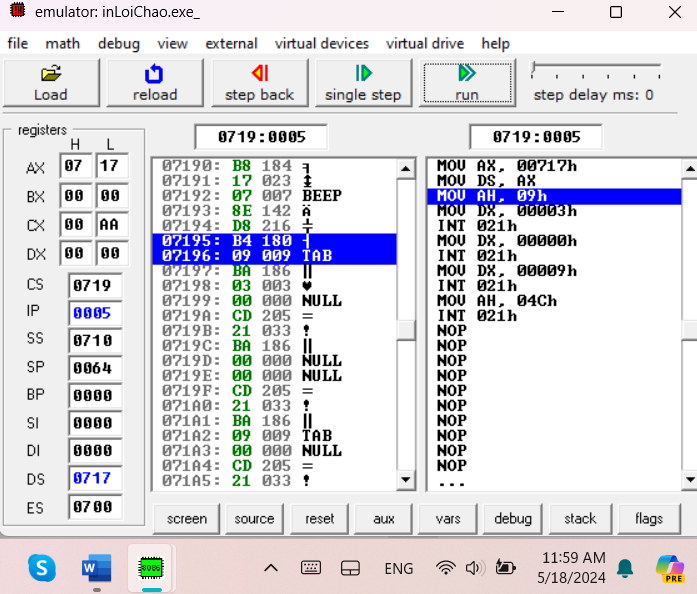
Description automatically generated

1. **SS (Stack Segment)**: Là thanh ghi giữ địa chỉ của phân đoạn ngăn xếp trong bộ nhớ. Vì chỉ dùng 1 stack mặc định ban đầu nên SS không đổi = 710

A screenshot of a computer

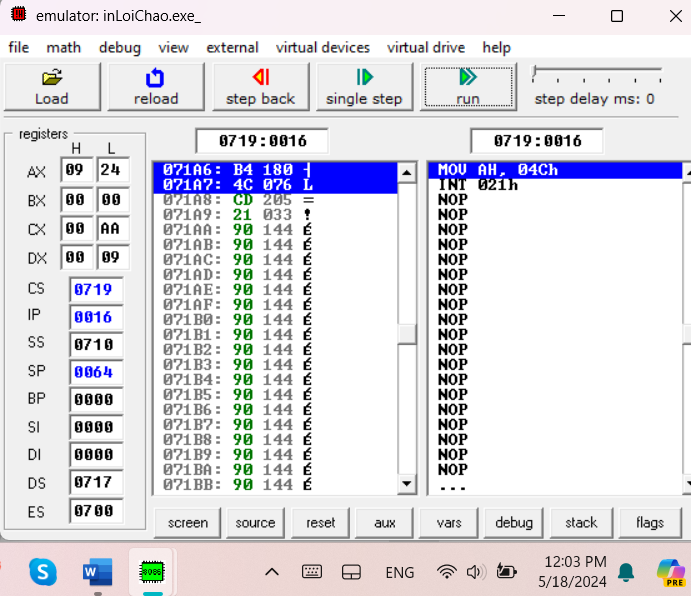
Description automatically generated

1. **CS (Code segment)**: Thanh ghi đoạn mã. CS chứa địa chỉ bắt đầu đoạn mã. Có sự thay đổi khi gọi đến các hàm ngắt của int 21h.

A screenshot of a computer

Description automatically generated

1. **BP (Base Pointer)**: Con trỏ cơ sở, chứa địa chỉ của dữ liệu trong đoạn ngăn xếp SS hoặc các đoạn khác. Trong chương trình này không có sự thay đổi giá trị.



1. **SP (Stack pointer)**: Là thanh ghi giữ địa chỉ của đỉnh của ngăn xếp chương trình. Khi một hàm mới được gọi (bao gồm cả các hàm của ngắt int 21h), địa chỉ trả về của hàm được lưu vào trong stack, giá trị của SP trỏ đến đỉnh của hàm đó.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated