



**HỌC VIỆN KỸ THUẬT MẬT MÃ**

**THỰC HÀNH LẬP TRÌNH  
HỢP NGỮ TRÊN 8086**



# NỘI DUNG 5

## XỬ LÝ CHUỖI KÝ TỰ

### MỤC TIÊU

Viết được các chương trình xử lý chuỗi ký tự bằng các lệnh xử lý chuỗi.

### KIẾN THỨC CẦN CHUẨN BỊ

- Bảng mã ASCII.
- Kết quả của các bài trước
- Các hàm 01h, 02h, 06h, 08h, 09h, 0Ah của INT 21h và các lệnh xử lý chuỗi như MOVSB/W, SCASB/W, STOSB/W, CMPSB/W....



## 6.1. So sánh chuỗi

Soạn thảo như đoạn chương trình trên và lưu với tên là BAI\_6A.ASM. -  
Biên dịch và cho chạy file BAI\_6A.ASM để kiểm tra và xem kết quả

```
writeln macro bien1  
LOCAL bien1  
mov ah,09  
lea dx, bien1  
int 21h  
mov ah,02h  
mov dl, 0ah  
int 21h  
mov dl, 0dh  
int 21h  
endm  
dseg segment
```

```
tbao db "Chương trình so sanh oldpass va newpass$"  
oldpass db "0123456789"  
newpass db "1234567890"  
tbao1 db "Haichuoi giống nhau $"  
tbao2 db "Haichuoi không giống nhau $"  
dseg ends  
cseg segment  
assume cs:cseg, ds:dseg, es: dseg
```



## 6.1. So sánh chuỗi

Soạn thảo như đoạn chương trình trên và lưu với tên là BAI\_6A.ASM. - Biên dịch và cho chạy file BAI\_6A.ASM để kiểm tra và xem kết quả

```
begin: mov ax, dseg
mov ds, ax
mov es, ax
writeln tbao
cld ; chonchieu xu ly chuoi
mov cx, 10 ; so ky tu/so byte can so sanh
lea si, oldpass; (DS:SI)--> dia chi cua chuoi nguon
lea di, newpass; (ES:DI)--> dia chi cua chuoi dich
repe cmpsb ; so sanh tung ky tu/byte
```

```
je intb1
writeln tbao2
jmp thoat
intb1: writeln tbao2
thoat:
mov ah,08h
int 21h
mov ah, 4ch
int 21h
cseg ends
end begin
```



## 6.1. So sánh chuỗi

### Câu hỏi:

- Trong macro writeln, các lệnh nào có chức năng xuống dòng sau khi in xong chuỗi ký tự.
- Mục đích của việc khai báo LOCAL bien1 trong macro là gì?
- Hãy cho biết địa chỉ của DS và ES có giống nhau hay không? Điều này được thể hiện qua các câu lệnh nào trong đoạn chương trình mẫu? Tại sao người ta không khai báo DS và ES trên các phân đoạn khác nhau ?
- Tiền tố REPE trong đoạn chương trình mẫu trên có ý nghĩa như thế nào?
- Ta có thể thay đổi lệnh REPE CMPSB thành một nhóm lệnh khác được không? Nếu được hãy thay đổi nó, biên dịch và chạy chương trình để kiểm chứng.



## 6.1. So sánh chuỗi

### Câu hỏi:

- Thử thay đổi nội dung ở oldpass và newpass sao cho chúng giống nhau. Biên dịch và chạy chương trình xem kết quả, sau đó hãy giải thích cơ chế làm việc của đoạn lệnh từ lệnh CLD cho đến lệnh REPE CMPSB.
- Giả sử, người ta muốn thay thế lệnh cmpsb thành lệnh cmpsw, các bạn có cần sửa đổi các lệnh nào trong chương trình hay không ? Tại sao ?  
Biên dịch và chạy chương trình để kiểm chứng.



## 6.1. So sánh chuỗi

### Câu hỏi:

- Lệnh jmp thoát trong đoạn chương trình trên có nhiệm vụ gì? Thử bỏ lệnh jmp thoát sau đó biên dịch và chạy chương trình xem kết quả.
- Hãy sửa đổi file BAI\_6A.ASM và lưu với tên BAI\_6A1.ASM để có thể thực hiện được nhiệm vụ sau: nhập vào một chuỗi ký tự có tối đa 10 ký tự, trong lúc nhập chỉ hiện thị ra ký tự “\*”. Khi đã nhập đủ 10 ký tự hoặc khi gặp phím ESC thì sẽ in ra các ký tự đã nhập ra màn hình. Gợi ý: dùng hàm 08h, 02h hoặc 09h của int 21h, lệnh loop, cmp, ... Cần phải khai báo dùng đệm để lưu các ký tự đã nhập.



## 6.1. So sánh chuỗi

### Câu hỏi:

Hãy sửa đổi file BAI\_6A1.ASM, kết hợp với file BAI\_6A.ASM (chương trình mẫu) và lưu với tên BAI\_6A2.ASM để có thể thực hiện được nhiệm vụ sau: nhập vào một chuỗi ký tự có 10 ký tự, trong lúc nhập chỉ hiện thị ra ký tự “\*”. Sau đó so sánh với một oldpass có nội dung tùy ý (nhưng chỉ có độ dài là 10 ký tự mà do chúng ta gán trước, ví dụ như oldpass db “1234567890”). Nếu 10 ký tự vừa nhập có nội dung giống oldpass thì in ra câu thông báo “Ban da nhap dung roi” và thoát, ngược lại thì in ra câu thông báo “Ban da nhap sai roi va vui long nhap lai” và quay trở lại nhập cho đến khi nào đúng mới thoát. Gợi ý: dùng hàm 08h, 02h, 09h của int 21h và các lệnh loop, cmpsb





## 6.2. Chuyển nội dung giữa các chuỗi

**Soạn thảo như đoạn chương trình mẫu và lưu với tên là BAI\_6B.ASM.**

- Biên dịch và cho chạy file BAI\_6B.ASM để kiểm tra và xem kết quả.**



## 6.2. Chuyển nội dung giữa các chuỗi

dseg segment

string1 db "Khong co gi quy hon doc lap tu do"

string2 db 34 dup('\$')

dseg ends

cseg segment

assume cs:cseg, ds:dseg, es: dseg

begin: mov ax, dseg

mov ds, ax

mov es, ax

cld ; chon chieu xu ly chuoi

end begin

mov cx, 33 ; so ky tu/so byte can di chuyen

lea si, string1 ; (DS:SI)--> dia chi cua chuoi nguon

lea di, string2 ; (ES:DI)--> dia chi cua chuoi dich

rep movsb ; di chuyen tung byte

mov ah, 09h

lea dx, string2

int 21h

mov ah, 08h ; dung man hinh de xem ket qua

int 21h

mov ah, 4ch ; thoat ve Dos

int 21h

cseg ends



## 6.2. Chuyển nội dung giữa các chuỗi

### Câu hỏi:

- Tại sao chỉ di chuyển 33 ký tự/byte mà lại khai báo biến string2 db 34 dup('\$'). Thử thay 34 thành 33 và tiến hành biên dịch, chạy chương trình để xem kết quả. Có nhận xét gì về vấn đề này không ?. Giải thích ý nghĩa của việc khai báo này.
- Thay lệnh CLD trong đoạn chương trình mẫu thành STD. Biên dịch và cho chạy chương trình để kiểm chứng kết quả. Cho nhận xét về kết quả nhận được.
- Nếu thay lệnh movsb thành MOVSW thì chúng ta có phải thay đổi giá trị nào trong đoạn chương trình mẫu trên không? Nếu có thay đổi, hãy biên dịch và chạy chương trình để kiểm chứng lại kết quả.



## 6.2. Chuyển nội dung giữa các chuỗi

### Câu hỏi:

- Có thể thay thế lệnh REP MOVSB bởi một số lệnh khác hay không? Nếu được thì hãy thay đổi và sau đó biên dịch, cho chạy chương trình để kiểm chứng lại kết quả.
- Giả sử ta có nội dung của một biến string1 có tổng số byte  $> 256$  byte thì lúc đó chúng ta phải khai báo lại các biến này như thế nào ?. Lúc này có khó khăn gì xảy ra không ?. Hãy thử sửa lại, sau đó biên dịch và chạy chương trình để xem kết quả.
- Có cách nào xác định chiều dài của một biến bất kỳ hay không ?. Hãy cho biết các cú pháp của lệnh có thể thực hiện được yêu cầu này. Sau đó thử áp dụng để xác định chiều dài của biến string1



## 6.3. Tìm ký tự trong chuỗi

**Soạn thảo như đoạn chương trình mẫu và lưu với tên là BAI\_6C.ASM**

- Biên dịch và cho chạy file BAI\_6C.ASM để kiểm tra và xem kết quả.

```
write macro bien1
mov ah, 09h
lea dx, bien1
int 21h
endm
dseg segment
string1 db "NGAC NHIEN CHUA ?"
tb1 db "co ky tu A trong chuoai string1 $"
tb2 db "khong ky tu A trong chuoai string1 $"
dseg ends
end begin
assume cs:cseg, ds:dseg, es: dseg
```



## 6.3. Tìm ký tự trong chuỗi

```
begin: mov ax, dseg
mov ds, ax
mov es, ax
cld ; chon chieu xu ly chuoi
mov cx, 17 ; so ky tu can tim
mov al, 'A' ; tim kien ky tu A trong string1
lea di, string1 ; (ES:DI)--> dia chi cua chuoi dich
repne scasb ; lap lai viec tim kiem ky tu cho den
jne intb2 ; khi gap duoc hoac den het chuoi
write tb1
jmp thoat
intb2: write tb2
thoat: mov ah, 08h ; dung man hinh de xem ket qua
int 21h
mov ah, 4ch ; thoat ve Dos
int 21h
cseg ends
end begin
```



## 6.3. Tìm ký tự trong chuỗi

### Câu hỏi:

- Thay lệnh MOV AL, 'A' thành MOV AL, 'B', sau đó biên dịch và chạy chương trình để xem kết quả.
- Giải thích nhiệm vụ của các lệnh từ CLD cho đến JNE INTB2. Có thể thay thế lệnh REPNE SCASB thành các lệnh khác được không ?. Nếu được hãy thay thế chúng, biên dịch và chạy chương trình để kiểm chứng lại kết quả.