Chương 13 :LẬP TRÌNH XỬ LÝ MẢNG & CHUỖI

- GiỚI THIỆU
- CỜ HƯỚNG DF
- CÁC LỆNH THIẾT LẬP VÀ XÓA CỜ HƯỚNG
- CÁC LỆNH THAO TÁC TRÊN CHUỖI
- MỘT SỐ THÍ DỤ MINH HỌA
- THƯ ViỆN LIÊN QUAN ĐẾN CHUỖI

GiỚI THIỆU CHUỖI

Trong ASM 8086 khái niệm chuỗi bộ nhớ hay chuỗi là 1 mảng các byte hay word.

➔ Các lệnh thao tác với chuỗi cũng được thiết kế cho các thao tác với mảng.

Cờ hướng DF

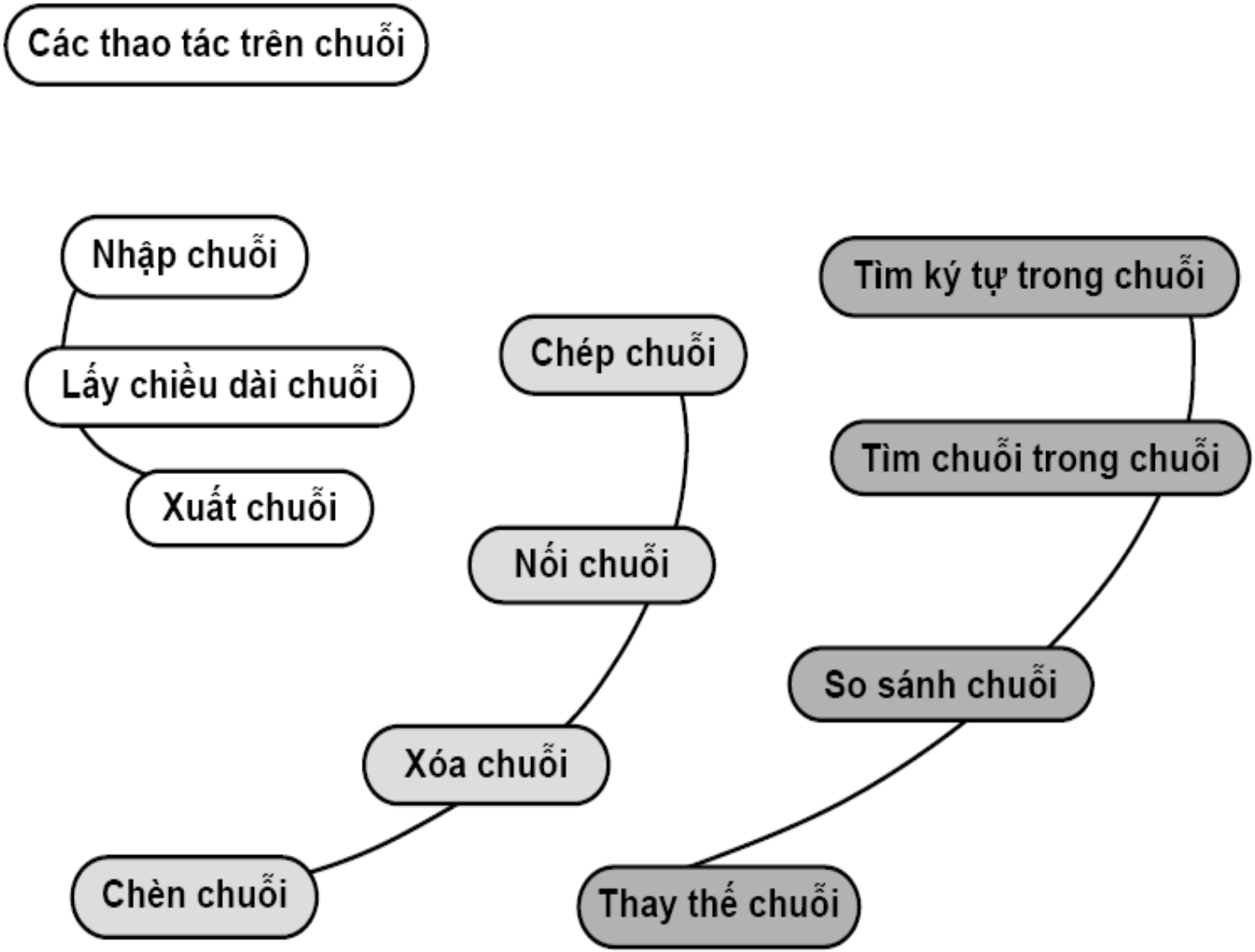
Cờ định hướng (Direction Flag) : xác định hướng cho các thao tác chuỗi.

DF=0 chuỗi được xử lý theo chiều tăng tức địa chỉ vùng nhớ chứa chuỗi tăng dần.
(chuỗi được xử lý từ trái qua phải).

DF=1 chuỗi được xử lý theo chiều tăng tức địa chỉ vùng nhớ chứa chuỗi giảm dần.
(chuỗi được xử lý từ phải qua trái).

Trong DEBUG DF=0 ký hiệu là UP DF=1 ký hiệu là DN


LỆNH LIÊN QUAN ĐẾN CỜ HƯỚNG
CLD (CLEAR DIRECTION FLAG)
XÓA CỜ HƯỚNG DF =0
STD (SET DIRECTION FLAG)
THIẾT LẬP CỜ HƯỚNG DF=1

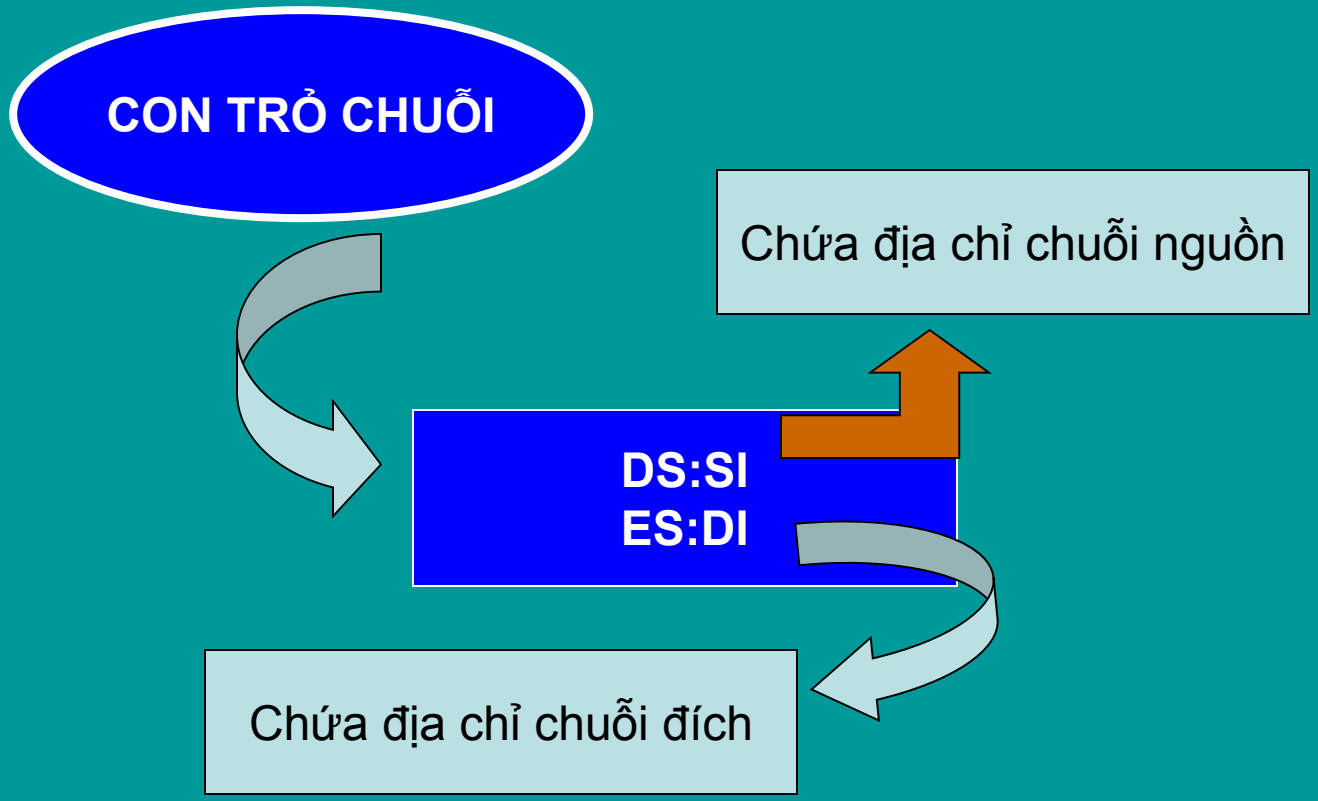

Các thao tác trên chuỗi
Nhập chuỗi
Lấy chiều dài chuỗi
Xuất chuỗi
Chép chuỗi
Nối chuỗi
Xóa chuỗi
Chèn chuỗi
Tìm ký tự trong chuỗi
Tìm chuỗi trong chuỗi
So sánh chuỗi
Thay thế chuỗi

➔ Trước khi sử dụng các lệnh xử lý chuỗi, ta phải xác định hướng xử lý chuỗi bằng cách set hay clear cờ hướng.

Lệnh đặt cờ hướng :

CLD : xóa cờ hướng, chuỗi được xử lý từ trái → phải

STD : đặt cờ hướng, chuỗi được xử lý từ phải → trái

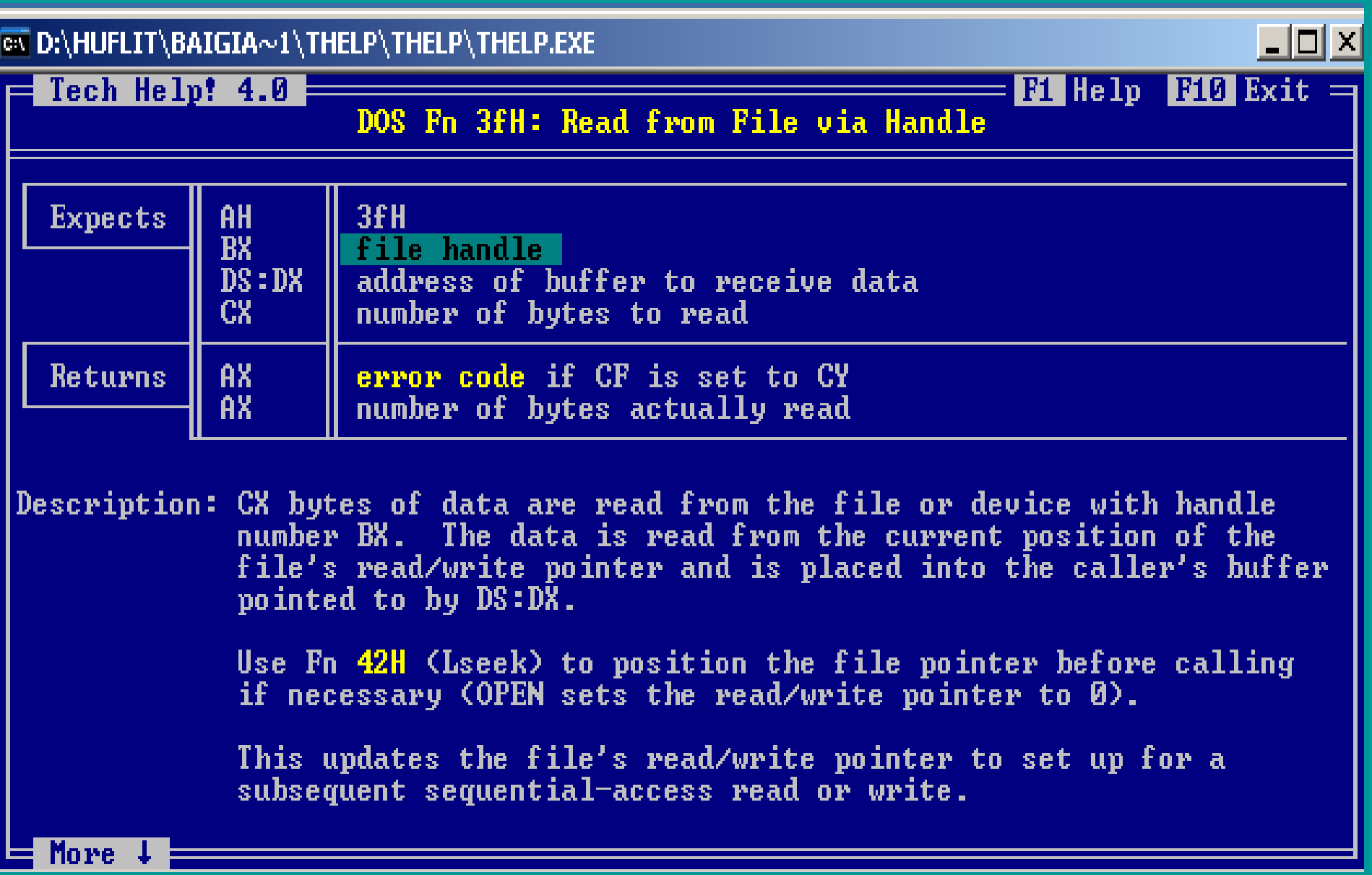


CÁC THAO TÁC XỬ LÝ CHUỖI

NHẬP CHUỔI

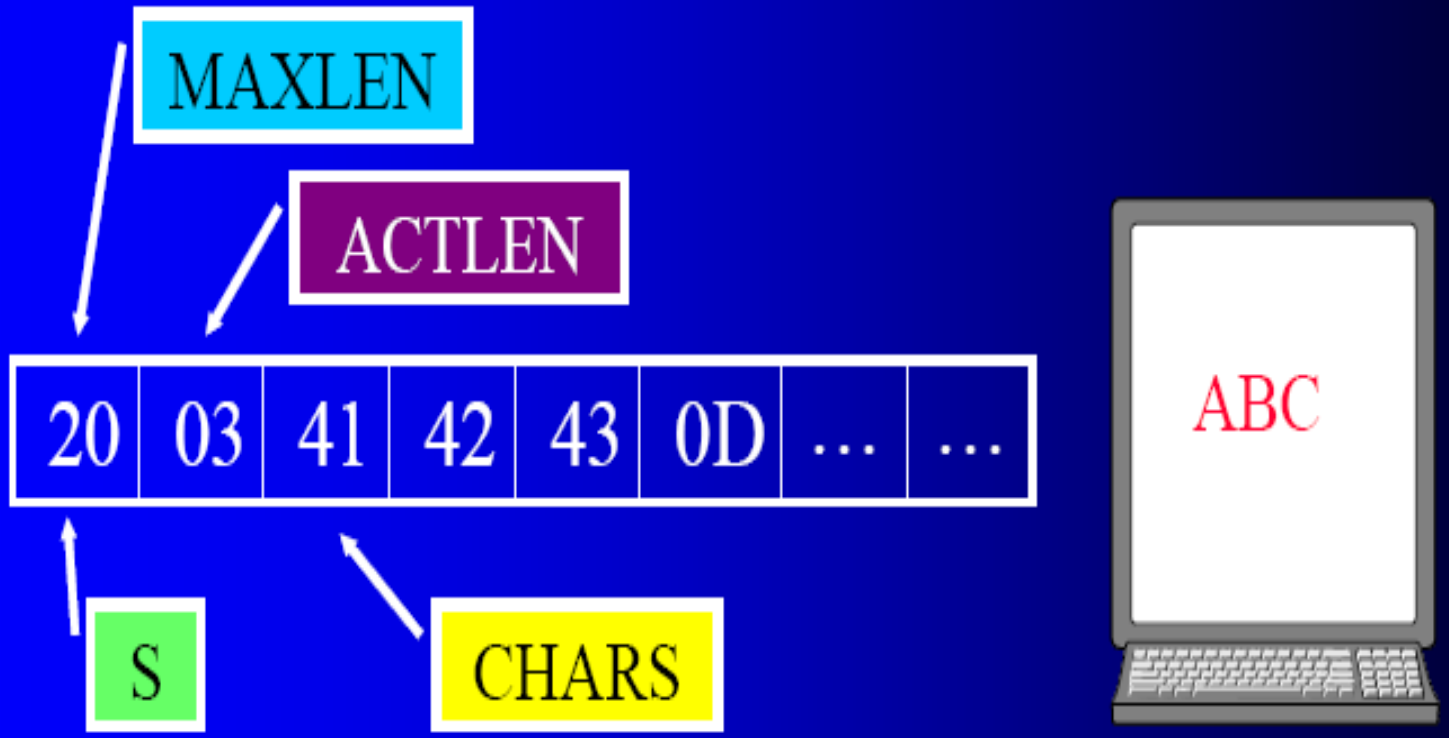
Input : AH = 0AH, ngắt 21H
DS:DX = địa chỉ của buffer, trong đó buffer[0] là kích thước tối đa của chuỗi,
buffer[1] sẽ là kích thước dữ liệu nhập.
Output : Chuỗi buffer chứa nội dung nhập vào từ buffer[2] trở đi

Yêu cầu xem thêm các chức năng AH = 3FH và AH = 40H của ngắt 21H.


D:\HUFLIT\BAIGIA~1\THELP\THELP\THELP.EXE
Tech Help! 4.0
F1 Help
F10 Exit
DOS Fn 3fH: Read from File via Handle
Expects
AH
3fH
BX
file handle
DS:DX
address of buffer to receive data
CX
number of bytes to read
Returns
AX
error code if CF is set to CY
AX
number of bytes actually read
Description: CX bytes of data are read from the file or device with handle number BX. The data is read from the current position of the file's read/write pointer and is placed into the caller's buffer pointed to by DS:DX.
Use Fn 42H (Lseek) to position the file pointer before calling if necessary (OPEN sets the read/write pointer to 0).
This updates the file's read/write pointer to set up for a subsequent sequential-access read or write.
More ↓

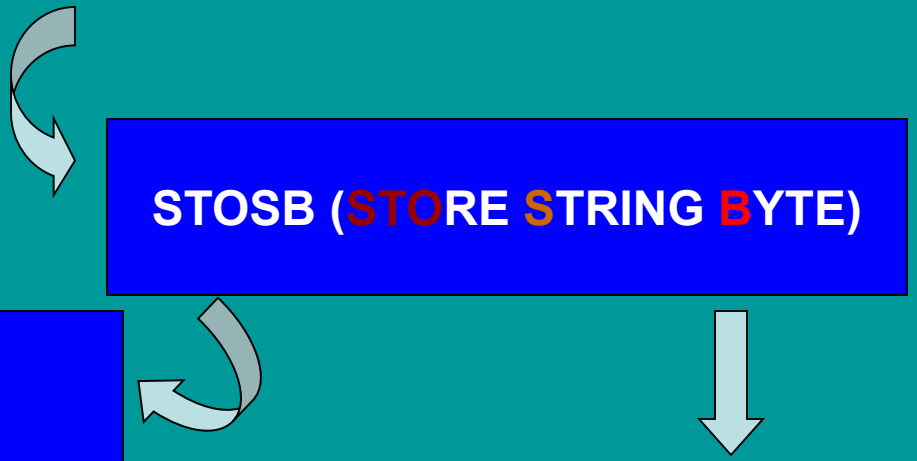
NHẬP CHUỖI

- Hàm 0Ah, INT 21h: Nhập chuỗi từ bàn phím, kết thúc Enter



NHẬP CHUỖI

Ta cũng có thể dùng hàm 1 INT 21h đọc 1 ký tự từ bàn phím để nhập 1 chuỗi bằng cách dùng vòng lặp và lưu chuỗi bằng lệnh STOSB.



NHẬP CHUỖI

Ta cũng có thể dùng hàm 1 Int 21h đọc 1 ký tự từ bàn phím để nhập 1 chuỗi bằng cách dùng vòng lặp và lưu chuỗi bằng lệnh STOSW.

THÍ DỤ

```
.MODEL SMALL
.STACK 100H
.DATA
  STRING1 DB 'HELLO'
  .CODE
 MAIN PROC
  MOV AX,@DATA
  MOV ES,AX
  LEA DI, STRING1      ; khởi tạo ES
  CLD
 MOV AL,'A'            ; xử lý từ trái ➔ phải
 STOSB                 ; AL chứa ký tự cần lưu
 STOSB                 ; lưu ký tự ‘A’
MOV AH,4CH             ; lưu ký tự thứ 2
INT 21H
MAIN ENDP
END MAIN
```

THÍ DỤ

```
READSTR PROC
PUSH AX
PUSH DI
CLD
XOR BX,BX
MOV AH,1
INT 21H
LAP:
        CMP AL,0DH
        JE ENDLAP
        CMP AL,8H
        JNE ELSE1
        DEC DI
        DEC BX
        JMP READ
ELSE1 :
        STOSB
        INC BX
READ :
        INT 21H
        JMP LAP
ENDLAP :
        POP DI
        POP AX
RET
READSTR ENDP
```

Giải thích :
DI chứa offset của chuỗi
BX chứa số ký tự nhập
8H mã ASCII của Backspace
không → lưu nó vào chuỗi
tăng số ký tự lên 1
Đúng → lùi con trỏ DI
giảm số ký tự nhập được

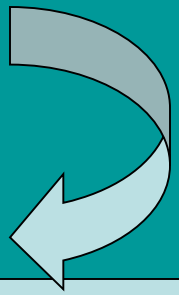
NHẬP XUẤT CHUỖI

HiỂN THỊ CHUỖI

AH = 09, ngắt 21H
Vào : DX = địa chỉ offset của chuỗi.
Chuỗi phải kết thúc bằng kí tự '$'.
Chú ý : thay vì dùng lệnh MOV OFFSET ta có thể dùng lệnh LEA.

CÁC THAO TÁC XỬ LÝ CHUỖI

HiỂN THỊ CHUỖI



Nạp 1 chuỗi

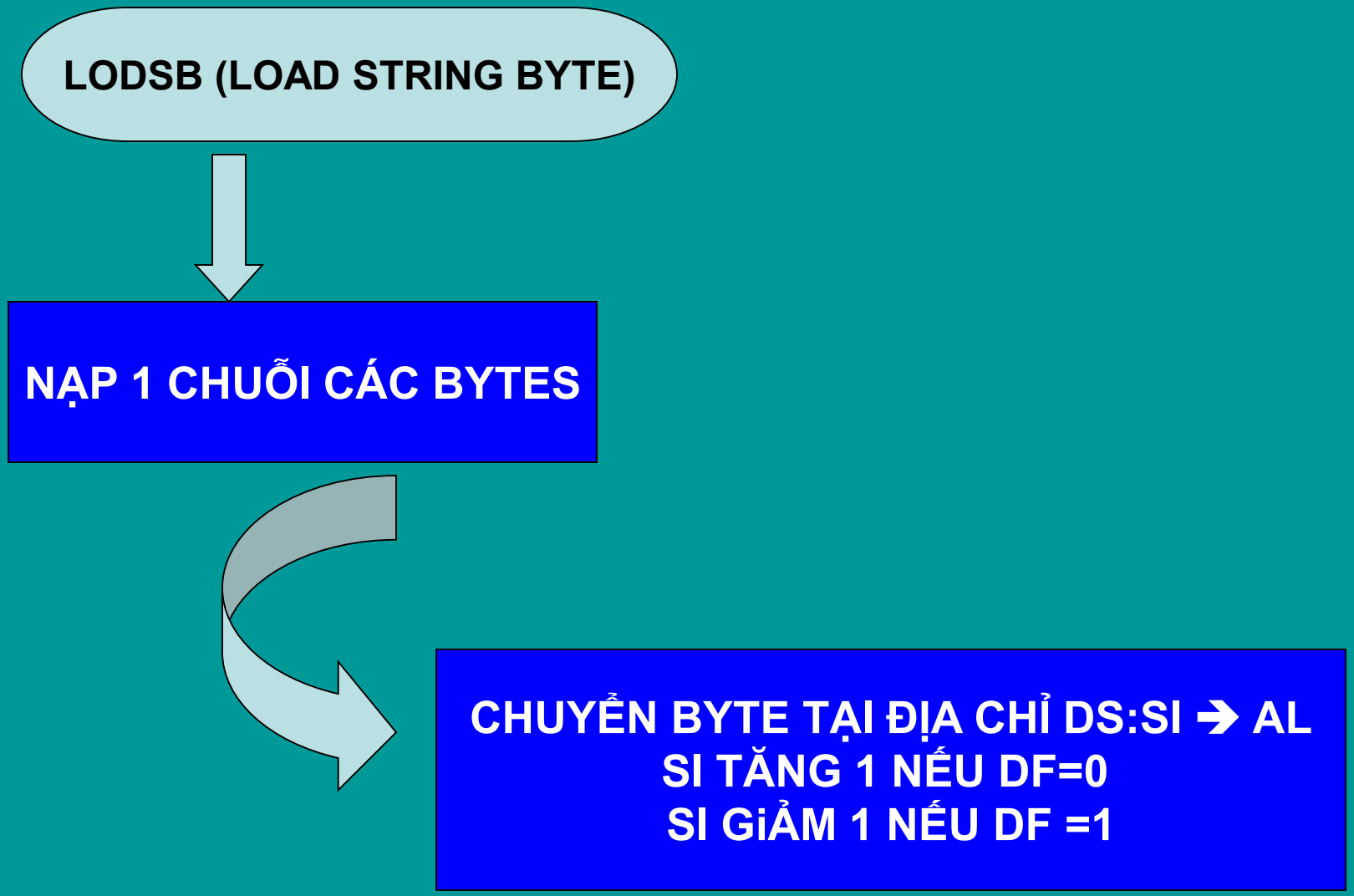
For counter Do

Nạp chuỗi cần hiển thị vào AL

Chuyển vào DL

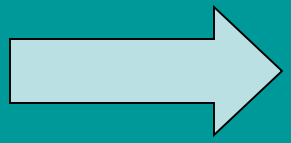
Hiển thị ký tự

EndFor

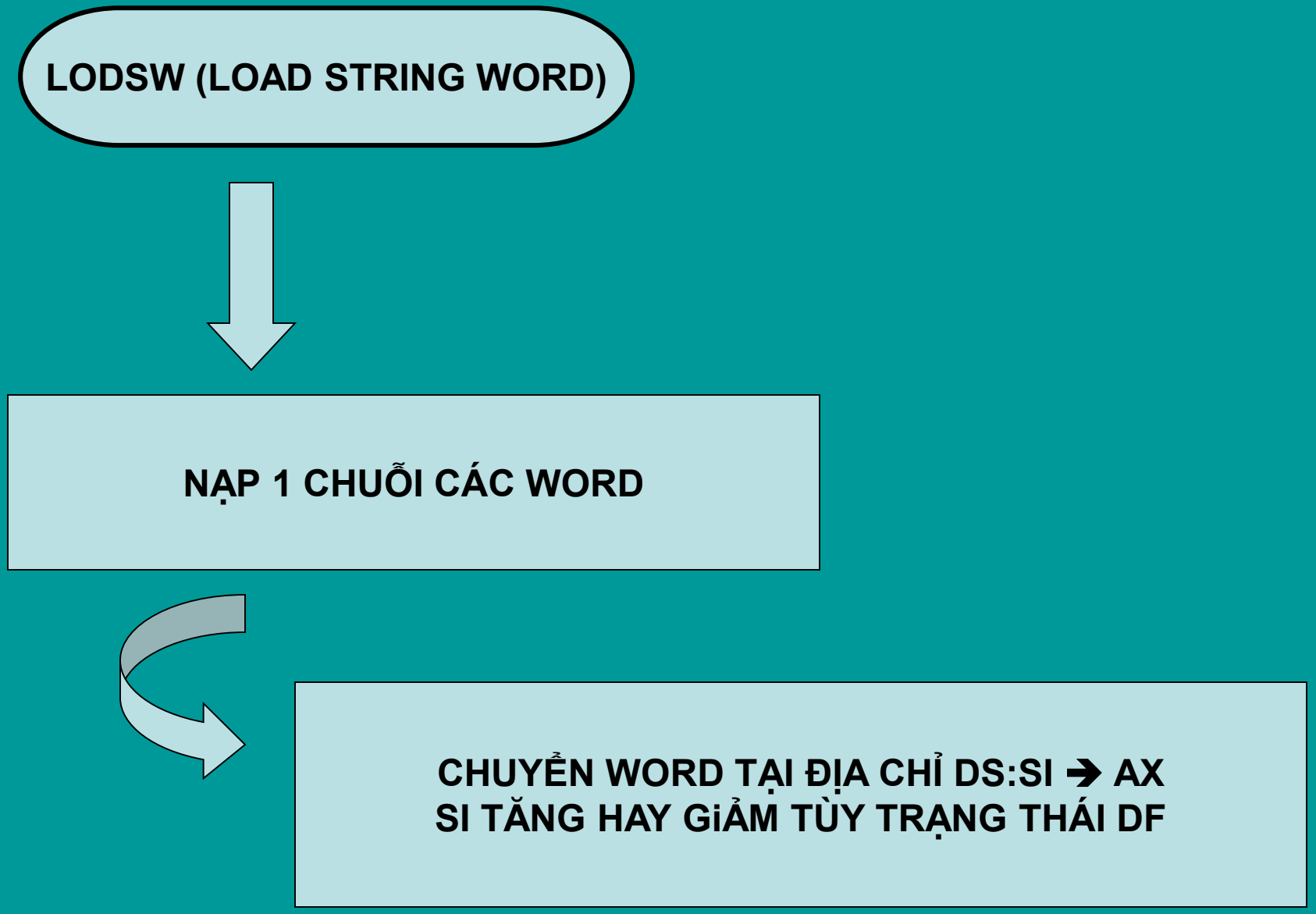

LODSB (LOAD STRING BYTE)
NẠP 1 CHUỖI CÁC BYTES
CHUYỂN BYTE TẠI ĐỊA CHỈ DS:SI ➔ AL
SI TĂNG 1 NẾU DF=0
SI GiẢM 1 NẾU DF =1

THÍ DỤ

```
STRING1 DB ‘ABC’
MOV AX,@DATA
MOV DS,AX
LEA SI, STRING1
CLD
LODSB
LODSB
…….
```



NẠP BYTE THỨ 1 VÀ THỨ 2 ➔ AL


LODSW (LOAD STRING WORD)
NẠP 1 CHUỖI CÁC WORD
CHUYỂN WORD TẠI ĐỊA CHỈ DS:SI ➔ AX
SI TĂNG HAY GiẢM TÙY TRẠNG THÁI DF

THÍ DỤ

Hiển thị chuỗi nhập

```
DISPSTR PROC
PUSH AX
PUSH BX
PUSH CX
PUSH DX
PUSH SI
MOV CX, BX
JCXZ EXIT
CLD
MOV AH,2
LAP :
LODSB
MOV DL, AL
INT 21H
LOOP LAP
```

```
EXIT :
POP SI
POP DX
POP CX
POP BX
POP AX
RET
DISPSTR ENDP
```

CHƯƠNG TRÌNH HÒAN CHỈNH

Viết chương trình nhập 1 chuỗi ký tự tối đa 80 ký tự, hiển thị 15 ký tự của chuỗi đã nhập ở dòng kế.

```
.MODEL SMALL
.STACK 100H
.DATA
STRING1 DB 80 DUP(0)
XDONG DB 0DH,0AH,'$'
.CODE
MAIN PROC
MOV AX,@DATA
MOV DS,AX

MOV ES,AX
LEA DI, STRING1
CALL READSTR
LEA DX,XDONG
MOV AH,9
INT 21H
```

```
LEA SI, STRING1
MOV BX, 15
CALL DISPSTR
MOV AX,4C00H
INT 21H
MAIN ENDP
; READSTR PROC
   ……..
; DISPSTR PROC
……..
END MAIN
```

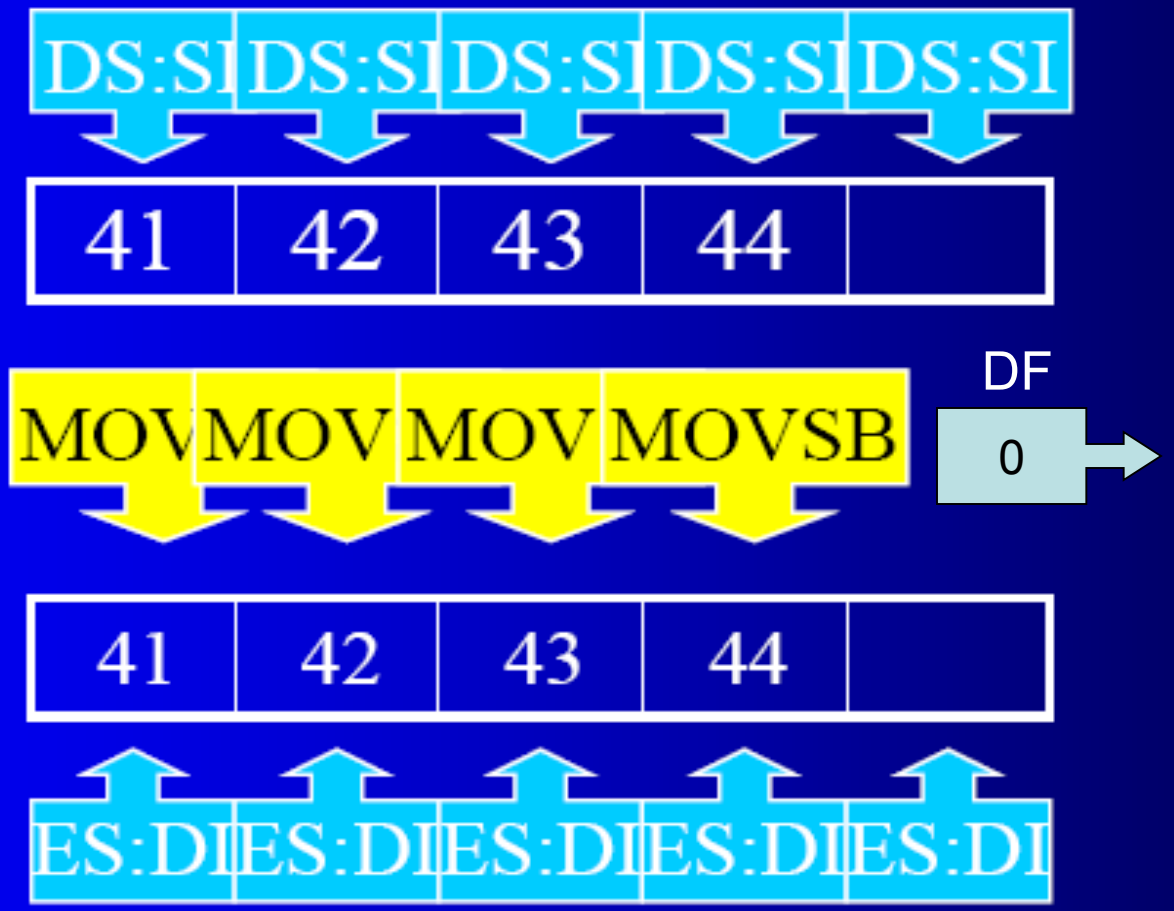
CÁC THAO TÁC XỬ LÝ CHUỖI

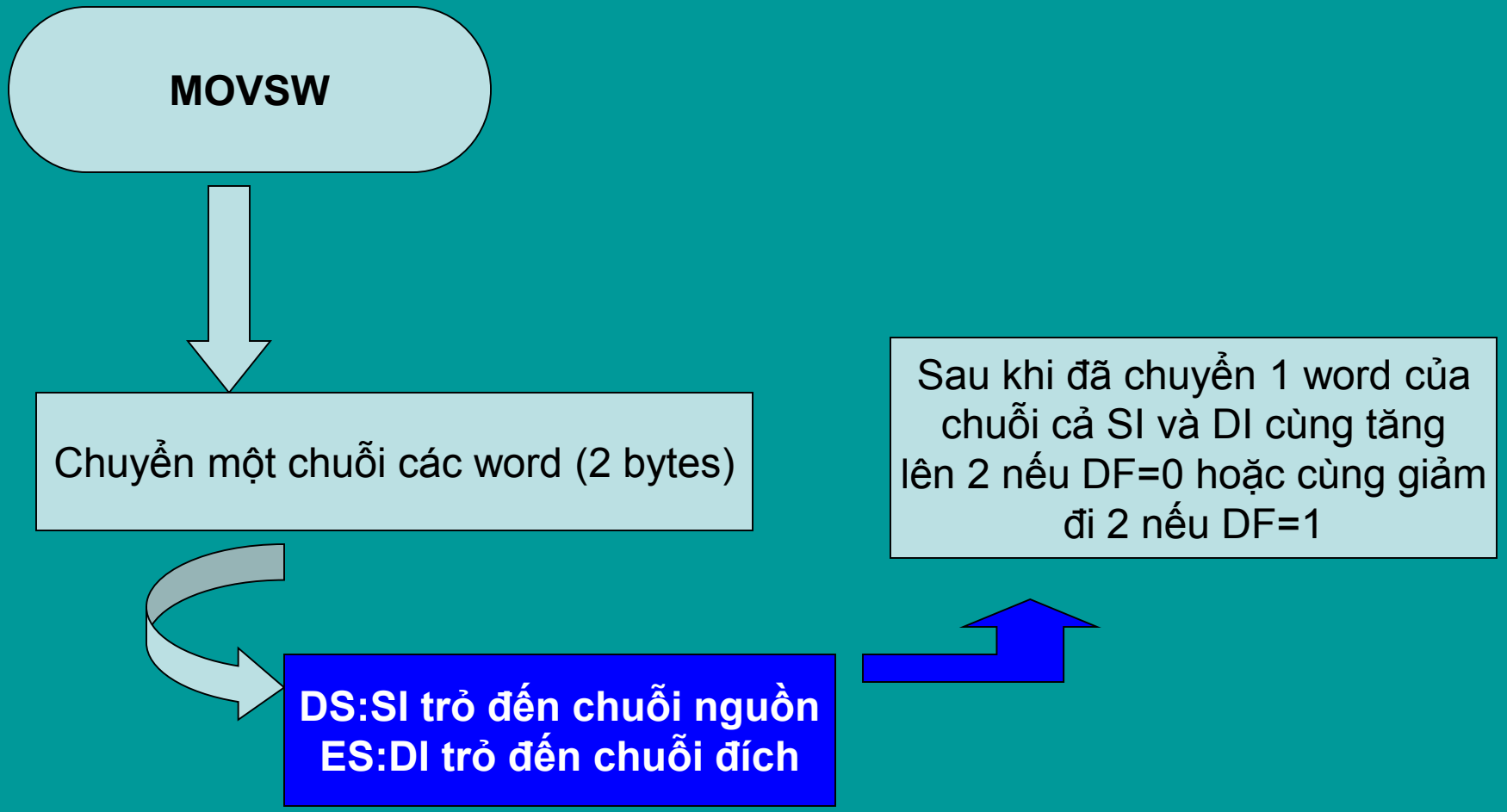
Chuyển một BYTE : MOVSB

chuyển nội dung của byte được định bởi DS:SI đến byte được chỉ bởi ES: DI.
Sau đó SI và DI tự động tăng lên 1 nếu cờ DF = 0
hay giảm 1 nếu DF = 1.

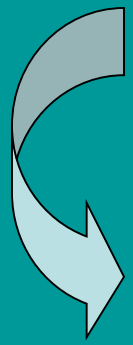
MOVSB chỉ chuyển 1 byte. Vậy cả chuỗi ta làm thế nào ?

- MOVSB

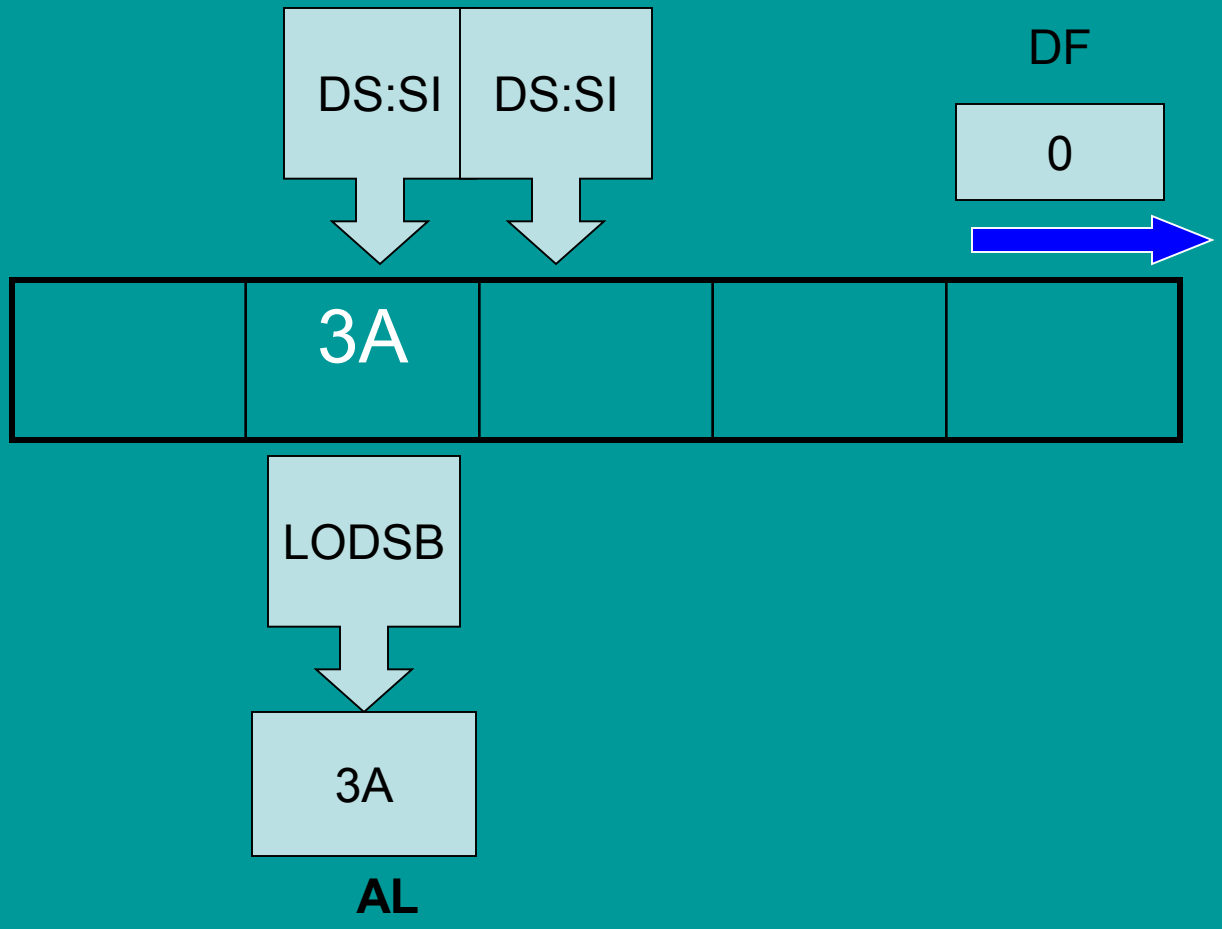

DS:SI DS:SI DS:SI DS:SI DS:SI
41 42 43 44
MOV MOV MOV MOVSB
DF
0
41 42 43 44
ES:DI ES:DI ES:DI ES:DI ES:DI


MOVSW
Chuyển một chuỗi các word (2 bytes)
DS:SI trỏ đến chuỗi nguồn
ES:DI trỏ đến chuỗi đích
Sau khi đã chuyển 1 word của chuỗi cả SI và DI cùng tăng lên 2 nếu DF=0 hoặc cùng giảm đi 2 nếu DF=1

LODSB (Load String Byte)



Chuyển byte chỉ bởi DS:SI ➔ AL
tăng SI lên 1 nếu DF=0
giảm SI xuống 1 nếu DF=1

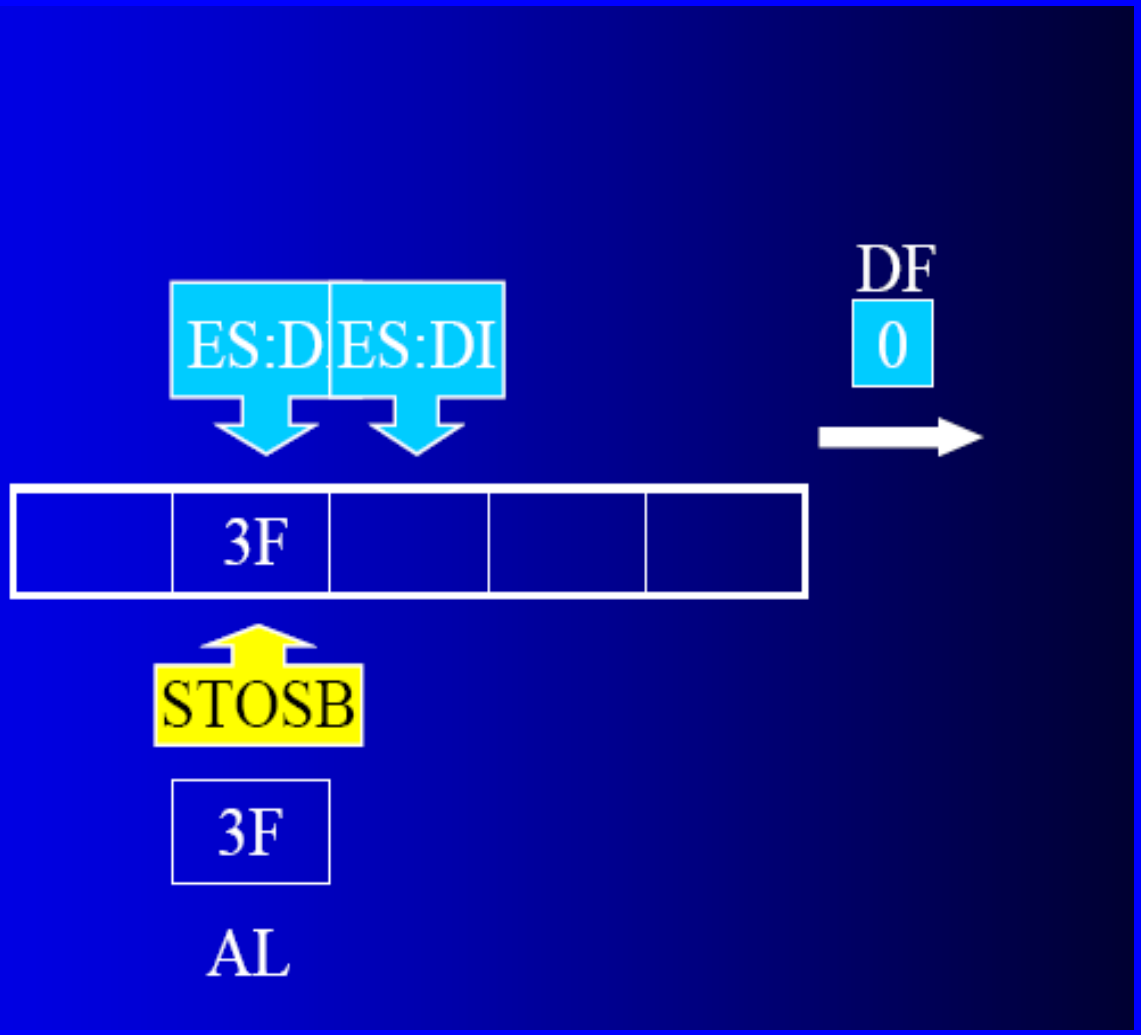

DS:SI
DS:SI
DF
0
3A
LODSB
3A
AL

- MOVSW


DS:SI
DS:SI
DS:SI
41
42
43
44
MOVSW
MOVSW
DF = 0
41
42
43
44
ES:DI
ES:DI
ES:DI

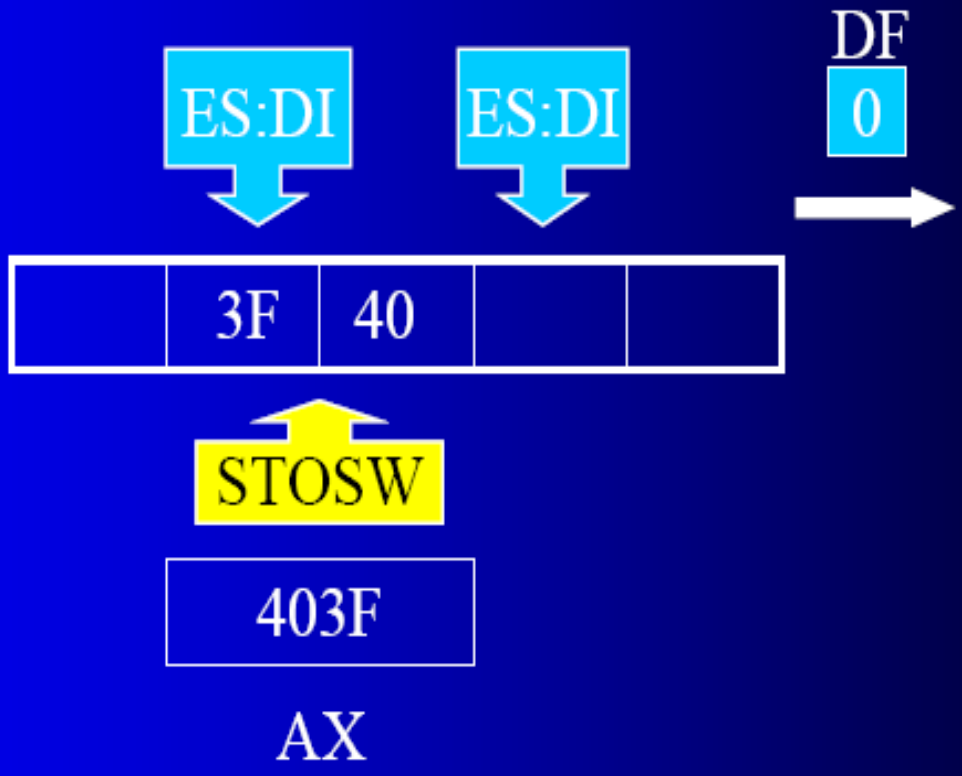
STOSB (LƯU CHUỖI BYTE)

- STOSB

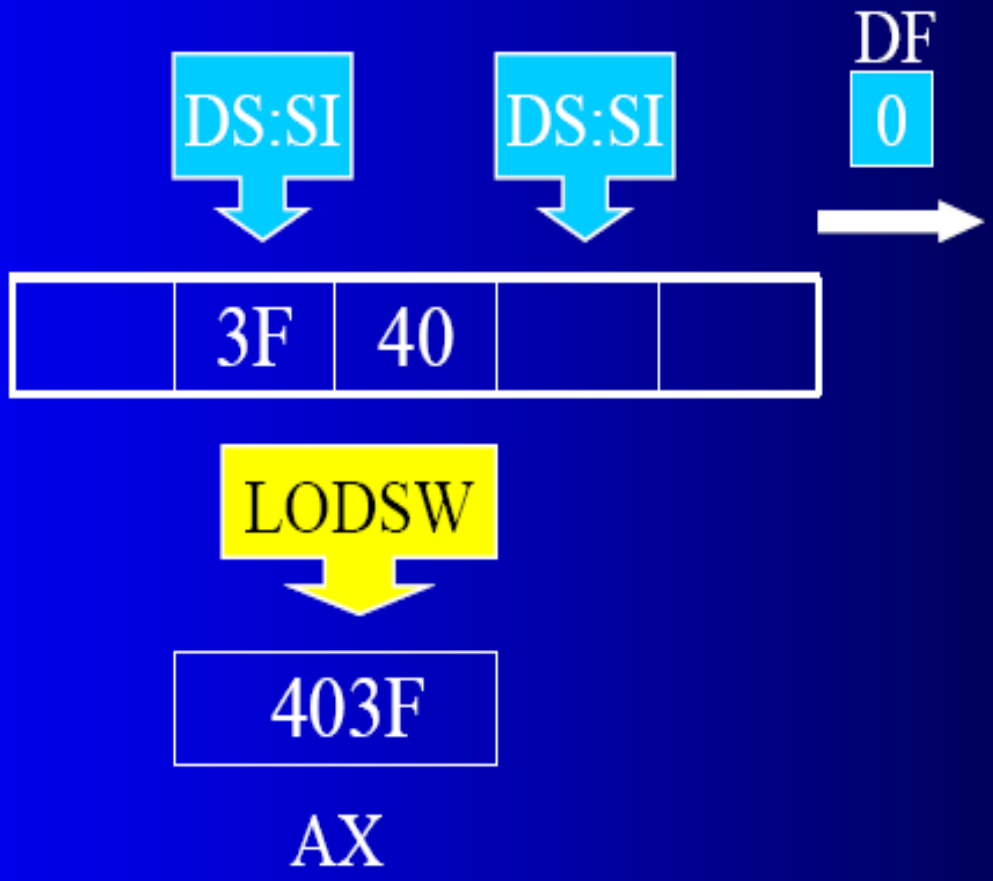


STOSW (LƯU CHUỖI WORD)

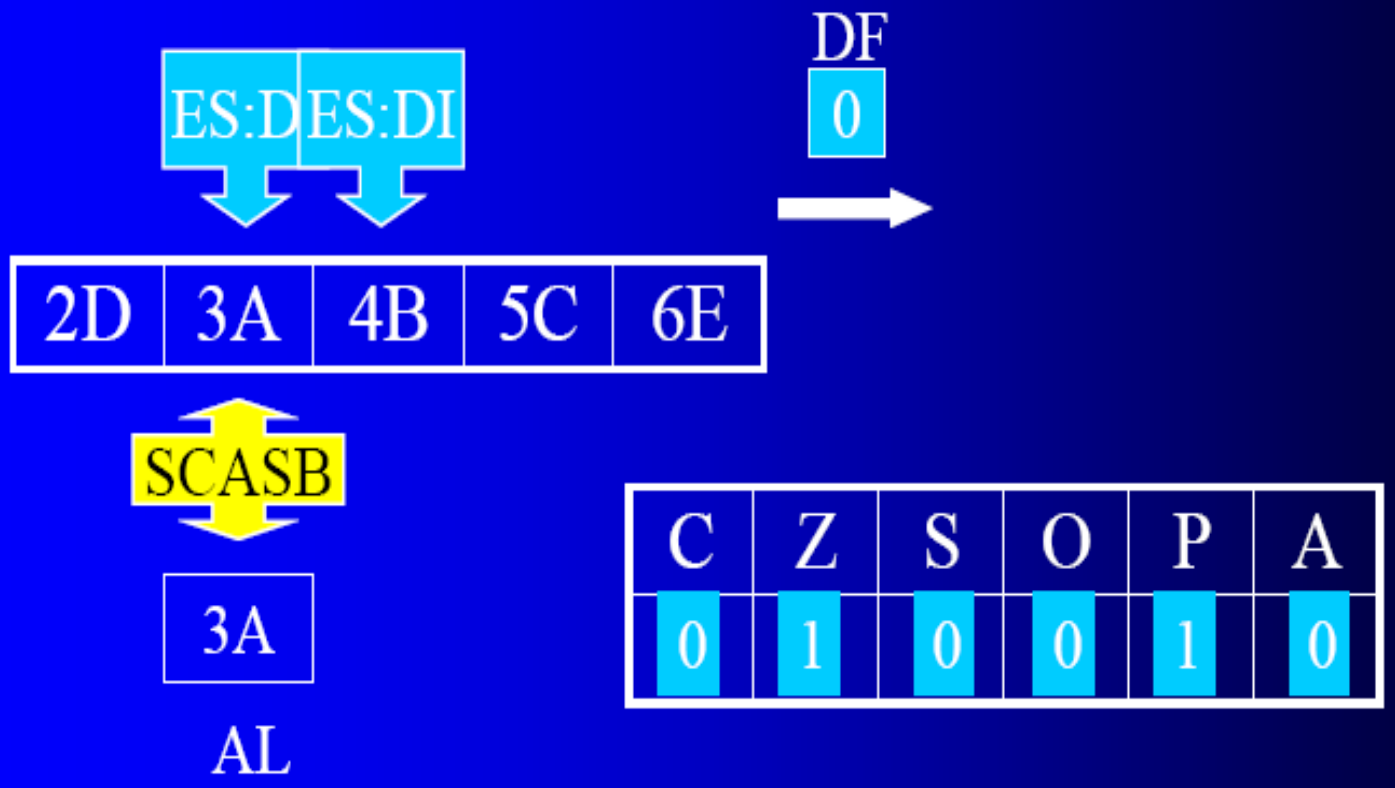
- STOSW



- LODSW

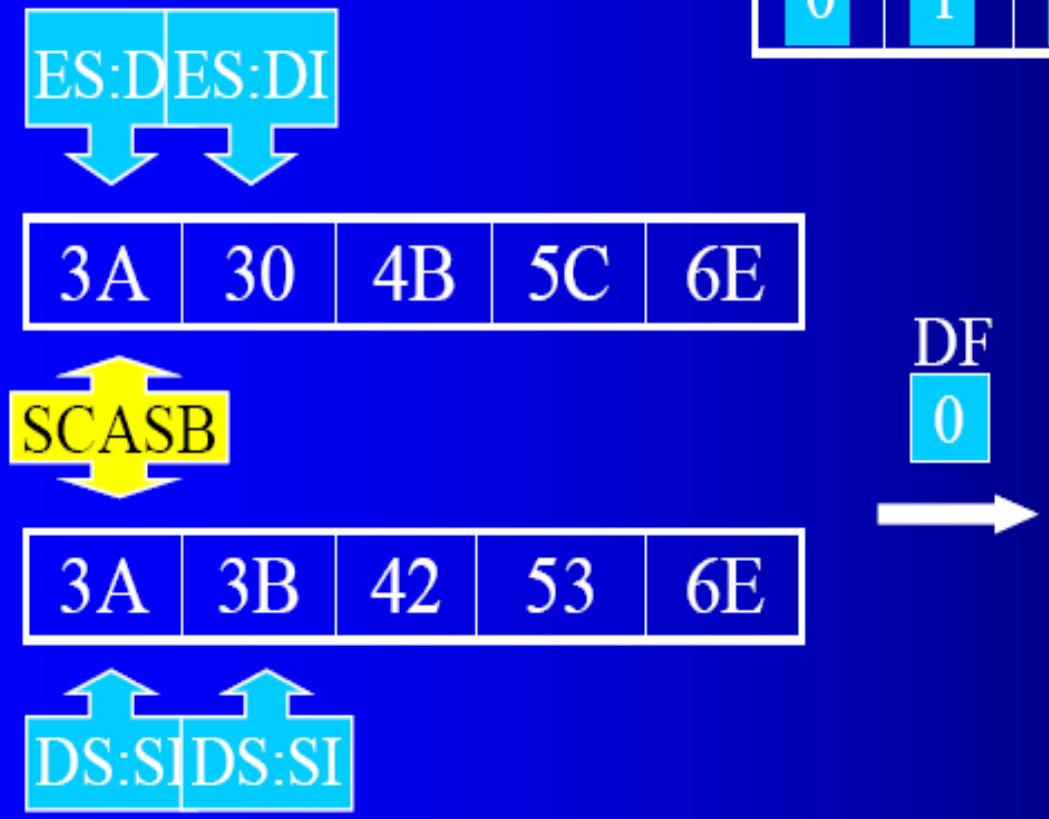


- SCASB

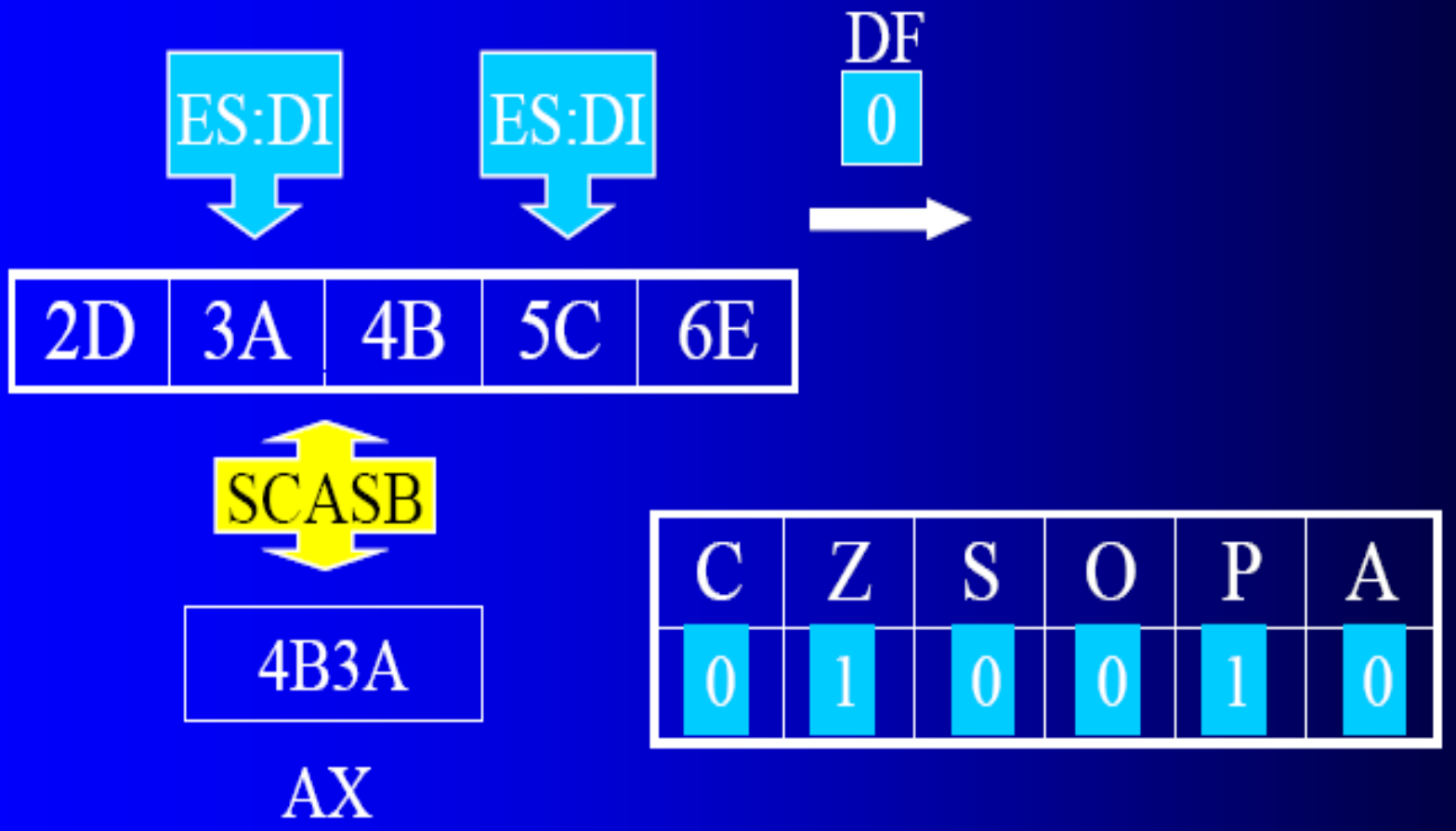


- CMPSB

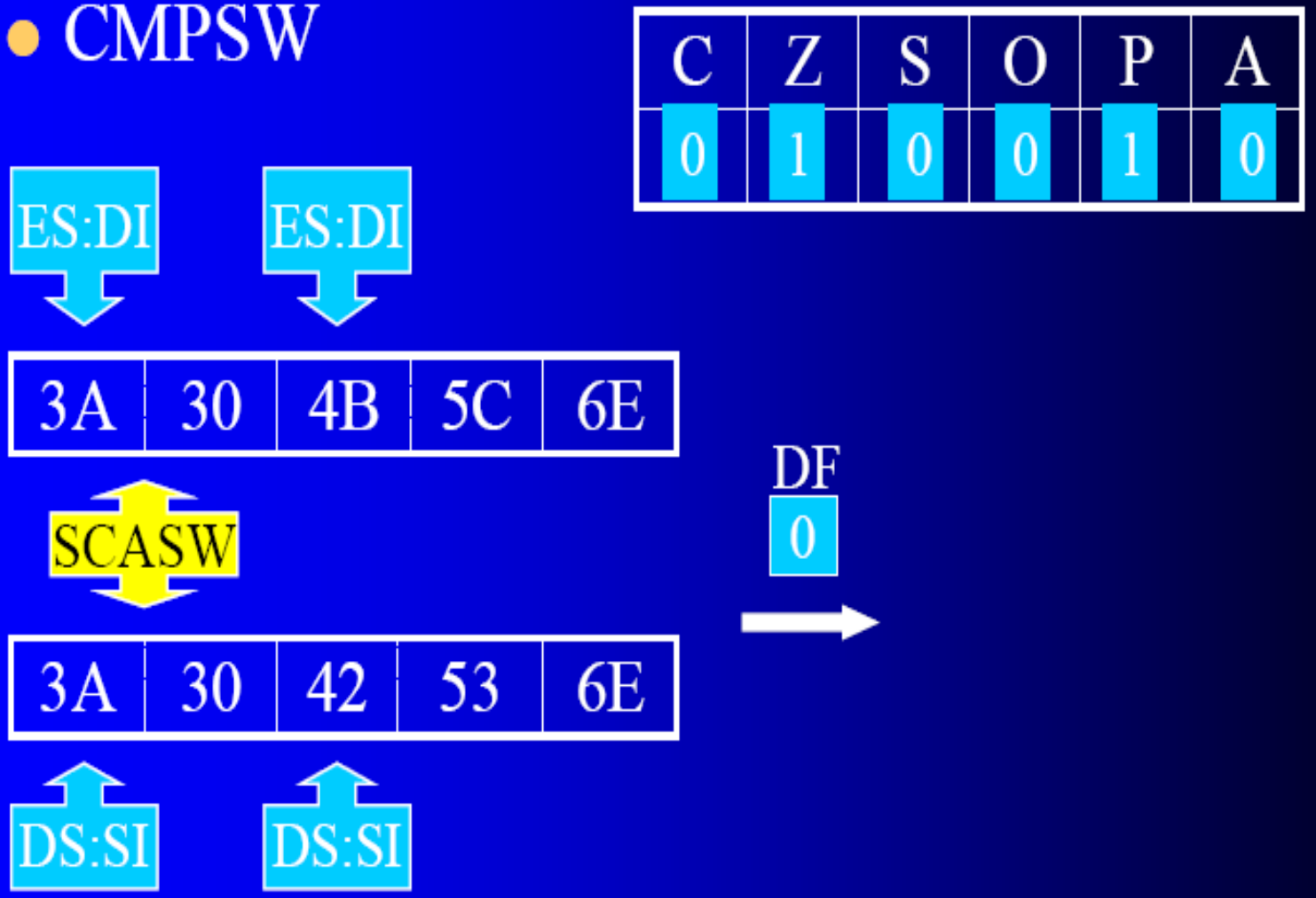
C	Z	S	O	P	A
0	1	0	0	1	0

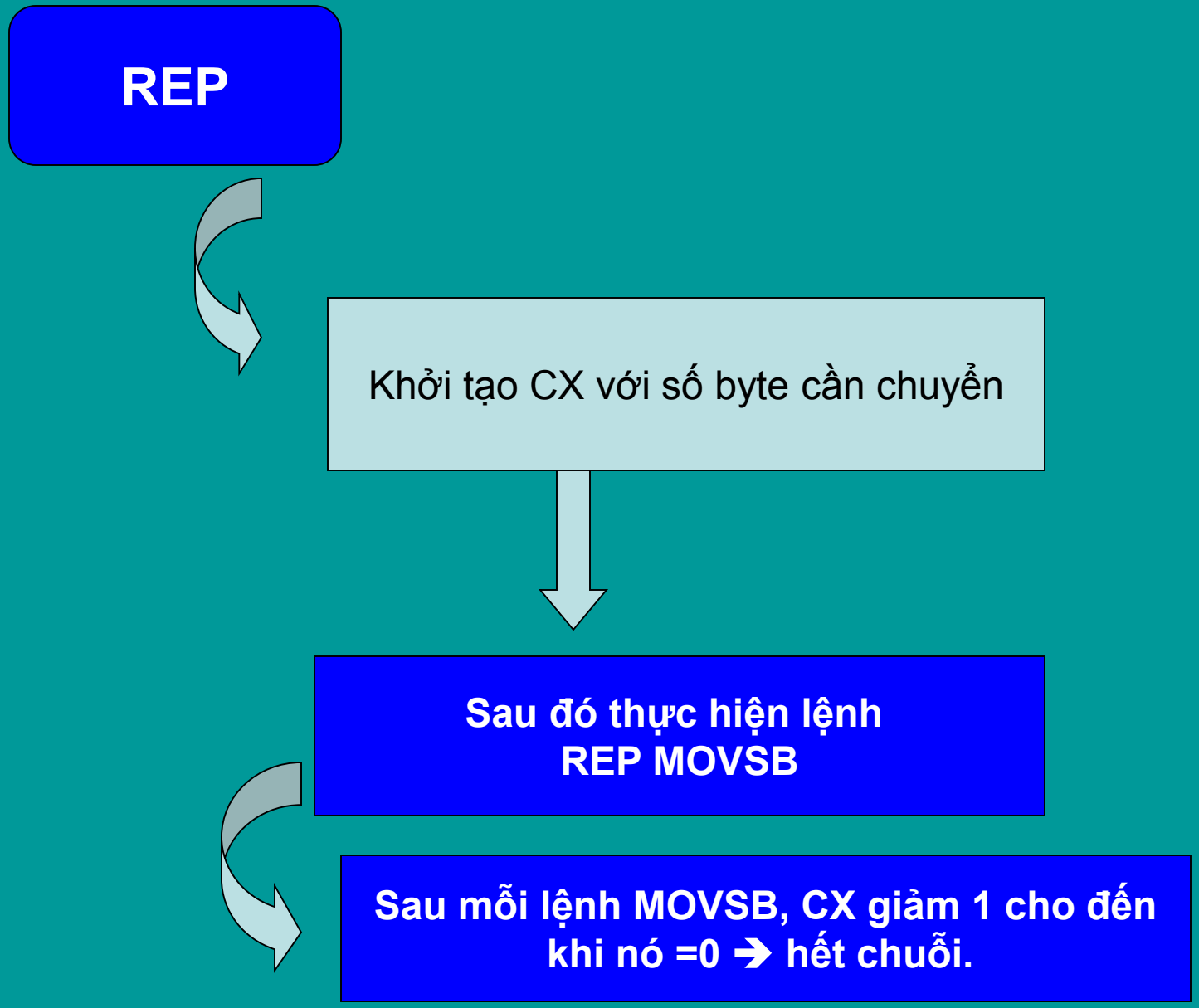


- SCASW



- CMPSW


C Z S O P A
0 1 0 0 1 0
ES:DI
ES:DI
3A 30 4B 5C 6E
SCASW
DF
0
3A 30 42 53 6E
DS:SI
DS:SI


REP
Khởi tạo CX với số byte cần chuyển
Sau đó thực hiện lệnh
REP MOVSB
Sau mỗi lệnh MOVSB, CX giảm 1 cho đến khi nó =0 ➔ hết chuỗi.

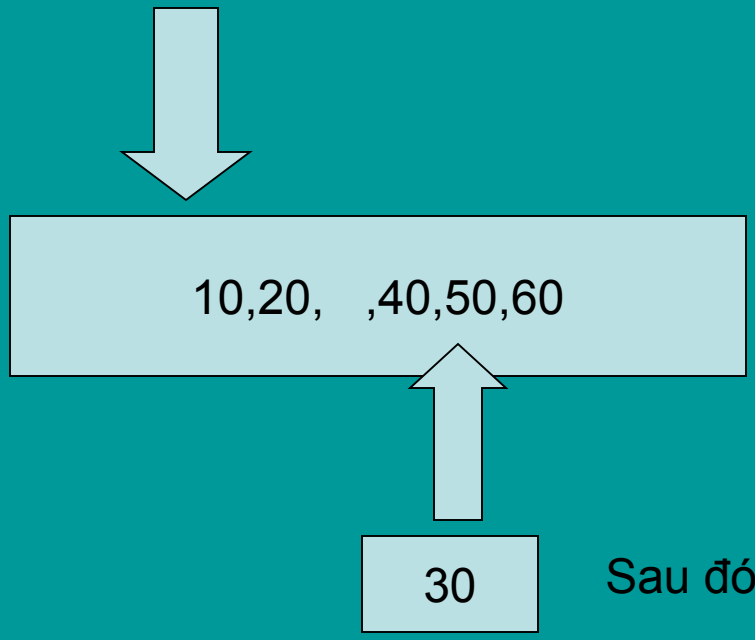
THÍ DỤ MINH HỌA

```
.DATA
STRING1 DB ‘HELLO’
STRING2 DB 5 DUP(?)
…..
CLD
LEA SI, STRING1
LEA DI, STRING2
MOV CX, 5
REP MOVSB
……..
```

Bài tập :
Viết đoạn chương trình chép chuỗi STRING1 ở thí dụ trước vào chuỗi STRING2 nhưng theo thứ tự ngược lại.

THÍ DỤ MINH HỌA

Cho mảng sau
ARR DW 10,20,40,50,60,?
Viết các lệnh để chèn 30 vào giữa 20 và 40 (giả sử rằng DS và ES đã chứa địa chỉ đoạn dữ liệu)



```
STD
LEA SI, ARR+8H
LEA DI, ARR+AH
MOV CX, 3
REP MOVSW
MOV WORD PTR[DI],30
```

MẢNG 1 CHIỀU

Một dãy các phần tử có cùng kiểu dữ liệu, có cùng 1 tên gọi.

Khai báo

MKT DB 'abcdef' ; mảng ký tự

MNB Dw 10h,20h,30h,40h,50h,60h ; mảng số

ArrA DB 100 DUP(0) ; khai báo mảng có 100 phần tử có giá trị khởi tạo bằng 0.

Các chương trình con

READSTRING - (Đọc tập tin bàn phím)
Vào : DS:DX = địa chỉ đệm nhận;
CX = số ký tự nhận tối đa.
Ra : DS:DX = địa chỉ chuỗi ASCIIZ.

STRLEN - Lấy chiều dài chuỗi ASCIIZ.
Vào : ES:DI = địa chỉ chuỗi ASCIIZ.
Ra : AX = chiều dài chuỗi không kể số 0.

WRITESTRING - Xuất một chuỗi ra màn hình.
Vào : DS:DX= địa chỉ chuỗi ASCIIZ.
Ra : Không.

STRCOPY - Chép chuỗi nguồn sang chuỗi đích.
Vào : DS:SI = địa chỉ chuỗi nguồn.
ES:DI = địa chỉ chuỗi đích.
Ra : Không.

Các chương trình con (tt)

STRCOMP - So sánh chuỗi và lập cờ.
Vào : DS:SI = địa chỉ chuỗi nguồn.
ES:DI = địa chỉ chuỗi đích.
Ra : thay đổi cờ.
CY = 1 : chuỗi nguồn < chuỗi đích.
ZF = 1 : chuỗi nguồn = chuỗi đích.

STRCHR - tìm ký tự trong chuỗi.
Vào : ES:DI = địa chỉ chuỗi.
AL = ký tự cần tìm.
Ra : CY = 0 : tìm thấy ký tự.
ES:DI = địa chỉ vị trí xuất hiện đầu tiên.
CY = 1 : không tìm thấy ký tự trong chuỗi.

STRSTR - tìm chuỗi trong chuỗi.
Vào : DS:SI = địa chỉ chuỗi nguồn.
DX = chiều dài chuỗi nguồn.
ES:DI = địa chỉ chuỗi đích.
BX = chiều dài chuỗi đích.
Ra : CY = 0 : tìm thấy
ES:DI = địa chỉ vị trí xuất hiện đầu tiên.
CY = 1 : không tìm thấy.

BÀI TẬP

Bài 1 : Viết chương trình nhập 1 số từ 1-12, in ra tên tháng tương ứng.

Bài 2 : Viết chương trình nhập 1 số từ 1-7, in ra tên thứ tương ứng.

MỘT SỐ BÀI TẬP MINH HỌA LẬP TRÌNH XỬ LÝ CHUỖI

Nhập 1 chuỗi dài tối đa 255 ký tự từ bàn phím. Cho phép dùng phím BackSpace để sửa khi nhập sai và kết thúc nhập khi gỏ phím Enter.

Hướng dẫn :

Dùng hàm 0AH INT 21H để nhập chuỗi
DS:DX địa chỉ của buffer đệm lưu chuỗi.
Byte 0 : số byte tối đa có thể nhập.
Byte 1 : chứa giá trị 0
Byte 2 trở đi : để trống (lưu các ký tự sẽ nhập)

Để nhập 1 chuỗi ký tự vào Buffer đệm ta khai báo như sau :

```
.DATA
BUFFERN DB 80,0,80 DUP(?)
```

B1. Viết chương trình nhập vào 1 từ, sau đó in từng ký tự trong từ theo chiều dọc.

Thí dụ Nhập CONG

Xuất : C
O
N
G

B2. Viết chương trình nhập vào 1 chuỗi, sau đó đổi tất cả chuỗi thành chữ hoa và in chuỗi ra màn hình ở dòng kế.

B3.Viết chương trình nhập hai chuỗi ký tự , kiểm tra xem chuỗi thứ hai có xuất hiện trong chuỗi thứ nhất hay không.

Ví dụ : Nhập chuỗi thứ nhất : computer information

Nhập chuỗi thứ hai : compute

Xuất: Chuỗi thứ hai có xuất hiện trong chuỗi thứ nhất.

B4. Viết chương trình nhập 1 chuỗi ký tự viết hoa các ký tự nguyên âm, viết thường các ký tự phụ âm.

Ví dụ : Nhập chuỗi : “aBcdE”

Xuất chuỗi: “AbCdE”

B5. Viết chương trình nhập vào 2 chuỗi ký tự s1, s2 và 1 số nguyên dương n. Chèn chuỗi s2 vào chuỗi s1 ở vị trí ký tự thứ n trong chuỗi s1 .

Ví dụ : Nhập chuỗi s1 : “abcde”

Nhập chuỗi s2 : “fgh”

Nhập n = 3

Xuất kết quả : “abcfghde”

B6. Viết chương trình nhập vào từ bàn phím 1 chuỗi và tính số lần xuất hiện của các nguyên âm (a,e,i,o,u, y), cac phu am, cac khoang trang, trong chuỗi tương ứng.

Ví dụ : Nhập chuỗi : “dai hoc khoa hoc tu nhien thanh pho ho chi minh”

Xuất : Số lần xuất hiện của các nguyên âm là : 14 , phu am la: 24, khoang trang la: 9

B7. Viết chương trình nhập vào từ bàn phím 1 chuỗi gồm các ký tự trong bảng chữ cái. Đếm xem trong chuỗi có bao nhiêu từ.

Ví dụ : Nhập chuỗi : “ hO Chi mINh ”

Xuất : chuỗi gồm có 3 từ

B8. Viết chương trình nhập vào từ bàn phím 4 số . Xuất ra màn hình 4 số đó theo thứ tự tăng dần .

Ví dụ : Nhập : 14 7 26 11

Xuất : 7 11 14 26

B9. Viết chương trình nhập vào từ bàn phím 4 số và sau đó xuất số lớn nhất và nhỏ nhất ra màn hình.

Ví dụ : Nhập : 13 21 1 49

Xuất : Số lớn nhất : 49

Số nhỏ nhất : 1

Viết chương trình nhập vào từ bàn phím chuỗi 1 (chuỗi dài), chuỗi 2 (chuỗi ngắn) và một ký tự. Sau đó, làm các công việc sau :

- Tìm chuỗi 2 trong chuỗi 1 và in ra vị trí xuất hiện đầu tiên của chuỗi 2 trong chuỗi 1 nếu tìm thấy. Ngược lại in ra không tìm thấy.
- Tìm ký tự đã nhập trong chuỗi 1 và in ra vị trí xuất hiện đầu tiên của ký tự nếu tìm thấy.Ngược lại in ra không tìm thấy.
- Thay chuỗi 2 trong chuỗi 1 bằng ký tự (nếu được).