

تمرين 1:

يطلب كتابة تمرين يقوم بتخزين عشرين قيمة 0CDH في موقع الذاكرة بدءاً من 0200H ومن ثم نقلها إلى مجال العناوين بدءاً من 0240H.

ORG 100H

CALL FILL

CALL MOVE

INT 5H

;

FILL:

; load source into ds:si,

; load target into es:di:

; load AL with value to be stored:

MOV SI, 0200H

MOV DI, 0200H

MOV AL, 0CDH

MOV CX, 20H

REP STOSW

RET

MOVE:

MOV SI, 0200H

MOV DI, 0240H

MOV CX, 20H

REP MOVSW

RET

يتم وضع القيم في البداية للتخزين ضمن المقطع SI ومن ثم سيتم نقلها إلى المقطع DI

يتم تحميل كل من SI, DI بالعنوان الأول لبدء التخزين ووضع القيمة المطلوبة ضمن المسجل AX وتحميل CX بقيمة العداد (وهنا هو عشرين رقم).

تقوم تعليمة STOSW بتخزين القيم بصورة تلقائية بنسخ القيم الموجودة في AX إلى موقع SI, DI وانقاص العداد بصورة تلقائية

يتم تحميل كل من SI, DI بالعنوان الأول لبدء التخزين ووضع القيمة المطلوبة ضمن المسجل AX وتحميل CX بقيمة العداد (وهنا هو عشرين رقم). انتبه هنا أنه يتم النقل إلى العنوان 240 والذي تم وضعه ضمن DI

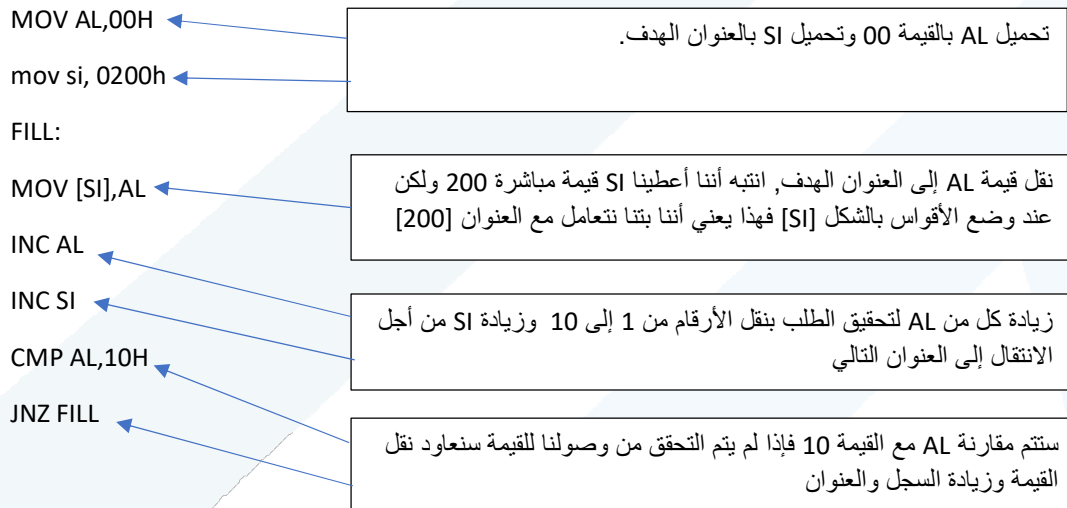
تقوم تعليمة MOVSW بنقل القيم بصورة تلقائية بنسخ القيم الموجودة في SI إلى موقع DI, وانقاص العداد بصورة تلقائية

تمرين 2:

يطلب القيام بتخزين الأرقام من 1 إلى 10 بدءاً من موقع الذاكرة 0200

الحل:

بما أن التخزين يتم في موقع ذاكرة وهذا الموقع هو 200 عندها يجب تحميل هذا العنوان إلى المسجل SI وبما أننا نريد أن نقوم بالتخزين بصورة متتالية لذلك حملنا AL بالقيمة 00H ومن ثم سنقوم بنقلها إلى الموقع المحدد وبعد ذلك نزيد على قيمة AL وعلى العنوان ونقل القيم بصورة تكرارية إلى أن نصل للقيمة الأخيرة والتي تتم بعملية المقارنة في نهاية البرنامج.

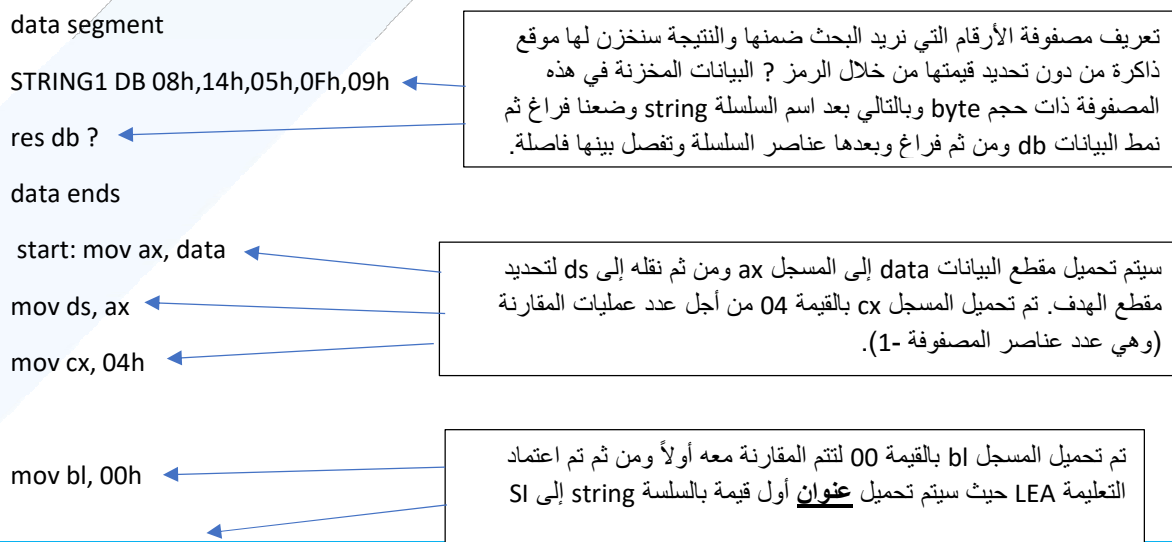


تمرين 3:

يطلب كتابة برنامج يقوم بتخزين مصفوفة من الأرقام في الذاكرة (مجال عناوينها غير محدد) ومن ثم إيجاد العدد الأكبر فيها.

الحل:

بما أنه لم يتم تحديد موقع الذاكرة والذي سيتم تخزين القيم ضمنه لذلك يجب أن نقوم بإدخال القيم كـ directive وتخزين موقع النتيجة بموقع ذاكرة ما آخر.



```
LEA SI, STRING1
```

```
up:
```

```
mov al, [SI]
```

```
cmp al, bl
```

```
jl nxt
```

```
mov bl, al
```

```
nxt:
```

```
inc si
```

```
dec cx
```

```
jnz up
```

```
mov res,bl
```

```
end start
```

تحميل محتوى العنوان [SI] إلى al وبالتالي سيتم تحميل أول قيمة في السلسلة string إلى المسجل AL

سيتم مقارنة al,bl فإذا كان al أقل من bl سيتم الانتقال إلى القيمة التالية في السلسلة أي القفز إلى nxt وإلا سيتم نقل قيمة al إلى bl لأنها أكبر ومن ثم مقارنة القيمة التالية. يتم انقاص cx وتكرار عملية المقارنة إلى أن تصل cx إلى الصفر

بعد انتهاء عمليات المقارنة سيتم تخزين القيمة bl والتي هي القيمة الأكبر في المسجل .res

تمرين 4:

يطلب كتابة برنامج يقوم بجداء قيمتين من نمط word وخزنتين في الذاكرة.

الحل:

بما أن القيم المخزنة من نمط كلمة فإنها تعرف بالشكل DW وبما أننا لم نذكر موقع التخزين لذا فإنها تعرف كموجه وتم تسمية المتحولين var1,var2 ومن ثم نقلهما إلى المسجلات ax,bx لإجراء عملية الجداء.

```
ORG 100h
```

```
.DATA
```

```
VAR_1 DW 12DAH
```

```
VAR_2 DW 3F24H
```

```
.CODE
```

```
MOV AX, VAR_1
```

```
MOV BX, VAR_2
```

```
MUL BX
```

```
RET
```

تمرين 5

ORG 100h

.DATA

VAR DB 23h

.code

MOV AX,@DATA ;Sets AX equal to the starting address of data segment

MOV DS,AX ;Copy AX to DS

LEA DX,VAR ;Loads the address of VAR

RET ;stops the program