

تمرینهای فصل اول

- ۱- هر بایت از **هشت** بیت تشکیل شده است
- ۲- هر کیلو بایت معادل **۱۰۲۴** بایت است
- ۳- در هر *word* اعداد ۰ تا **۶۵۵۳۵** را میتوان جا داد
- ۴- در هر بایت اعداد **-۱۲۸** تا **+۱۲۷** را می توان قرار داد
- ۵- در کامپیوتر برای نمایش اعداد منفی از **مکمل ۲** استفاده می شود
- ۶- در کامپیوتر به جای عمل تفریق از **جمع با مکمل ۲** استفاده می گردد
- ۷- در هر *word* به تعداد **۲^{۱۶}** معادل **۶۵۵۳۶** وضعیت مختلف ۰،۱ وجود دارد
- ۸- عدد **۲۰**- را به سیستم دودویی تبدیل نمایید؟

$$(۲۰)_{۱۰} = ۱۰۱۰۰$$

$$۱۰۱۰۰ \text{ تبدیل به هشت رقم } ۰۰۰۱۰۱۰۰$$

$$۰۰۰۱۰۱۰۰ \text{ مکمل دو ان را به دست می اوریم}$$

$$۱۱۱۰۱۱۰۰$$

$$(-۲۰)_{۱۰} = ۱۱۱۰۱۱۰۰$$

۹- عمل ۱۸-۲۵ را با استفاده از روش مکمل دو انجام دهید؟

$$(۲۵)_{۱۰} = (۰۰۰۱۱۰۰۱)_{۲}$$

$$(۱۸)_{۱۰} = (۰۰۰۱۰۰۱۰)_{۲}$$

$$۱ \text{ محاسبه مکمل } : ۱۱۱۰۱۱۰۱$$

$$۲ \text{ محاسبه مکمل } : ۱۱۱۰۱۱۰۱+$$

۱

$$۱۱۱۰۱۱۱۰$$

$$۰۰۰۱۱۰۰۱+ \text{ جمع نمایش باینری } ۲۵ \text{ با مکمل } ۲ \text{ عدد } ۱۸$$

$$۱۱۱۰۱۱۱۰$$

$$\text{carry } 1 \quad \text{-----} \quad 0000111$$

۱۰- عدد ۱۱۰۱۱۰۱ را به سیستم دهدهی تبدیل نمائید؟

$$(1101101)_2 = (109)_{10}$$

$$1+4+8+32+64=109$$

۱۱- عدد ۱۰۱۱۰۱.۱۱۰۰۱ را به سیستم دهدهی تبدیل کنید؟

$$(101101.11001)_2$$

$$32+8+4+1+2/1+4/1+32/1=45.78125$$

۱۲- عدد $2FABC$ را به سیستم هشتایی تبدیل کنید؟

ابتدا هر رقم در مبنای ۱۶ را به عدد چهار رقمی در مبنای ۲ تبدیل میکنیم

$$(2FABC)_{16} = (00101111101010111100)_2$$

سپس سه رقم از سمت راست جدا میکنیم حاصل آنرا در مبنای ۸ مینویسیم

$$(000101111101010111100) = (0575274)_8$$

۱۳- عدد ۴۳۲۷۱ در سیستم مبنای هشت را به سیستم دودویی تبدیل نمائید؟

$$(43271)_8 = (100011010111001)_2$$

۱۴- مشخص نمائید MSB یک مقداری یک باشد آیا ان مقدار منفی است؟

در صورتی که عدد را به صورت علامت دار در نظر بگیریم مقدار منفی است . ولی اگر عدد را به صورت بدون علامت در نظر بگیریم یک بودن MSB نشان دهنده مقدار منفی نیست و یک ارزش عددی دارد.

- ۱۵- مشخص نمائید که آیا میتوان عمل ضرب و تفریق و تقسیم را به عمل جمع تبدیل نمود؟
عمل ضرب: جمع متوالی مضروب به تعداد مضروب فیه می توان عمل ضرب
عمل تفریق: جمع عدد اول با مکمل دو عدد دوم
عمل تقسیم: با تفریق متوالی مقسوم علیه از مقسوم
- ۱۶- (ب)
۱۷- (ب)
۱۸- (الف)

جواب تمرینهای فصل دوم

- ۱- دستور العمل *ADD* بروی فلگ های *CF, PF, AF, ZF, SF, OF* اثر میگذارد.
۲- دستور العمل *DEC* بروی فلگ های *CF, TF, IF, DF* اثر ندارد.
۳- دستور العمل *IDIV* مقادیر فلگ های *AF, CF, OF, PF, ZF, SF* تعریف نشده است.
۵- جواب در کتاب الگوریتم صفحه ۶۸ و برنامه صفحه ۱۸۹
- ۶

ADD AX, BX
ADD AX, CX
ADD AX, DX

-۷

MOV AX, 0
SUB AX, AX
XOR AX, AX

- ۸- دستور العمل *MUL* فقط بر روی فلگ های *CF, AF, ZF, SF* تعریف نشده هستند و بر روی فلگ های *TF, IF, DF* اثری نمی گذارد

-۹

MOV AX, X
ADD AX, X+2
ADD AX, X+4
ADD AX, X+6
ADD AX, X, 8

-۱۰

MOV AX, X
MOV DX, X+2
ADD AX, X+4
ADC DX, X+6
ADD AX, X+8
ADC DX, X+10
ADD AX, X+12
ADC DX, X+14
ADD AX, X+16
ADC DX, X+18

۱۱- بجای استفاده از دستور العمل *LEA* میتوان از دستور *MOV* استفاده نمود بعنوان مثال بجای

LEA BX, X

می توان دستور زیر بکار برد:

MOV BX, OFFSET X

-۱۲

ADD BX, 1

-۱۳

X DD 100000
Y DD 200000
Z DD 300000
W DD ?
MOV AX, X
MOV DX, X+2
ADD AX, Y
ADC DX, Y+2
SUB AX, Z
SBB DX, Z+2
ADD AX, 30
ADC DX, 0

-۱۴

MUL BL

MUL CL
MUL DL

در این برنامه فرض شده است که حاصلضرب ثبات ها از یک بایت بیشتر نمی شود

-۱۵

X DB
Y DB
Z DB
W DB
MOV AL, Y
MUL Z
ADD AL, X
SUB AL, W
ADD AL, 100
MOV W, AL

-16

X DB 2,3,4,5
Y DB ?
MOV AL, X
MUL X+1
MUL X+2
MUL X+3
MOV Y, AL

در این برنامه فرض شده است که نتیجه حاصلضرب حداکثر یک بایت می باشد در غیر اینصورت

جواب نهایی اشتباه می باشد

-۱۷

X DW 100
Y DW 200
Z DW 300
W DW ?
MOV AX, X
SUB AX, Y
ADD AX, Z
SUB AX, 200
MOV W, AX

-18

در صورتی که مقدار AL بدون علامت فرض شده باشد کافی است که مقدار AX را برابر صفر قرار دهیم تا محتوی AL به یک $Word$ تبدیل شود و نتیجه در AX قرار گیرد. در صورتی که مقدار AL علامت دار فرض شود در این حالت اگر MSB ثبات برابر صفر باشد در AH مقدار صفر قرار می دهیم و در حالتی که MSB ثبات AL برابر یک باشد محتوی AH را برابر $OFFH$ قرار می دهیم. ۱۹- در صورتی که محتوی ثبات AL علامت دار فرض شود مقدار $OFFH$ در ثبات AH قرار می دهیم در صورتی که محتوی ثبات AL بدون علامت فرض شود مقدار صفر در ثبات AH قرار داده می شود

-۲۰

```
X DW 10
Y DW 5
Z DW 20
W DW ?
MOV AX, X
MOV DX, 0
DIV Y
MUL Z
SUB AX, 10
MOV W, AX
```

جواب تمرینهای فصل پنجم

۱- دستورالعمل CMP مشابه دستورالعمل مشابه دستورالعمل SUB می باشد با این تفاوت که نتیجه عملیات تفریق در جایی ذخیره نمی شود و فقط بر روی فلکها تأثیر دارد.

۲- در دستورالعمل CMP هر دو عملونه در یک زمان نمی توانند متغیر باشد. بنابراین دستور $CMP X.Y$ مجاز نمی باشد.

۳- دستورالعمل JNA زمانی استفاده می شود که بخواهیم بررسی کنیم آیا در دستورالعمل CMP قبل از آن مقدار بدون علامت $OPr1$ از $OPr2$ بزرگتر نباشد. ($Not\ OPr1 > OPr2$)

۴- دستورالعمل **JLE** زمانی استفاده می شود که بخواهیم بررسی کنیم آیا در دستورالعمل **CMP** قبل

از آن مقدار علامت دار **OPr1** کوچکتر یا مساوی **opr2** باشند. $(opr1 \leq opr2)$

۵- در دستورالعمل **Loop** شرط تکرار حلقه، عدم صفر بودن ثبات **CX** می باشد $(\phi CX <>)$ در

حالی که در دستورالعمل **Loop z** شرط تکرار حلقه عدم صفر بودن ثبات **CC** و صفر بودن **ZF**

می باشد $(cx <> \phi \wedge = F = \phi)$

۶- در صورتی که در ابتدا محتوی **CX** برابر صفر باشد. حلقه $16=65536$ ۲ بار تکرار می شود. زیرا

در دستورالعمل **Loop** ابتدا عملیات کاهش مقدار **CX** انجام می شود و سپس صفر بودن **CX** بررسی

می شود. وقتی اولین بار مقدار **CX** کاهش می یابد مقدار آن برابر عدد ۱- می شود که این مقدار برابر

FFFFH یا همان عدد 65535 می باشد. بنابراین در کل 65536 بار حلقه تکرار می شود.

۷- این برنامه حاصل جمع اعداد یک تا **N** را محاسبه می کند و در متغیر **S** قرار می دهد.

از ثبات **CX** بجای متغیر **I** استفاده می کنیم. فرض می کنیم مقدار **N** برابر باشد.

S DW?

N DW 5

MOV Cx, N

MOV s, ϕ

START: ADD S, CX

Loop START

۸- این برنامه مجموعه آرایه ۱۰ عنصری **X** را محاسبه کرده و در متغیر **S** قرار می دهد. از بنات **SI**

بجای متغیر **I** استفاده می کنیم.

از بنات **AX** برای محاسبه مجموع مقادیر آرایه استفاده می نمایم.

X DW ۱ ϕ , ۲ ϕ , ۳ ϕ , ۴ ϕ , ۵ ϕ , ۶ ϕ , ۷ ϕ , ۸ ϕ , ۹ ϕ , ۱۰ ϕ

S DW?

MOV SI, 10

φ MOV Cx, 1

φ MOV Ax,

START: ADD Ax, X (SI)

Loop START

۹-فرض نمایید که X نام آرایه N عنصر از نوع بایت و متغیرهای N و TED از نوع Word باشند.

MOV CX, N

φ MOV TED,

MOV SI, φ

φ START: CMP X (SI),

JL NEXT

INC TED

NEXT: INC SI

Loop START

۱۱-

MOV CX, N

MOV SI,-1

MOV DI, N

START: IN SI

DEC DI

CMP SI, DI

JNB FINISH

MOV AL, X (SI)

XCHG x (DE), A

MOV x (SI), AL

Loop START

FINISH:


```
φ MOU SI,  
φ MOV DI,  
MOV CX, N  
φ START: CMP X (SI),  
J.NE NEXT  
MOV Ax, x (SI)  
MOV y (DI), Ax  
NEXT: ADD SI, 2  
Loop START
```

```
fact VW?  
MOV CX, N  
MOV Ax, 1  
START: MOV CX  
Loop START  
MOV fact, Ax
```

از اجرای برنامه مقدار فاکتوریل در متغیر fact از نوع Word ذخیره می شود.

ذخیره می شود. در این برنامه MIN در متغیر X بعد از اجرای برنامه مقدار کوچکترین عنصر آرایه

بدون علامت فرض شده است. X محتوای آرایه

```
MIN DD?  
MOV CX, N
```

```
LEA BX, x
MOV AX, (BX)
MOV DX, (BX +2)
START: CMP DX, (BX+2)
JA NEXT
CMP AX, (BX)
JAE NEXT
MOV AX, (BX)
MOV DX, (BX+2)
NEXT: ADD BX, 2
Loop START
MOV MIN, AX
MOV MIN +, DX
```

۱۶- در این برنامه اندیس بزرگترین عنصر آرایه در متغیر Max Indey ذخیره می‌شود عناصر آرایه

بدون علامت فرض می‌شود.

```
MAXINDEX DW?
MOV Cx, N
φ MOV MAXINDEX,
φ MOV SI,
MOV DX, X (SI)
START: CMP x (SI), DX
JAN NEXT
MOV DX, x (SI)
MOV MAXINDEX, SI
NEXT: ADD SI, 2
Loop START
```

```
TOTAL DW,?  
MOV CX, N  
φ MOV DX  
MOV AX, CX  
MOV CX  
ADD DX, AX  
Loop START  
MOV TOTAL,
```

مجموع مجذور اعداد یک تا N در متغیر TOTAL ذخیره می گردد.

-۱۸

```
MOV CX, N  
DEC CX  
START: MOV AX, N  
MOV DX, φ  
DIV CX  
CMP DX, φ  
DIV CX  
CMP DX, φ  
CMP CX, 2  
JE ONE  
Loop START  
ONE:  
NOTONE:
```

۲۰

```
NX DB?  
MOV CX, X  
MOV AL, N
```

START: MOV N
Loop START
MOV NX, AL

جواب تمرینهای فصل ۶

۱- دستورالعمل ROL DL,2

در دستورالعمل ROL مقدار بی واسط شیفت فقط عدد یک می تواند باشد. در صورتی که مقدار شیفت بیشتر از یک باشد باید آنرا در ثبات CL قرار دهیم.

SHL BX, CX: در دستورالعمل SHL، ثبات مقدار شیفت فقط CL می تواند باشد.

MOD X, BX: در دستورالعمل MOD هر دو مملو از همزمان نمی توانند متغیری در حافظه باشند.

۲- بعد از اجرای دستور CL و SAL BL محتوی BL چهار بیت به سمت چپ شیفت داده می شود و از سمت راست صفر دارد ثبات می شود. در نتیجه محتوی آن برابر $\phi B \phi H$ می شود.

در دستورالعمل CL و XOR BL:

BL: $\phi 1 2 0 \phi \phi$

CL: $0 0 0 0 1 0 0$

$XOR BL, CL \Rightarrow 0 1 0 0 = \phi B \xi H$

جواب تمرینهای فصل ۶

۴ -

OF	DF	IF	TF	SF	ZF	AF	PF	CF	دستور
ϕ	-	-	-	✓	✓	?	✓	ϕ	XOR
ϕ	-	-	-	✓	✓	?	✓	ϕ	AND
ϕ	-	-	-	✓	✓	?	✓	ϕ	TEST
-	-	-	-	-	-	-	-	-	MOV
?	-	-	-	✓	✓	?	✓	✓	SHL
*	-	-	-	-	-	-	*	*	RCR

✓ : تأثیر دارد

؟ : تعریف نشده (بعد از اجرای دستور مقدار تعریف نشده است)

- : تغییر نمی کند

ϕ : در تمام شرایط محتوای برابر صفر می شود.

۵-

```
STR DB This is a test
LEN DW 14? , string Length
MOV CX, LEV,
MOV LN.
STARTL: MOV AH, (BX)
CMP AH, bIH
JB NEXTCH
CMP AH, 7AH
JA NEXTCH
INC LN
NEXTCH: INC BX
LOOP STARTL
```

۶-

بیت‌های شماره فرد AX را به یک تبدیل کند

```
OR AX, 1010101010101010B
```

بیت‌های شماره زوج AX به صفر تبدیل می‌شود

```
AND AX, 1010101010101010B
```

۷-

```
TEST BL, 10100100B
```

```
JZ ZEROBITS
```

```
ZEROBITS:
```

۸-

NOT AX

TEST AX, 5555H

JZ ONEVAL

ONEVAL

۹ -

START: TEST AX 8000H

JNZ EXIT

SHIFT: SHL AX, 1

JMP START

EXIT:

۱۰ -

	دستور	AX	DX
Ax: ... ۰۱۰۰ ۱۰۱۱ ۱۱۰۰ Dx: ۱۱۱۱ ۰۰۱۰ ۱۰۱۱ ۱۱۰۰ Ax: ۱۱۱۱ ۰۱۱۰ F 6 0 0	MOV Ax , 4BCH MOV Ox , ϕF2BCH XOR AX,DX	4BCH ϕF2BCh ϕF6ϕϕH	
Dx: F6ϕϕ- Dx F5F8	SUB Dx , 8		ϕF5F8H
Dx: ۱۱۱۱ ۰۰۱۰ ۱۱۱۱ ۱۰۰۰ ۰۱۱۱ ϕ A ϕ 7	NOT DX		ϕAϕ7H
F6ϕϕ+ 16D ≡ 1ϕH ⇒ 1ϕ F61ϕ	ADD Ax,16		ϕF61ϕH
DX: ۱۰۱۰ ۰۱۱۱ Ax: ۱۱۱۱ ۰۱۱۰ ۰۰۰۱ ۸ ۰۰۱۰	AND DX,AX		200H

Dx:	ϕ	2	ϕ	ϕ		
-----	--------	---	--------	--------	--	--

۱۱ -

دستور	
STC	CF=1
STD	DF=1
STI	IF=1

```
N DB 5
MOV CL, N
INC CL
SHR AX, CL
JC ONE
JMP ZERO
ONE:
ZERO:
```

۱۲ -

```
 $\phi$  MOV AX, :۱ روش
 $\phi$  AND AX, :۲ روش
XOR AX, AX :۳ روش
SUB AX, AX :۴ روش
```


جواب تمرینهای فصل ۷

- ۱- تفاوت بین روال و ماکرو در این قسمت است که ماکرو در برنامه در زمان اجرا وجود خارجی ندارد و کدهای ماکرو در هر بار استفاده در زمان ترجمه درون کد برنامه جایگزین می‌شوند ولی روال در زمان اجرای برنامه وجود داشته و فراخوانی می‌کند و فقط یک بار در کد برنامه نوشته می‌شود و در برنامه اصلی هر بار فراخوانی می‌شود.
 - ۲- ماکروها می‌بایست در ابتدای برنامه تعریف شوند.
 - ۳- روال‌ها در بخش سگمنت برنامه *Code Segment* تعریف می‌شوند و نیاز به تعریف روال قبل از استفاده آن نمی‌باشد.
 - ۴- عملگر `;` باعث می‌شود که اسمبلر *Comment* ها را در موقع بسط ماکرو حذف نماید.
 - عملگر `&` دو *Text* یا سیمبل را بهم وصل نموده و در کنار یکدیگر قرار می‌دهد.
 - ۶- دایرکتور *Local* باعث می‌شود که اسمبلر برای عنصر در *dummy-list* ایجاد یک سیمبل منحصر بفرد نماید.
- شکل کلی:

LOCAL dummy-list

مثال :

```
WAIT MACRO Count
LOCAL NEXT
MOV CX, Count
NEXT: LOOP NEXT
POP CX
ENDM
```

حال اگر در برنامه سه بار از این ماکرو استفاده شود:

```
.
.
.
WAIT Count 1
.
.
.
WAIT Count 2
.
.
.
WAIT Count 3
.
.
```

و ماکروها را بسط دهیم بصورت زیر نمایان می گردد

PUSH CX

MOV CX, Count 1

NEXT 00: LOOP NEXT 00

POP CX

.

.

PUSH CX

MOV CX, Count 2

NEXT 01: LOOP NEXT 01

POP CX

.

.

PUSH CX

MOV CX, Count 2

NEXT 02: LOOP NEXT 02

POP CX

.

.

PUSH CX

MOV CX, Count 2

SAVEREG MACRO

PUSH AX

PUSH BX

PUSH CX

PUSH DX

PUSH SI

PUSH DI

ENDM

۸- در هنگام رخداد وقفه بایستی برنامه سرویس روتین آن وقفه اجرا شود . به آدرس این برنامه ، برداروقفه (Interrupt Vector) می گویند.

۹- جدول بردار وقفه شامل ۲۵۶ بردار وقفه ۴ بایتی می باشد که (یک کیلو بایت) در ابتدای حافظه قرار دارد.

-۱۰

STR DB 'Please Enter, Name: \$

LEA DX, STR

MOV AH, 9

INT 21 H

-۱۲

HRS DB ?

MINS DB ?

SECS DB ?

HSECS DB ?

Calculate Time 2 Key Press PROC NEAR

GET_KEY 1: MOV AH, 1

INT 21H

MOV AH, 2CH; Read Start Time

INT 21

-۱۲

MOV HRS, CH; and Save It

MOV MINS, CL

MOV SECS, DH

MOV HSECS, DL

GET_KEY2:

MOV AH, 1

INT 21H

MOV AH, 2CH

INT 21H; Read ent time

SUB DL, HSECS; calculate the difference

JNC SUB_SECS

ADD DL, 100

DEC DH

SUB_SECS: SUB DH, SECS

JNC SUB_MINS

ADD DH, 60

DEC CL

SUB-MINS: SUB CL, MINS

JNC SUB_HRS

ADD CL, 60

DEC CH

SUB_HRS: SUB CH, HRS

```
Myproc proc NEAR
MOV AH, 00
MOV AL, 03
INT LOH; clear screen
MOV AH, 2
MOV BH,2
MOV DH,10 ;row
MOV DL,40 ; column
INT LOH
MOV AH, 02H
MOV DL, 21H
```

```
SUMN MACRO N, S
PUSH AX
PUSH CX
LOCAL START
MOV AX, 0
MOV CX,0
MOVE CL, N
START: ADD AX, CX
LOOP START
MOV S, AX
POP CX
POP AX
ENDM
```

```
BUBBLE PROC NEAR
    LEA     DI, X
    MOV     BP, DI
    MOV     DX, N
    MOV     CX, DX
    MOV     BX, 1
    DEC     DX
    JZ      SORTED
    MOV     CX, DX
    MOV     DI, BP
    ADD     DI, 1
    MOV     AL, [DI]
    CMP     [DI+1], AL
    MOV     [DI], AL
    SUB     BX, BX
CONT:
    LOOP    NEXT
    CMP     BX, 0
    JE      INIT
    ENDP
```

```
MINIMUM MACRO N, X, MIN
    PUSH   AX
    PUSH   CX
    PUSH   DI
    LOCAL START, NEXT
    MOV    DI, 0
    MOV    CX, N
    MOV    AX, X      ; MINIMUM
START:
    CMP   [DI], AX
    JAE  NEXT
    MOV   AX, X [DI]
NEXT:
    ADD  DI, 2
    LOOP SRART
    MOV  MIN, AX
    POP  DI
    POP  CX
    POP  AX
```

```
ONEM MACRO N, S
    MOV     AX, N
    MOV     DX, 0
    MOV     BX, 2
    DIV     BX
```

```
MOV CX, AX
SRART:
MOV DX, 0
MOV AX, N
DIV CX
CMP DX, 0
JE ONE-NUM
LOOP START
MOV ONE, 0 ; NOT PRIME NUMBER
JMP NEXT
ONE-NUM:
MOV ONE ; PRIME NUMBER
EXIT:
ENDM
```

-19

```
KMM MACRO N, M, K
MOV PI, 0
MOV SI, N
MOV BX, M
START:
INC DI
MOV AX, DI
MUL BX
DIV SI
CMP DX, 0
JE FOUND
JMP START
FOUND:
MOV AX, DI
MUL BX
MOV K, AX
ENDM
```

-20

```
SquareRoot MACRO N, R
MOV CX, N
MOV BX, 0
START:
INC BX
MOV AX, BX
MUL BX
CMP AX, N
JE FOUND
LOOP START
FOUND:
MOV R, BX
ENDM
```

جواب تمرینهای فصل 8

۱- پیشوند REP در دستورهای رشته ای زیر کاربرد دارد

MOVSB, MOVSW, STOSB, STOSW, LODSB, LODSW

۲- پیشوند REPZ در دستورهای رشته ای زیر کاربرد دارد

CMPSB, CMPSW, SCASB, SCASW

۳- دستور SCAS بروی فلاگ های OF,SF,ZF,AF,PF,CF تاثیر دارد

دستور STOS بروی هیچ فلاگی تاثیر ندارد.

-۴

```
PATTERN DB '-----'  
STRG DB 100 DUP(?)  
CLD  
MOV CX, 20  
LEA SI, PATTERN  
LEA DI, STRG  
REP MOVSB
```

-5

```
STRLEN EQU 100  
STRG DB 100 DUP(?)  
CLD  
MOV CX, STRLEN  
MOV AL, 20H; BLANK CHARACTER  
LEA SI, STRG  
START:  
REPNE SCASB  
JNZ EXIT  
DEC DI  
MOV BYTE PTR [DI], '*'  
JCXZ EXIT  
INC DI  
JMP START
```

-6

```
STRLEN EQU 100  
STRG DB 100 DUP(?)  
CLD  
MOV AL, '*'  
LEA DI, STRG  
REP STOSB
```



```
STRLEN EQU 100
STRG DB 100 DUP(?)
MOV CX, STRLEN
MOV AL, '*'
CLD
LEA DI, TRG
REPNE SCASB
JNZ NOT-FOUND
DEC DI
MOV BYTE PTR [DI], '&'
.
.
.
NOT-FOUND:
```

```
STRG DB 100 DUP(?)
STRG DB 20 DUP(?)
CLD
MOV CX, 20
LEA SI, STRG1
LEA DI, STRG2
ADD SI, 40
REP MOVSB
```

```
STRG1 DB 50 DUP(?)
STRG2 DB 100 DUP(?)
LEA SI, STRG1
LEA DI, STRG2
ADD DI, 50
MOV CX, 50
CLD
REPE CMPSB
JZ EQUAL
JMP NOT-EQUAL
.
.
.
EQUAL:
```

NOT-EQUAL:

.
. .
.

-۱۰

```
STRG DB 50 DUP(?)
MOV AL, '*'
CLD
START:
MOV CX, 50
LEA DI, STRG
REPNE SCASB
JNZ EXIT
MOV SI, DI
DEC DI
REP MOVSF
MOV BYTE PTR [DI], 20H; BLANK
JMP START
EXIT:
```

دانشنامه تخصصی کامپیوتر

dt-computer.blogfa.com