



مؤسسه آموزش عالی بهار

غیرانتفاعی - غیردولتی



زبان ماشین و اسمبلی

تهیه کننده: فرزاد زندی

zandi_farzad@yahoo.com

zandi8farzad@gmail.com

ثباتها:

۱- ثباتهای عمومی:

(هر کدام ۱۶ بیت)

Ax: ثبات انباشتگر, برای محاسبات. ۱۶ بیت شامل ۸ بیت al و ۸ بیت ah.

Bx: ثبات پایه, برای توسعه آدرس. ۱۶ بیت شامل ۸ بیت bl و ۸ بیت bh.

Cx: ثبات شمارنده, برای شمارنده. ۱۶ بیت شامل ۸ بیت cl و ۸ بیت ch.

Dx: ثبات داده, برای محاسبات. ۱۶ بیت شامل ۸ بیت dl و ۸ بیت dh.

ثباتها:

۲- ثباتهای سگمنت:

(هر کدام ۱۶ بیت)

DS: ثبات سگمنت داده. آدرس سگمنت داده را نگهداری می کند.

CS: ثبات کد. آدرس سگمنت کد را نگهداری می کند.

SS: ثبات پشته. آدرس سگمنت پشته را نگهداری می کند.

ES: ثبات اضافی. آدرس سگمنت اضافی را نگهداری می کند.

ثباتها:

۳- ثباتهای اندیس:

(هر کدام ۱۶ بیت)

Si: برای شمارش. آدرس رشته مبدا را نگهداری می کند.

Di: برای شمارش. آدرس رشته مقصد را نگهداری می کند.

Bp: اشاره گر توسعه گر Sp. برای کمک به Sp به کار می رود.

Sp: اشاره گر بالای پشته. آدرس بالاترین عنصر پشته را نگهداری می کند.

ثباتها:

۴- ثبات پرچم:

(۱۶ بیت)

R	R	R	R	OF	DF	IF	TF	SF	ZF	R	AF	R	PF	R	CF
---	---	---	---	----	----	----	----	----	----	---	----	---	----	---	----

CF: وقتی یک می شود که در یک آخر رقم نقلی ایجاد شود.

PF: بسته به نوع توازن (فرد یا زوج) مقدار یک یا صفر می گیرد.

AF: وقتی یک می شود که از بیت چهارم رقم نقلی ایجاد شود.

ZF: وقتی یک می شود که نتیجه محاسبات صفر شود.

SF: مقدار آخرین بیت در اعداد علامت دار را می گیرد.

OF: وقتی یک می شود که سرریز رخ دهد.

DF: برای کنترل جهت عملیات دستورات رشته بکار میرود.

TF: برای پیدا کردن اشتباه بکار می رود.

ساختار کلی یک برنامه اسمبلی:

segment نام

دستورات تعریف داده

Ends نام

segment نام

دستورات تعریف پشته

Ends نام

segment نام

کدهای برنامه

Ends نام

data segment

دستورات تعریف داده

data Ends

stack segment

دستورات تعریف پشته

stack Ends

code segment

کدهای برنامه

code Ends

مثال

ساختمان کامپیوتر:

نوع پردازنده	تعداد بیت
۸۰۸۸	۸ بیت
۸۰۱۸۸	۸ بیت
۸۰۸۶	۱۶ بیت
۸۰۱۸۶	۱۶ بیت
۸۰۲۸۶	۱۶ بیت
۸۰۳۸۶	۳۲ بیت
۸۰۴۸۶	۳۲ بیت
۸۰۵۸۶	۳۲ بیت
Pentium	۳۲ بیت
پردازنده های نسل جدید	۶۴ بیت

$$1\text{mb} = 2^{20} =$$

0000,0000,0000,0000,0000

تا

1111,1111,1111,1111,1111

F F F F F

نشانه شروع سگمنت

آدرس

00000h

00006h

...

6F4F0h

2F0E0h

3F670h

FFFFEh

FFFFFh

ساختمان یک
مگابایت
Ram از

دستورات تعریف داده:

▪ دستور db:

(تعریف متغیر از نوع بایت) یک متغیر از نوع بایت با مقادیر ۰ تا ۲۵۵ می توان تعریف کرد.

شکل کلی:

مثال:

مقدار اولیه db نام متغیر

- D1 db 100
- D2 db 200
- D3 db 300
- ~~D4 db 255~~
- D5 db ?
- D6 db 'Tehran\$'
- D7 db 'a\$'
- D8 db 10 dup(?)

دستورات تعریف داده:

▪ دستور dw:

تعریف متغیر از نوع کلمه (دو بایت). یک متغیر از نوع بایت با مقادیر ۰ تا 65535 می توان تعریف کرد.

شکل کلی:

مثال:

مقدار اولیه dw نام متغیر

- D1 dw 55000 D2 dw 200
- D3 dw 3000
- ~~▪ D4 dw 70000~~
- D5 dw ?
- D6 dw 'Tehran\$'
- D7 dw 'a\$'
- D8 dw 10 dup(?)

دستورات تعریف داده:

▪ دستور dd:

متغیر دو کلمه (چهار بایت) تعریف می کند.

D1 dd 32445678H

▪ دستور dq:

متغیر چهار کلمه (هشت بایت) تعریف می کند.

D2 dq 1234567891254568H

▪ دستور dt:

متغیر ۱۰ بایت تعریف می کند.

D3 dt 45369467327889789010H

دستور انتساب:

• شکل کلی:

Mov عملوند مبدا, عملوند مقصد

```
Mov ah, 10  
Mov al, 20  
Mov di, 10  
Mov si, 3  
Mov bx, 500  
Mov x, 25  
Mov cl, y  
Mov dl, 'e'
```

```
Mov ah, bx  
Mov bl, 300
```

مثال:

نکته ۱:

عملوند مقصد و مبدا نمی توانند همزمان متغیر باشند.

نکته ۲:

اندازه عملوند مقصد باید از عملوند مبدا بزرگتر باشد.

دستورات افزایشی و کاهششی:

- دستور افزایش:

یک واحد اضافه می کند.

```
D1 db 15  
Inc d1
```

- دستور کاهش:

یک واحد کم می کند.

```
D1 db 15  
dec d1
```

دستور تعویض محتوای عملوند:

▪ شکل کلی:

عملوند دوم, عملوند اول **xchg**

▪ مثال:

- Xchg al, bl
- Xchg cx, m
- Xchg n, dh
- ~~Xchg dl, 50~~
- ~~Xchg m, 80~~
- ~~Xchg al, bx~~

دستورات محاسباتی:

- دستور جمع:
شکل کلی:

عملوند ۲, عملوند ۱ Add

مثال:

- Add m1, 5 -> $m1 = m1 + 5$
- Add m2, 30 -> $m2 = m2 + 30$
- Add m3, m4 -> $m3 = m3 + m4$
- Add ax, bh -> $ax = ax + bh$
- Add cx, 560 -> $cx = cx + 560$
- ~~Add 40, 50~~

دستورات محاسباتی:

- دستور جمع با رقم نقلی:
عملوند ۱ را با عملوند ۲ جمع کرده و حاصل را با بیت پرچم نقلی جمع می کند.
شکل کلی:

عملوند ۲, عملوند ۱ **Adc**

مثال:

- | | | |
|----------------------|----|----------------------|
| • Adc m1, 5 | -> | $m1 = m1 + 5 + cf$ |
| • Adc m2, 30 | -> | $m2 = m2 + 30 + cf$ |
| • Adc m3, m4 | -> | $m3 = m3 + m4 + cf$ |
| • Adc ax, bh | -> | $ax = ax + bh + cf$ |
| • Adc cx, 560 | -> | $cx = cx + 560 + cf$ |

دستورات محاسباتی:

- دستور تفریق:
شکل کلی:

عملوند ۲, عملوند ۱ Sub

مثال:

- Sub m1, 5 -> $m1 = m1 - 5$
- Sub m2, 30 -> $m2 = m2 - 30$
- Sub m3, m4 -> $m3 = m3 - m4$
- Sub ax, bh -> $ax = ax - bh$
- Sub cx, 560 -> $cx = cx - 560$
- ~~Sub 40, 50~~

دستورات محاسباتی:

- دستور تفریق با رقم قرضی:
عملوند ۱ را از عملوند ۲ کم کرده بعلاوه بیت پرچم نقلی کم می کند.
شکل کلی:

عملوند ۲, عملوند ۱ **Adc**

مثال:

- | | | |
|----------------------|----|------------------------|
| • Adc m1, 5 | -> | $m1 = m1 - (5 + cf)$ |
| • Adc m2, 30 | -> | $m2 = m2 - (30 + cf)$ |
| • Adc m3, m4 | -> | $m3 = m3 - (m4 + cf)$ |
| • Adc ax, bh | -> | $ax = ax - (bh + cf)$ |
| • Adc cx, 560 | -> | $cx = cx - (560 + cf)$ |

دستورات محاسباتی:

- دستور ضرب:

۱- ضرب بایت در بایت:

دو عملوند یک بایتی در هم ضرب می‌شود, پاسخ یک عدد(دو بایتی) ۱۶ بیتی است. یکی از عملوندها باید در **al** قرار گیرد و دومی جلوی دستور **mul** و حاصل در **ax** قرار می‌گیرد.

```
Mov al, 10
```

```
Mov bl, 200
```

```
Mul bl          ->      ax = al*bl
```

~~Mul 200~~

۲- ضرب کلمه در کلمه:

دو عملوند دو بایتی در هم ضرب می‌شود, پاسخ یک عدد(چهار بایتی) ۳۲ بیتی است. یکی از عملوندها باید در **ax** قرار گیرد و دومی جلوی دستور **mul** و حاصل در **dx:ax** قرار می‌گیرد.

```
Mov ax, 1000
```

```
Mov bl, 200
```

```
Mul bl          ->      dx:ax = ax*bl
```

~~Mul 200~~

نکته:

دستور **mul** برای ضرب اعداد بی علامت و دستور **imul** برای ضرب اعداد علامت دار است.

دستورات محاسباتی:

• دستور تقسیم:

۱- تقسیم کلمه بر بایت:

مقسوم ۱۶ بیتی در **ax** و مقسوم علیه یک بیتی جلوی دستور **div** نوشته می شود. خارج قسمت در **al** و باقیمانده در **ah** قرار می گیرد.

۲- تقسیم کلمه مضاعف بر بایت:

مقسوم ۳۲ بیتی در **dx:ax** و مقسوم علیه ۱۶ بیتی جلوی دستور **div** نوشته می شود. خارج قسمت در **ax** و باقیمانده در **dx** قرار می گیرد.

نکته:

دستور **div** برای ضرب اعداد بی علامت و دستور **idiv** برای ضرب اعداد علامت دار است.

مثال:

• اعداد ۵۰ و ۶۰ را در دو متغیر تعریف و سپس آنها را ضرب کنید.

M1 db 50	->	M1=50
M2 db 60	->	M2=60
M3 dw ?	->	M3 = ?
Mov al, M1	->	al = M1
Mul M2	->	ax = al*M2
Mov M3, ax	->	M3 = ax

مثال:

- نتیجه تقسیم اعداد ۳۵۸ بر ۲۵ را محاسبه و خارج قسمت و باقیمانده در را دو متغیر قرار دهید.

M1 dw 358	->	M1=358
M2 db 25	->	M2=25
M3 dw ?	->	M3=?
M4 dw ?	->	M4=?
Mov ax, M1	->	ax = M1
Div M2	->	al = ax/M2, ah=ax%M2
Mov M3, al	->	M3 = al
Mov M4, ah	->	M4 = ah

مثال:

- نتیجه تقسیم اعداد ۳۰۰۰ بر ۱۵۰ را محاسبه و خارج قسمت و باقیمانده در را دو متغیر قرار دهید.

M1 dw 3000	->	M1 = 3000
M2 db 150	->	M2 = 150
M3 dw ?	->	M3 = ?
M4 dw ?	->	M4 = ?
Mov dx, 0	->	dx = 0
Mov ax, M1	->	ax = M1
Div M2	->	ax = ax/M2, dx=ax%M2
Mov M3, ax	->	M3 = ax
Mov M4, dx	->	M4 = dx

وقفه (Interrupt):

وقفه سیگنالی است که از طرف یک دستگاه جایی به برنامه در حال اجرا فرستاده می‌شود و درخواست انجام کاری را می‌کند.

هر گاه وقفه‌ای رخ دهد محتویات ثباتهای Cs و Ip در پشته نگهداری تا پس از اجرای وقفه بازیابی گردد.

انواع وقفه:

۱- وقفه برنامه نویس

۲- وقفه سیستم

الف) وقفه سخت‌افزاری

ب) وقفه نرم‌افزاری

وقفه خروج از برنامه:

از تابع شماره 4ch و وقفه شماره 21h استفاده می‌شود.

Mov ah, 4ch
Int 21h } Exit();

مشابه

Mov al, 0h
Mov ah, 4ch
Int 21h } Exit();

نکته:

۱- شماره تابع باید در ثبات ah قرار گیرد.

۲- برای اجرای وقفه از دستور Int استفاده می‌شود.

وقفه دریافت کاراکتر با نمایش:

از تابع شماره 1h و وقفه شماره 21h استفاده می‌شود.

Mov ah, 1h
Int 21h

} Getche();

Ch db ?
Mov ah, 1h
Int 21h
Mov ch, al

} Ch = getche();

مثال

نکته:

بعد از گرفتن
کاراکتر از صفحه
کلید, به صورت
خودکار در ثبات al
قرار می‌گیرد.

وقفه دریافت کاراکتر بدون نمایش:

از تابع شماره 8h و وقفه شماره 21h استفاده می‌شود.

Mov ah, 8h	}	Getch ();
Int 21h		
Ch db ?	}	Ch = getch ();
Mov ah, 8h		
Int 21h		
Mov ch, al		

مثال

نکته:

بعد از گرفتن
کاراکتر از صفحه
کلید، به صورت
خودکار در ثبات al
قرار می‌گیرد.

وقفه چاپ کاراکتر:

از تابع شماره 2h و وقفه شماره 21h استفاده می‌شود.

مثال

```
Ch db 'A'  
Mov dl, Ch  
Mov ah, 2h  
Int 21h
```

putch('A');

نکته:

کد اسکی کاراکتری
که قرار است چاپ
شود باید در ثبات
dl قرار گیرد.

وقفه انتقال مکان نما:

از تابع شماره 2h و وقفه شماره 10h استفاده می شود.
مثال:

مکان نما را به نقطه (10,20) منتقل می کند.

```
Mov dh, 20  
Mov dl, 10  
Mov ah, 2h  
Int 10h
```

نکته:

شماره سطر باید در
ثبات dh و شماره
ستون باید در ثبات
dl قرار گیرد.

وقفه چاپ رشته:

از تابع شماره 9h و وقفه شماره 21h استفاده می‌شود.

مثال:

چاپ رشته 'I like assembly'

```
St1 db "I like assembly$"
```

```
Lea dx, St1
```

```
Mov ah, 9h
```

```
Int 21h
```

نکته:

- ۱- هر هنگام تعریف و مقداردهی به رشته, انتهای آن علامت \$ می‌گذاریم.
- ۲- آدرس رشته مورد نظر باید در ثبات dx قرار گیرد.

وقفه دریافت رشته:

از تابع شماره 0ah و وقفه شماره 21h استفاده می‌شود.
مثال:

```
String1 label byte  
Max1 db 20  
Len1 db 20  
Str1 db 20 dup(?)
```

```
Lea dx, String1  
Mov ah, 09h  
Int 21h
```

نکته:

آدرس ساختمان
داده تعریف شده
برای رشته باید در
ثبات dx قرار
گیرد.

وقفه چاپ پنجره:

از تابع شماره 6h و وقفه شماره 10h استفاده می‌شود.
مثال:

```
Mov ch, 10  
Mov cl, 12  
Mov dh, 15  
Mov dl, 25  
Mov bh, 14h  
Mov ah, 6h  
Int 10h
```

نکته:

در ثبات ah باید رنگ زمینه و رنگ متن قرار گیرد.
عدد سمت چپ رنگ زمینه و عدد سمت راست رنگ متن است.

نکته:

شماره ستون گوشه بالا سمت چپ در Cl و شماره سطر گوشه بالا سمت چپ در ch و شماره ستون گوشه پایین سمت راست در dl و شماره سطر گوشه پایین سمت راست در dh باید قرار گیرد.

دستورات شرطی:

انواع پرش:

۱- پرش شرطی (با انجام مقایسه)

۲- پرش غیر شرطی (بدون انجام مقایسه)

مثال پرش غیر شرطی:

مثال اسمبلی

L1:

دستورات

jmp L1;

مثال C++

L1:

دستورات

Goto L1;

دستور مقایسه:

Cmp عملوند ۲, عملوند ۱

اعداد بدون علامت	اعداد علامت دار	معادل
Jb	Jl	عملوند ۱ > عملوند ۲
Jbe	Jle	عملوند ۱ >= عملوند ۲
Ja	Jg	عملوند ۱ < عملوند ۲
Jae	Jge	عملوند ۱ <= عملوند ۲
Je	Je	عملوند ۱ == عملوند ۲
jne	jne	عملوند ۱ != عملوند ۲

مثال:

مثال C++

```
If (x>10)  
    دستورات ۱  
Else  
    دستورات ۲
```

مثال
اسمبلی

```
If1:  
    cmp x,10  
    Ja p1  
    دستورات ۲  
    jmp Endif  
P1:  
    دستورات ۱  
Endif:
```

مثال:

مثال C++

If ($x \geq 10$)

دستورات ۱

Elseif ($y < 10$)

دستورات ۲

Else

دستورات ۳

مثال اسمبلی

If1:

```
cmp x,10  
Jge p1  
cmp y, 10  
jl p2  
jmp p3
```

P1:

دستورات ۱
jmp endif

P2:

دستورات ۲
Jmp endif

P3:

دستورات ۳

Endif:

If (x>10 || y<50)

دستورات

If1:

cmp x,10

jg P1

cmp y, 50

jl P1

jmp Endif

P1:

دستورات

Endif:

If (x<10 && x>1)

دستورات

If1:

cmp x,10

jge Endif

cmp x, 1

jle Endif

دستورات

Endif:

مثال

If (x>10 || x>1)

دستورات ۱

Else

دستورات ۲

If1:

cmp x,10

jle P1

cmp x, 1

jg P1

دستورات ۲

jmp Endif

P1:

دستورات ۱

Endif:

مثال: برنامه‌ای بنویسید که دو کاراکتر از صفحه کلید دریافت کرده و با هم مقایسه کند و پیغام مناسب چاپ کند.

```
Ch1 db ?  
Ch2 db ?  
G db '>'  
L db '<'  
E db '='  
St1 db "Enter a character: $"
```

```
Lea dx, st1  
Mov ah, 9h  
Int 21h
```

```
Mov ah, 1h  
Int 21h  
Mov ch1, al
```

```
Mov ah, 1h  
Int 21h  
Mov ch2, al
```

If1:

```
mov al, ch1  
Cmp al, ch2  
jl P1  
jg P2  
mov ah, 2h  
mov dl, E  
Int 21h  
jmp Endif
```

P1:

```
mov ah, 2h  
mov dl, L  
Int 21h  
jmp Endif
```

P2:

```
mov ah, 2h  
mov dl, G  
Int 21h
```

Endif:

مثال: برنامه‌ای بنویسید که بر اساس یکی از حروف "۱"، "۲" تا "۹" معادل حروفی آنرا در خروجی چاپ کند.

```
Ch1 db ?
St1 db "One $"
St2 db "Two $"
St3 db "Three $"
St4 db "Four $"
St5 db "Five $"
St6 db "Six $"
St7 db "Seven $"
St8 db "Eight $"
St9 db "Nine $"
St10 db "Enter a number: $"
```

```
Lea dx, st1
Mov ah, 9h
Int 21h
```

```
Mov ah, 1h
Int 21h
Mov ch1, al
```

```
Cmp ch1, "1"
```

```
Je P1
```

```
Jmp End
```

```
.
```

```
.
```

```
.
```

```
Cmp ch1, "9"
```

```
Je P2
```

```
Jmp End
```

```
P1:
```

```
mov dx, st1
```

```
int 21h
```

```
jmp End
```

```
.
```

```
.
```

```
.
```

```
P9:
```

```
mov dx, st9
```

```
int 21h
```

```
End:
```

حلقه‌های تکرار:

```
i db 0
```

```
For1:
```

```
    cmp i, 10  
    Je Endfor1
```

دستورات

```
    inc i
```

```
    jmp For1
```

```
Endfor1:
```

حلقه for

```
i db 0
```

```
K db 1
```

```
While1:
```

```
    cmp i, 10
```

```
    Je Endwhile1
```

```
    Mov al, 2
```

```
    Mul k
```

```
    Mov k, al
```

```
    inc i
```

```
    jmp Endwhile1
```

```
Endwhile1:
```

$K = k * 2$

حلقه
while

مثال: برنامه‌ای بنویسید که یک رشته از ورودی دریافت کرده و تعداد حروف آنرا شمارش نماید.

```
String1 label byte  
Max1 db 20  
Len1 db ?  
Str1 db 20 dup(?)  
i db 0
```

```
Lea dx, String1  
Mov ah, 0ah  
Int 21h
```

```
Lea si, Str1  
While1:  
    mov bl, [si]  
    Cmp bl, 13  
    je Endwhile  
    Inc i  
    inc si  
    jmp while1  
Endwhile1:
```

مثال: برنامه‌ای بنویسید که رشته‌ای را از ورودی دریافت و حروف آنرا به بزرگ تبدیل کند.

```
String1 label byte  
Max1 db 20  
Len1 db ?  
Str1 db 20 dup(?)  
Str2 db 20 dup(?)
```

```
Lea dx, String1  
Mov ah, 0ah  
Int 21h
```

```
Lea si, Str1  
Lea si, Str2
```

```
While1:  
    mov bl, [si]  
    Cmp bl, 13  
    je Endwhile1  
    sub bl, 32  
    Mov [di], bl
```

```
Inc si  
inc di  
Jmp Endwhile1  
Endwhile1:
```

```
Lea dx, str2  
Mov ah, 9h  
Int 21h
```

مثال: برنامه‌ای بنویسید که یک عدد در یک متغیر تعریف کرده و تشخیص دهد زوج است یا فرد و پیغام مناسب چاپ کند.

```
Msg1 db "Fard$"  
Msg2 db "Zoj$"  
Num dw 120
```

```
Mov ax, x  
Mov cl, 2  
Div cl
```

```
If1:  
    cmp ah, 0  
    Je P1  
    lea dx, msg2  
    mov ah, 9h  
    Int 21h  
    jmp Endif1:
```

```
P1:  
    lea dx, msg1  
    mov ah, 9h  
    int 21h
```

```
Endif1:
```

مثال: برنامه‌ای بنویسید که کاراکترهای ۰ تا ۲۵۵ را در سطرها و ستونهای جداگانه چاپ نماید.

```
Mov cl, 0
```

```
For1:
```

```
    cmp cl, 256  
    je Endfor1
```

```
    mov dl, cl  
    mov ah, 2h  
    int 21h
```

```
    mov dl, ' '  
    mov ah, 2h  
    int 21h
```

```
    mov al, cl  
    mov ah, 0  
    mov bl, 10  
    div bl  
    inc cl
```

```
Cmp ah, 0  
jne for1
```

```
    mov ah, 2h  
    mov dl, 10  
    int 21h
```

```
    mov ah, 2h  
    mov dl, 13  
    int 21h
```

```
Endfor1:
```

مثال: برنامه‌ای بنویسید که خروجی زیر را تولید کند.

Mov si, 0

For1:

cmp si, 3
je Endfor1
mov di, 0

For2:

cmp di, 5
je Endfor2
mov dl, '*'
mov ah, 2h
int 21h
inc di
jmp For2

Endfor2:

mov ah, 2h
mov dl, 10
Int 21h

mov ah, 2h

mov dl, 13

int 21h

inc si

jmp For1

Endfor1:

مثال: برنامه‌ای بنویسید که خروجی زیر را تولید کند.

```
Mov si, 1
```

```
For1:
```

```
    cmp si, 6  
    je Endfor1  
    mov di, 0
```

```
For2:
```

```
    cmp di, si  
    je Endfor2  
    mov dl, '*'  
    mov ah, 2h  
    int 21h  
    inc di  
    jmp For2
```

```
Endfor2:
```

```
    mov ah, 2h  
    mov dl, 10  
    Int 21h
```

```
mov ah, 2h
```

```
mov dl, 13
```

```
int 21h
```

```
inc si
```

```
jmp For1
```

```
Endfor1:
```

```
*
```

```
**
```

```
***
```

```
****
```

```
*****
```

مثال: برنامه‌ای بنویسید که کاراکتر * را در سطر اول از ستون اول تا ستون آخر حرکت دهد.

```
i db 0
Mov si, 0
```

For1:

```
    cmp si, 79
    je Endfor1
    mov dl, '*'
    mov ah, 2h
    int 21h
    mov dl, i
    mov dh, 0
    mov ah, 2h
    int 10h
    mov dl, ' '
    mov ah, 2h
    int 21h
    inc si
    inc i
    jmp for1
```

Endfor1:

مثال: برنامه‌ای بنویسید که رشته‌ای را از ورودی دریافت کرده و آنرا به صورت معکوس چاپ نماید.

```
String1 label byte  
Max1 db 20  
Len1 db ?  
Str1 db 20 dup(?)  
i db 0
```

```
Mov ah, 0ah  
Lea dx, String1  
Int 21h
```

```
Lea si, str1  
Mov cl, len1  
Mov ch, 0  
Add si, cx
```

```
For1:  
    dec si  
    cmp i, cl  
    je Endfor1
```

```
    mov dl, [si]  
    mov ah, 2h  
    int 21h  
    inc i  
    jmp For1  
Endfor1:
```

مثال: برنامه‌ای بنویسید که رشته‌ای را از ورودی دریافت کرده و تشخیص دهد همه حروف آن بزرگ است یا کوچک و پیغام مناسب چاپ کند.

```
String1 label byte
Max1 db 20
Len1 db ?
Str1 db 20 dup(?)
i db ?
j db ?
k db ?
Msg1 db "Large...!$"
Msg2 db "Small...!$"
Msg3 db "Complex...!$"
```

```
Mov ah, 0ah
Lea dx, String1
Int 21h
```

```
Lea si, str1
```

```
For1:
    mov cl, [si]
    cmp cl, 13
```

```
je Endfor1
cmp cl, 'a'
jl Endif1
cmp cl, 'z'
jg Endif1
mov i, 0
mov j, 0
mov k, 0
Endif1:
cmp cl, 'A'
jl Endif2
mov i, 1
mov j, 0
mov k, 0
Endif2:
mov i, 0
mov j, 0
mov k, 1
cmp i, 1
je L1
```

```
cmp j, 1
```

```
je S1
```

```
cmp k, 1
```

```
je C1
```

```
L1:
```

```
    lea dx, msg1
```

```
    mov ah, 9h
```

```
    int 21h
```

```
    jmp Endfor1
```

```
S1:
```

```
    lea dx, msg2
```

```
    mov ah, 9h
```

```
    int 21h
```

```
    jmp Endfor1
```

```
C1:
```

```
    lea dx, msg3
```

```
    mov ah, 9h
```

```
    int 21h
```

```
inc si
```

```
jmp for1
```

```
Endfor1:
```