

به نام خدا

گزارش کار آزمایشگاه ریزپردازنده

آزمایش ۹

مدرس : مهندس بی طالبی

تارا برقیان

مهرشاد سعادت‌نی نیا

نیم سال اول ۱۴۰۰-۰۱



## فهرست

۳	..... : مقدمه
۳	..... : پاسخ سوال ۱
۴	..... : پاسخ سوال ۲

## مقدمه :

در این تمرین با برنامه نویسی زبان اسمبلی ۸۰۸۶ آشنا شدیم.

## پاسخ سوال ۱:

در این سوال از تابع `printnum` سوال دوم استفاده کردیم که در ادامه توضیح خواهیم داد. با این تفاوت که به جای عدد ۳۲ بیت ۸ بیت کافی بود.

ابتدا کرسر تنظیم میشود، سپس مقدار رجیستر `al` را برای نگه داری شمارنده صفر میکنیم. دیتاهای ۱ و ۲ هم در رجیسترهای `bx` و `di` نگه میداریم (چاره ی دیگری هم نداریم). چون `CX` برای شمارش است، `ax` که استفاده شد و `dx` برای چک کردن باید استفاده شود. دیتاهای داده شده هر دو شامل ۲۰ بایت هستند. چون هر `word` ۲ بایت دارد و هر `DD` نیز ۴ بایت. برای همین `cx=20` خواهد شد (۱ تا ۲۰ حلقه میزنیم).

```
.CODE
MAIN PROC FAR
    mov ax, @DATA
    mov ds, ax

    mov ah, 0
    mov al, 03h
    int 10h

    mov si, 02h
    mov al, 0 ;result
    mov di, offset DATA1
    mov bx, offset DATA2
    mov cx, 20 ; number of bytes to compare
```

در ادامه با هر گردش حلقه ابتدا بایت های مورد نظر در `dh` و `dl` قرار میگیرند و مقایسه میشوند اگر برابر بودند `al` یکی اضافه میشود. به بایت بعدی هر دو دیتا میرویم و `CX` یکی کم میشود. مجدد حلقه تکرار میشود ( تا ۲۰ بار)

```
_loop:
    mov dh, [di]
    mov dl, [bx]
    cmp dl, dh
    jne false
    inc al
false:
    inc di
    inc bx

    dec cx
    jg _loop
```

در پایان مقدار شمارنده را در ادرس result میریزیم و روتین چاپ عدد صدا زده میشود:

```
mov bx, offset RESULT
mov [bx], dl
mov al, [bx]
call print
```

روتین چاپ نیز بسیار ساده است. هر بار عدد به ۱۰ تقسیم میشود

و باقی مانده در استک پوش میشود و بعد دانه دانه pop میشوند و چاپ میشوند.

یک نمونه خروجی با اعداد بزرگ :

```
DOSBox 0.74, Cpu speed: 3000 cycles, F
ENTER YOUR NUMBER : 720000
ENTER YOUR NUMBER : 420000
GCD IS : 60000
LCM IS : 5040000
```

## پاسخ سوال ۲:

در این سوا ابتدا اعداد خواسته شده را میگیریم (با کمک روتین getNum) و در دیتا ۱ و ۲ قرار میدهیم. تابع GCD صدا میشود تا ب.م.م را محاسبه کند. ما برای هندل کردن تقسیم ۳۲ بیت بر ۳۲ بیت پس از جست و جوی زیاد و کمک گرفتن از دوستان متوجه شدیم با import کردن 386. در ابتدای برنامه یک سری رجیستر ۳۲ بیت خواهیم داشت که برای نام گذاری یک e اول نام همگی اضافه شده است. به جای تقسیم یا ضرب طولانی میتوان از idiv و imul کمک گرفت.

نکته ی مهم دیگر که ترفند زدیم این بود که برای محاسبه ک.م.م باید دو عدد ضرب میشدند و تقسیم بر ب.م.م میشد که حاصل ضرب ۶۴ بیت بود برای همین اول عدد بزرگتر را به ب.م.م تقسیم کردیم بعد در عدد دیگر ضرب کردیم تا از کد سخت و مشکل زا جلو گیری شود.

برای محاسبه ب.م.م هم بسیار ساده و از روش نردبانی کمک گرفتیم یعنی شبه کد زیر به اسمبلی تبدیل شده :

```
a = 72
b = 42
gcd = 0
while True:
    if a % b != 0:
        r = a % b
        a = b
        b = r
    else:
        gcd = b
        break
print(gcd)
```

```
77 GCD proc
78     push eax
79     push ebx
80     push edx
81     push esi
82     push edi
83
84     mov bx, offset DATA1
85     mov esi, [bx]
86     mov bx, offset DATA2
87     mov edi, [bx]
88     cmp esi, edi
89     jge false ;if(esi < edi) : swap
90     mov esi, edi
91     mov bx, offset DATA1
92     mov edi, [bx]
93     false:
94     loop1:
95         cmp edi, 0
96         jle gcd_end
97
98         mov eax, esi
99         sub edx, edx
100        idiv edi ; edx = rem
101        mov esi, edi ; a = b
102        mov edi, edx ; b = rem
103
104        jmp loop1
105
106 gcd_end:
107     mov bx, offset DATA3
108     mov [bx], esi
109
110     pop edi
111     pop esi
112     pop edx
113     pop ebx
114     pop eax
115     ret
116 GCD endp
```

برای ورودی گرفتن نیز از 0ah یا همان بافر ورودی استفاده کردیم. با توجه به جدول زیر هر بار از رقم مورد نظر باید ۴۸ تا کم شود تا مقدار عدد واقعی حاصل شود. (بخش از جدول آمده)

Digit	ASCII Value
0	48
1	49
2	50
3	51

سپس ضبدر ۱۰ میشود و با مقداری که تا الان داریم جمع میشود.

(تبدیل استرینگ به int)

....

تابع چاپ هم در سوال قبلی توضیح دادیم فقط چون اعداد ۳۲ شده اند باید از e قبل از اسم هر رجیستر استفاده کرد تا رجیستر ۳۲ بیتی مناسب استفاده شود مثلاً :

```
sub edx, edx
idiv ecx
```

خروجی سوال عبارت است از :

```
DOS
FOR DOSBox 0.74, Cpu speed: 3000 cycles, Frame
10
C:\>S_
```