

پرسش ۱

- الف) دو مورد کاربرد زبان اسمبلی در برنامه نویسی را مشخص کنید. (۰,۵)
 ب) دو مزیت زبانهای سطح بالا نسبت به زبانهای سطح پایین چیست؟ (۰,۵)
 ج) در صورت عدم اطلاع برنامه نویس از نوع CPU و سخت افزار، از چه نوع زبانی برای برنامه نویسی میتوان استفاده نمود، توضیح دهید؟ (۰,۵)
 د) آدرس منطقی و فیزیکی در CPU x86 را توضیح دهید. (۰,۵)

پرسش ۲

الف) نمایش 200.25- با توجه به قالب زیر را بنویسید. (۱)

۳۱	۳۰	۲۳	۲۲	۰
S	e'		M	

- علامت عدد s = تعداد بیت‌های اختصاص داده شده به $b=e'$, $e'=e+2^{b-1}$ ماتیس $M=0.1m$
 ب) تفاوت نمایش یک عدد بصورت BCD و شانزده شانزدهی چیست؟ (۰,۵)
 ج) 000101011101 نمایش یک عدد BCD است. نمایش آن در سیستم ده دهی و در سیستم شانزده شانزدهی چیست؟ (۰,۵)

پرسش ۳

- الف) $add\ ah,01h$ برای $ah=FFh$ چه تاثیری بر sf و zf دارد. (۰,۵)
 ب) $sar\ ah,1$ برای $ah=81h$ چه تاثیری بر of و cf دارد. (۰,۵)

پرسش ۴

برای نمایش کاراکترهای یک رشته بر روی مانیتور از طریق حافظه اختصاص داده شده به کارت گرافیک میتوان کاراکتر به کاراکتر را به آدرس $B800:0000$ انتقال داد. آدرس مذکور مشخص کننده مکان نمایش کاراکتر در گوشه بالای سمت چپ مانیتور است. کاراکتر بعدی (همان سطر به طرف سمت راست) در آدرس $B800:0002$ قرار دارد. در نتیجه هر کاراکتر در دو بایت، بایت اول اسکی کد کاراکتر و بایت دوم برای رنگ کاراکتر. برای نمونه با قرار دادن مقدار $41h$ (اسکی کد A) در $B800:0000$ و $71h$ در $B800:0001$ کاراکتر A با رنگ مشکی بر روی زمینه سفید نمایش داده میشود. هر سطر دارای 80 کاراکتر در نتیجه آدرس شروع سطر بعدی $B800:00A0$ (هر سطر 160 بایت) است. برنامه ای بنویسید برای خواندن یک سطر متن حد اکثر 80 کاراکتری با استفاده از تابع $0Ah$ اینترایت $21h$ و سپس با انتقال کاراکتر به کاراکتر رشته دریافت شده به آدرس کارت گرافیک با بکارگیری فرمانهای $LODSB$ ، $STOSW$ و $LOOP$ و قرار دادن $71H$ در ثبت AH رشته را در سطر سوم صفحه مانیتور نمایش دهید. (۴)

پرسش ۵

مراحل اجرایی تابع $09h$ و $08h$ و $02h$ و $4Ch$ اینترایت $21h$ در قالب چهار MACRO در فایل $lib.mac$ بنویسید. تابع $09h$ را با پارامتر رشته ای که باید نمایش داده شود و $02h$ را با پارامتر کاراکتری که باید نمایش داده شود بنویسید. با استفاده از ماکرو های فوق برنامه ای بنویسید که با پیغام مناسب یک کلید را دریافت و سپس با پیغام مناسب کلید فشار داده شده را نمایش دهد. (۲)

پرسش ۶

برنامه ای بنویسید در دو فایل جدا برای تبدیل یک عدد یک بایتی به رشته و ذخیره آن در سگمنت داده. فایل $main.asm$ حاوی سگمنت استک (۲۵۶ بایت)، سگمنت داده برای ذخیره $flda$ (عدد صحیح یک بایتی بدون علامت برای تبدیل) و $adadascii$ برای ذخیره عدد تبدیل شده به رشته و سگمنت کد برای $main\ proc\ far$ برای فراخوانی $proc\ itoa$ جهت تبدیل عدد به رشته. فایل $procs.asm$ حاوی $proc\ itoa\ far$ برای تبدیل عدد ذخیره شده در $flda$ به رشته و ذخیره آن در $adadascii$.
 الف) دسترسی به $flda$ توسط $proc\ itoa$ از طریق استک با روش آدرس دهی $base$ و $adadascii$ با استفاده از روش آدرس دهی $Index$. (۱)

- ب) دسترسی به $flda$ و $adadascii$ در $proc\ itoa$ با روش آدرس دهی $direct$. (۱)
 ج) دسترسی به $flda$ و $adadascii$ در $proc\ itoa$ با روش آدرس دهی $base$ از طریق سگمنت داده. (۱)
 د) موقعیت و مراحل تغییر SP و BP و محتویات استک را در الف) نمایش دهید. (۱)

پرسش V

الف) برنامه ای بنویسید برای خواندن یک سطر متن حداکثر ۸۰ کاراکتری از صفحه کلید با تابع `int 21h, 0ah` و با استفاده از فرمان های `repne` و `scasb` کاراکتر اول هر کلمه را یافته و در صورت کوچک بودن حرف، آنرا به معادل بزرگ تبدیل کند. (۱)

ب) فرمانهای لازم برای تقسیم AX بر ۱۶ با بکارگیری `shr` و ذخیره باقیمانده در DX بنویسید. (۵,۰)

ج) دو دلیل `link` نمودن فایل `obj` برای بدست آوردن فایل `exe` را نام برده و توضیح دهید. (۵,۰)

پرسش A

نمایش داخلی و شمارنده مکان برای هر سطر برنامه را مشخص کنید. (۳)

<code>datasg</code>	Segment
<code>strg</code>	<code>db 'ABCDEF'</code>
<code>a2</code>	<code>db 11111111b,0ffh</code>
<code>b2</code>	<code>dw 250,10,-1</code>
<code>x</code>	<code>dw 6 dup(10)</code>
<code>datasg</code>	Ends
<code>codesg</code>	Segment
	Assume cs:codesg,ds:datasg
<code>main</code>	<code>proc far</code>
	<code>mov ax,seg datasg</code>
	<code>mov ds,ax</code>
	<code>mov ax,b2+2</code>
	<code>Push ax</code>
<code>l1:</code>	<code>mov ax,x</code>
	<code>inc ax</code>
	<code>cmp ax,5</code>
	<code>jne l2</code>
	<code>mov cl,a2+2</code>
	<code>loop l1</code>
	<code>mov al,0</code>
<code>l2:</code>	<code>Ret</code>
<code>main</code>	<code>Endp</code>
<code>codesg</code>	Ends
	<code>end main</code>

رژرو ۲ بایت برای `datasg = B8 + datasg`

```
mov ax,seg datasg =B8 + datasg
mov ds,ax=8E D8
mov ax=A1
push ax=50
mov ax=A1
inc ax= 40
cmp ax=83F8
jne=75
mov cl=8A0E
loop=E2
mov al=B0
ret=CB
```