

یک کامپیوتر دارای حافظه اصلی به گنجایش 2^{29} کلمه ۳۲ بیتی، طول واحد آدرس پذیر ۴ بیت، و ۱۶ ثبات همه منظوره R0 تا R15 است (محتویات ثبات R0 همیشه صفر است و قابل تغییر نیست). شیوه‌های نشانی‌دهی مورد استفاده ثباتی-مستقیم، بلافاصله و شاخص‌بندی، و شیوه نمایش اعداد ممیز ثابت مکمل ۲ و ممیز شناور با استاندارد IEEE 754 Single Precision Floating-point می‌باشند. دستورات کامپیوتر در سه قالب طبق جدول زیر کد می‌شوند:

Format I	4 bits OPCODE	4 bits <i>r</i>	32 bits <i>data</i>
----------	------------------	--------------------	------------------------

۱- طول تمامی ثباتهای ماشین را تعیین کنید. (۱ نمره)

OPCODE	Mnemonic	Operation
0000	mov <i>r</i> , # <i>data</i>	$r \leftarrow \text{data};$

$$e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$$

۲- برنامه‌ای به زبان اسمبلی بنویسید که برای عدد ممیز شناور *x* مقدار

را با دقت 2^{-10} محاسبه کند. (۳ نمره)

Format II	4 bits OPCODE	4 bits <i>r1</i>	4 bits <i>r2</i>
-----------	------------------	---------------------	---------------------

۳- برنامه زیر چه می‌کند؟ مقدار اولیه تمامی ثباتها مثبت است. (۳ نمره)

OPCODE	Mnemonic	Operation
0001	mov <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r2);$
0010	add <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r1) + (r2);$
0011	sub <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r1) - (r2);$
0100	and <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r1) \text{ and } (r2);$
0101	or <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r1) \text{ or } (r2);$
0110	xor <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r1) \text{ xor } (r2);$
----- FLOATING-POINT INSTRUCTIONS -----		
0111	ctf <i>r1</i> , <i>r2</i>	$r1 \leftarrow \text{float } (r2);$ // Convert integer to float
1000	cti <i>r1</i> , <i>r2</i>	$r1 \leftarrow \text{integer } (r2);$ // Convert float to integer
1001	efp <i>r1</i> , <i>r2</i>	$r1 \leftarrow \text{fraction } (r2);$ // Move fraction part of (<i>r2</i>) into <i>r1</i>
1010	fadd <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r1) + (r2);$
1011	fsub <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r1) - (r2);$
1100	fmul <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r1) * (r2);$
1101	fdiv <i>r1</i> , <i>r2</i>	$r1 \leftarrow (r1) / (r2);$
1110	fsqrt <i>r1</i> , <i>r2</i>	$r1 \leftarrow \text{SQRT } (r2);$ // Calculate square-root of (<i>r2</i>)

```

org 0

float1: dw -1.0e+109 ; e = +109
int1:   dw 3300000h
array:  dw 100 dup(?),
        dw 5000000h, 1040000h
        dw 80100h, 3200Ah, 0

        mov R11, R0
loop:   mov R2, array-8(R1)
        dw -40.5e+112 ; e = +112
        dw 38BFC22h, 39Ah
        loop- R1, R4, loop
out1:   add R10, R5
        loop- R1, R4, loop
out2:   add R11, R5
        loop- R1, R4, loop

je R0, R0, OS_control_point

end
    
```

Format III	8 bits 1111 OPCODE	4 bits <i>r1</i>	4 bits <i>r2</i>	32 bits <i>addr</i>
------------	-----------------------	---------------------	---------------------	------------------------

OPCODE	Mnemonic	Operation
1111 0000	mov <i>r1</i> , <i>addr</i> (<i>r2</i>)	$r1 \leftarrow (M_{\text{addr}+(r2)});$
1111 0001	mov <i>addr</i> (<i>r2</i>), <i>r1</i>	$M_{\text{addr}+(r2)} \leftarrow (r1);$
1111 0010	loop- <i>r1</i> , <i>r2</i> , <i>addr</i>	$r1 \leftarrow (r1) - (r2);$ if $(r1) < (r2)$ then $PC \leftarrow \text{addr};$
1111 0011	loop+ <i>r1</i> , <i>r2</i> , <i>addr</i>	$r1 \leftarrow (r1) + 1;$ if $(r1) < (r2)$ then $PC \leftarrow \text{addr};$
1111 0100	jl <i>r1</i> , <i>r2</i> , <i>addr</i>	if $(r1) < (r2)$ then $PC \leftarrow \text{addr};$
1111 0101	jh <i>r1</i> , <i>r2</i> , <i>addr</i>	if $(r1) > (r2)$ then $PC \leftarrow \text{addr};$
1111 0110	je <i>r1</i> , <i>r2</i> , <i>addr</i>	if $(r1) = (r2)$ then $PC \leftarrow \text{addr};$
1111 0111	jne <i>r1</i> , <i>r2</i> , <i>addr</i>	if $(r1) \neq (r2)$ then $PC \leftarrow \text{addr};$
----- FLOATING-POINT INSTRUCTIONS -----		
1111 1000	fjl <i>r1</i> , <i>r2</i> , <i>addr</i>	if $(r1) < (r2)$ then $PC \leftarrow \text{addr};$
1111 1001	fjh <i>r1</i> , <i>r2</i> , <i>addr</i>	if $(r1) > (r2)$ then $PC \leftarrow \text{addr};$
1111 1010	fjnan1 <i>r1</i> , <i>r2</i> , <i>addr</i>	if $(r1) = \text{NaN}$ or $(r2) = \text{NaN}$ then $PC \leftarrow \text{addr};$
1111 1011	fjnan2 <i>r1</i> , <i>r2</i> , <i>addr</i>	if $(r1) = \text{NaN}$ and $(r2) = \text{NaN}$ then $PC \leftarrow \text{addr};$
1111 1100	fjden1 <i>r1</i> , <i>r2</i> , <i>addr</i>	if $(r1)$ or $(r2)$ is denormalized then $PC \leftarrow \text{addr};$
1111 1101	fjden2 <i>r1</i> , <i>r2</i> , <i>addr</i>	if $(r1)$ and $(r2)$ are denormalized then $PC \leftarrow \text{addr};$
1111 1110	fjinf1 <i>r1</i> , <i>r2</i> , <i>addr</i>	if $(r1) = \pm\infty$ or $(r2) = \pm\infty$ then $PC \leftarrow \text{addr};$
1111 1111	fjinf2 <i>r1</i> , <i>r2</i> , <i>addr</i>	if $(r1) = \pm\infty$ and $(r2) = \pm\infty$ then $PC \leftarrow \text{addr};$

۴- برنامه اسمبلی سوال ۳ را به کد ماشین ترجمه

کنید. (۳ نمره)