1. Em um computador, as informações são armazenadas na memória como seqüências de bits. Essas seqüências podem representar diferentes tipos de dados ou instruções de máquina, dependendo da interpretação adotada. Considere as quatro seqüências binárias de 32 bits apresentadas a seguir:

Seqüência	Representação binária
A	10111111110001010010001010000011
B	11000000000001010010001100000011
C	00000000011000101000001110110011
D	11000000011101010010001000100011

Responda às questões a seguir, apresentando os cálculos e justificando suas respostas. Observações importantes:

- Considere que a arquitetura RISC-V é de 32 bits (RV32I), com registradores de 32 bits e memória endereçada por bytes.
- Os valores imediatos das instruções de acesso à memória são expressos em bytes em complemento de dois. Em especial valores imediatos que representam um *offset* podem ser negativos.
- Considere que todos os endereços acessados são válidos e estão adequadamente alinhados.
- Quando for interpretar a seqüência de bits como números de ponto flutuante, apresente os resultados decimais com pelo menos seis casas decimais.
- Utilize a página em anexo para consultar formatos e valores importantes da arquitetura RISC-V e da representação de números de ponto flutuante no padrão IEEE-754.
- Você pode usar a calculadora, mas deve apresentar os cálculos.

(a) Interprete as seqüências **A** e **C** como números inteiros de 32 bits com sinal, representados em complemento de dois. Determine seus respectivos valores decimais.

(b) Interprete as seqüências **B** e **D** como números de ponto flutuante de 32 bits, no padrão IEEE 754 de precisão simples (binary32 ou *single precision*). Para cada seqüência, identifique o bit de sinal, o expoente armazenado, o expoente real e a fração do significando. Determine o valor decimal correspondente.

(c) Considere agora que as quatro seqüências representam instruções de máquina da arquitetura RISC-V (32 bits). Decodifique cada seqüência binária e escreva sua instrução equivalente em linguagem *assembly* RISC-V, identificando os registradores e os valores imediatos utilizados.

(d) Considerando que as quatro instruções sejam executadas sequencialmente, na ordem **A**, **B**, **C** e **D**, explique o que o programa realiza. Em sua explicação, indique os endereços de memória acessados, os registradores envolvidos e a operação aritmética executada.

(e) Suponha que, antes da execução das instruções, o registrador `x10` contenha o endereço hexadecimal `0x20001000` e que a memória contenha os seguintes valores inteiros de 32 bits:

- No endereço `0x20000BFC`: valor decimal 12.
- No endereço `0x20000C00`: valor decimal -7.
- No endereço `0x20000C04`: valor decimal 0.

Determine o valor final dos registradores modificados pelas instruções e o conteúdo armazenado no endereço `0x20000C04` após a execução completa do programa.

ANEXO INFORMATIVO

1. Principais formatos das instruções em RISC-V

31 27	26 25	24 20	19 15	14 12	11 7	6 0	
funct7		rs2	rs1	funct3	rd	opcode	R-type
imm[11:0]			rs1	funct3	rd	opcode	I-type
imm[11:5]		rs2	rs1	funct3	imm[4:0]	opcode	S-type
imm[12|10:5]		rs2	rs1	funct3	imm[4:1|11]	opcode	B-type
imm[31:12]					rd	opcode	U-type
imm[20|10:1|11|19:12]					rd	opcode	J-type

2. Formato, opcode, funct3 e funct7 de algumas instruções RISC-V

Inst	Name	FMT	Opcode	funct3	funct7	Description (C)	Note
add	ADD	R	0110011	0x0	0x00	rd = rs1 + rs2	
sub	SUB	R	0110011	0x0	0x20	rd = rs1 - rs2	
xor	XOR	R	0110011	0x4	0x00	rd = rs1 ^ rs2	
or	OR	R	0110011	0x6	0x00	rd = rs1 | rs2	
and	AND	R	0110011	0x7	0x00	rd = rs1 & rs2	
addi	ADD Immediate	I	0010011	0x0	–	rd = rs1 + imm	
lb	Load Byte	I	0000011	0x0	–	rd = M[rs1+imm][0:7]	
lbu	Load Byte (U)	I	0000011	0x4	–	rd = M[rs1+imm][0:7]	zero-extends
lw	Load Word	I	0000011	0x2	–	rd = M[rs1+imm][0:31]	
sw	Store Word	S	0100011	0x2	–	M[rs1+imm][0:31] = rs2[0:31]	

3. Registradores de uso geral da RISC-V:

Register	ABI Name	Description	Saver
x0	zero	Zero constant	—
x1	ra	Return address	Caller
x2	sp	Stack pointer	Callee
x3	gp	Global pointer	—
x4	tp	Thread pointer	—
x5-x7	t0-t2	Temporaries	Caller
x8	s0 / fp	Saved / frame pointer	Callee
x9	s1	Saved register	Callee
x10-x11	a0-a1	Fn args/return values	Caller
x12-x17	a2-a7	Fn args	Caller
x18-x27	s2-s11	Saved registers	Callee
x28-x31	t3-t6	Temporaries	Caller

4. Formatos de ponto flutuante em IEEE-754:

Single Precision: 32 bits no total; 1 bit para o sinal, 8 bits para expoente com viés de 127, e 23 bits para a fração.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s	expoente								fração																						
(1)	(8 bits, viés de 127)								(23 bits)																						

$$\text{Viés} = 2^{(\text{bits do } e)-1} - 1$$

$$\text{Forma normalizada} = \underbrace{1}_{\text{MSB}} \cdot \overbrace{\text{bbb}\ldots\text{bbb}}^{\text{fração}} \times 2^{e} \quad \text{(significando)}$$