HONOR CODE
Questions Sheet.

A. Easy. Lets C. [6 Points]
  
  Q1-6

B. Easy. RISCV Magic. [7 Points]
  
  7. What is the minimum set of registers need to be stored onto the stack at this point: line 4. ? [1]
  8. What is the minimum set of registers need to be stored onto the stack at this point: line 14. ? [1]
  9. What is the minimum set of registers need to be restored from the stack at this point: line 30 ? [1]
  10. Assume you have the prologue and epilogue correctly coded. You set a breakpoint at 'line 6: CHECK". What does result contain when your program pauses at the breakpoint? [4]

C. RISC-V Instructions Encoding [5 points]
  
  11. For the instruction line 7: beq s1, x0, end. What is the immediate field
  12. What is actual opcode, rs1 and rs2 (not pseudo-names) ?
  13. What is funct7 and funct3 ?
  14. What is the instruction corresponding to 0x0004A503 ?

D. RISC-V Custom Opcodes [6]
  
  15. What is the minimum bits would be required for the opcode field? [1]
  16. If the opcode was encoded 5 bits and we would like to support the usual R-type instructions, 2 source and 1 destination. What is the maximum number of registers we can use? [1]
  17. Given the opcode is 5 bits wide. Each register field is 3 bits. What is the offset in terms of bytes that the branch instruction can jump forward. Note that instructions are 19 bits wide. [2]
  18. What is the max negative offset that an I instruction can use ? Opcode bits is same as Q15. Assume that register width is same as Q16. [2]

E. Easy Floating Point [5 points]
  
  19. What is the smallest non-zero positive value that can be represented? [1]
  20. How do you represent the number 4.5 ? [1]
  21. How do you represent $-2^{-9}$ [1]
  22. How many numbers can this 10 bit floating point represent in the range 1 ≤ f < 8). Hint: Write does the floating point expressions for 1 and 8 and the answer should be apparent. [1]
  23. Sort the following numbers 0x300, 0x100, 0x104, 0x328, 0x12c smallest to largest [1]

F. Unsigned/Signed Numbers [5]
  
  24. Represent -133 as a 8-bit NOT number. [1]
  25. Represent -124 as 8-bit NOT number. [1]
26. What is the range of numbers represented by 8 bit NOT8. [1]
27. What is the representation that requires fewest number of bits needed to cover the given range 0 to 10. [1]
28. What is the representation that requires fewest number of bits needed to cover the given range -1 to 4. [1]

G. Bitgames. Write down single line expressions that calculate the following. [4]
29. NegativeFloat(x) - Return bit-level equivalent of expression -f for floating point argument f. Assume f is N bits
30. is_float_power_of_2(float x, int e, int m).

H. Assembler Linker Compiler [6]
32. What is the symbol and relocation table ? [2]
33. Replace the labels of PC-relative targets with their immediate values. What is the offset value of bnez at address 0x20? Write your answer in decimal. [2]

G. RISCV
34. What is the value of s2 at the end of the program ? Write in hex (e.g., 0xFFFF) [3]

HONOR CODE

- I have not used any online resources during the exam.
- I have not obtained any help either from anyone in the class or outside when completing this exam.
- No sharing of notes/slides/textbook between students.
- NO SMARTPHONES.
- CANVAS ANSWERS WILL BE LOCKED AFTER 1ST TRY.

Questions Sheet.

Read all of the following information before starting the exam:

- For each question fill out the appropriate choice or write text on Canvas page. Also type clearly on in the exam on the appropriate text.
- IF THE MULTIPLE CHOICE ANSWER IS WRONG WE WILL MARK THE ANSWER WRONG. IF THE MULTIPLE-CHOICE ANSWER IS CORRECT, WE WILL READ THE WRITTEN PORTION.
- Show all work, clearly and in order, if you want to get full credit.
- We reserve the right to take off points if we cannot see how you logically got to the answer (even if your final answer is correct).
- Circle or otherwise indicate your final answers.
Please keep your written answers brief; be clear and to the point.
I will take points off for rambling and for incorrect or irrelevant statements. This test has seven problems.
A. Easy. Lets C. [6 Points]

Q1-6

Grayscale color values can be represented as an ascii value between 0---255. Consider square images of \(N \times N\) pixels.

We can organize these 2D images into a 1D array of \(N^2\) elements. Each element is an 8 bit number.

```c
char *img = malloc(sizeof(char) * 4);
img[0] = 0xA;
img[1] = 0xB;
img[2] = 0xC;
img[3] = 0xD;
```

Fill out the following function \(\text{tile}\). It returns a new, larger image array, which is the same image tiled \(\text{rep}\) times in both the \(x\) and \(y\) direction. You may or may not need all of the lines. For a better idea of what must be accomplished, consider the above example:

```c
char *t_img = tile(img, 2, 2);
// The contents of tiled_image
would then look like:
[0xA, 0xB, 0xA, 0xB]
[0xC, 0xD, 0xC, 0xD]
[0xA, 0xB, 0xA, 0xB]
[0xC, 0xD, 0xC, 0xD];
```

Now fill-in-the-blanks for the code shown below on canvas.

```c
char *tile(char *b, int n, int rep) {
    int w = ______Q1_______;
    char *t_img = malloc(________Q2______);
    for (int j = 0; j < w; j++) {
        for (int i = 0; i < w; i++) {
            int x = ___Q3___;
            int y = ___Q4___;
            int loc = _Q5___;
            t_img[loc] = b[x + y*n];
        }
    }_______Q6 (could be multiple lines)______;
    return t_img;
}
```
Q1: n * rep
Q2: w * w or w * w * sizeof(char)
Q3: i%n
Q4: j%n
Q5: j * w + i
Q6: Blank or nothing

B. Easy. RISCV Magic. [7 Points]

Assume we have two arrays input and output.

```c
int *input = malloc(8*sizeof(int));
int *result = malloc(8, sizeof(int));
for (int i = 0; i < 8; i++) {
    input[i] = i;
}
```

Study the following RISC-V code shown below and answer the questions.
You can assume a0:input a1:result a2:8
7. What is the minimum set of registers need to be stored onto the stack at this point: line 4. ? [1]

   t0, t1, t2

8. What is the minimum set of registers need to be stored onto the stack at this point: line 14. ? [1]

   s0 s1

9. What is the minimum set of registers need to be restored from the stack at this point: line 30 ? [1]

   s0 s1
10. Assume you have the prologue and epilogue correctly coded. You set a breakpoint at `line 6: CHECK`. What does result contain when your program pauses at the breakpoint? [4]

0 1 3 6 10 15 21 28

C. RISC-V Instructions Encoding [5 points]

Consider the standard RISC-V encoding below. Standard 32 bit instructions. Answer questions below

```
1 | main:
2 |   mv s1, a0
3 |   addi t2, t2 4
4 | Start:
5 |   beq s1, x0, End
6 |   lw a0, 0(s1)
7 |   addi a0, a0, 4
8 |   add s1, t2, s1
9 |  lw s1, 0(s1)
10 | jal x0, Start
11 | End:
12 |   addi a0, a0, 10
13 | ecall
```

11. For the instruction line 7: `beq s1, x0, end`. What is the immediate field 24?

12. What is actual opcode, rs1 and rs2 (not pseudo-names)?

   opcode: 0x63
   rs1: 01001 (x9)
   rs2: 00000 (x0)

13. What is funct7 and funct3?

   f3 - 0 f7 N/A

14. What is the instruction corresponding to `0x0004A503`?

   lw x10 0(x9) or
   lw a0, 0(s1) or
   line 6
Prof. Shriraman is designing a new CPU with fewer operations. He decides to adapt and rethink the design of RISC-V instruction. He only needs to support 9 different operations: ADD, MUL, XOR, LD, SW, LUI, ADDI, MULI and BLT. He decides that each instruction should be 19 bits wide.

The fields in each instruction are listed below (no funct3 and funct7):

- **R-type**: rs2, rs1, rd, opcode
- **I-type and Loads**: imm, rs1, rd, opcode
- **S-type**: imm, rs2, rs1, opcode
- **B-type**: imm, rs2, rs1, opcode
- **U-type**: imm, rd, opcode
- **UJ-type**: imm, rd, opcode

15. **What is the minimum bits would be required for the opcode field?** [1]

4 bits

16. **If the opcode was encoded 5 bits and we would like to support the usual R-type instructions, 2 source and 1 destination. What is the maximum number of registers we can use?** [1]

4 bits. 16 registers.

17. **Given the opcode is 5 bits wide. Each register field is 3 bits. What is the offset in terms of bytes that the branch instruction can jump forward. Note that instructions are 19 bits wide.** [2]

\[
19 - (3 + 3 + 5) = 8 \text{ bits for imm} \\
2^7 - 1 = 127
\]

18. **What is the max negative offset that an I instruction can use? Opcode bits is same as Q15. Assume that register width is same as Q16.** [2]

Opcode: 4 bits
Register: 4 bits. AddI requires 1 rd and 1 rs.

\[
\text{Imm} = 19 - 4 + 4 + 4 = 7 \text{ bits. } -2^6
\]

**E. Easy Floating Point [5 points]**
The TAs get tired of having to convert floating-point values into 32 bits. As a result they propose the following smaller floating-point representation which is useful in a number of machine learning applications. It consists of a total of 10 bits as show below. Exponent is biased similar to conventional floating point.

<table>
<thead>
<tr>
<th>Sign</th>
<th>Exponent</th>
<th>Mantissa</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 bit</td>
<td>5 bits.</td>
<td>4 bits.</td>
</tr>
</tbody>
</table>

- Numbers are rounded to the closest representable value. Any numbers that have 2 equidistant representations are rounded down towards zero.

19. What is the smallest non-zero positive value that can be represented? [1]

\[ 2^{-18} = \text{Denormalized form. exponent is } 2^{-14} \text{ and mantissa is } 2^{-4} \]

20. How do you represent the number 4.5 ? [1]

0x112

0 10001 0010

exponent: 17 - 15 = 2
\[ 2^2 \times 1.125 = 4.5 \]

21. How do you represent \(-2^{-9}\) [1]

0x260

1 00110 0000

22. How many numbers can this 10 bit floating point represent in the range \(1 \leq f < 8\). Hint: Write does the floating point expressions for 1 and 8 and the answer should be apparent. [1]

- 1 is 0x70 . 0 0111 0000 = \(1 \times 2^0\)
- 8 is 0xa0 . 0 1010 0000 = \(1 \times 2^3\)

There are 16 numbers, 0111 0000
There are 16 numbers, 1000 0000
There are 16 numbers, 1001 0000

Total - 48
23. Sort the following numbers 0x300, 0x100, 0x104, 0x328, 0x12c smallest to largest

Shown are the hex representations.

0x300 (-2), 0x100 (2), 0x104 (2.5), 0x328 (-12), 0x12c (14)

F. Unsigned/Signed Numbers [5]

Suppose that we define a new number format, NOT. Negative numbers are represented by the binary NOT ~ of the binary representations of their corresponding positive numbers. Like 2's complement most significant bit of NOT denotes the number's sign (0 for positive, 1 for negative). Answer the following questions.

\[
\text{NOT (\sim)} \]

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

24. Represent -133 as a 8-bit NOT number. [1]

N/A

25. Represent -124 as 8-bit NOT number. [1]

124 - 0111 1100
-124 - 1000 0011

26. What is the range of numbers represented by 8 bit NOT8. [1]

-127 to +127

27. What is the representation that requires fewest number of bits needed to cover the given range 0 to 10. [1]

Unsigned. 4 bits

28. What is the representation that requires fewest number of bits needed to cover the given range -1 to 4. [1]
G. Bitgames. Write down single line expressions that calculate the following. [4]

29. NegativeFloat(x) - Return bit-level equivalent of expression -f for floating point argument f. Assume f is N bits

\[ N \mid 0x1 << (31) \]

30. is\_float\_power\_of\_2(float x, int e, int m).

Normalized number: \( 1.m \times (2^e - 1 - bias) \)

\[ (x \& (2^e-1) << m) != (2^e -1)) && (x \& (2^m -1)) == 0 \]

H. Assembler Linker Compiler [6]

```
1  .data
2  str: string "The sum of 1..100 is %d \n"
3
4  .text
5  main:
6    add sp,sp -4
7    sw ra, 0 (sp)
8    mv a1, zero
9    li x9, 100
10   j check
11
12  loop:
13    mul s2,x9,x9
14    add a1,a1,s2
15    add x9,x9,-1
16
17  chk:
18    bnez x9, loop
19    la a0, str
20    jal printf
21    mv a0,zero
22    lw ra, 0(sp)
23    addi sp,sp,4
24    ret
```

8 instructions, 7 registers (sp,ra,a1,zero,s2,a1,a0)
mv,li,j,bnez,la,jal,mv,ret

32. What is the symbol and relocation table? [2]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>main</td>
<td>0x0</td>
</tr>
<tr>
<td>loop</td>
<td>0x14</td>
</tr>
<tr>
<td>chk</td>
<td>0x20</td>
</tr>
<tr>
<td>str</td>
<td>0x80</td>
</tr>
</tbody>
</table>

Relocation

<table>
<thead>
<tr>
<th>Inst</th>
<th>Addr</th>
<th>Dep</th>
</tr>
</thead>
<tbody>
<tr>
<td>la</td>
<td>0x24</td>
<td>std</td>
</tr>
<tr>
<td>jal printf</td>
<td>0x28</td>
<td>printf</td>
</tr>
</tbody>
</table>

33. Replace the labels of PC-relative targets with their immediate values. What is the offset value of bnez at address 0x20? Write your answer in decimal. [2]

-12

G. RISCV

34. What is the value of s2 at the end of the program? Write in hex (e.g., 0xFFFF) [3]

Initial values: s0 = 0xC, s1 = 0xA, a0 = 3, a1 = 0, a2 = 2
.text
la a3, start
start:
   beq a1,a0,End
lw  a4,20(a3)
slli a5,a2,12
add a4,a4,a5
sw a4, 20(a3)
add s2,s1,s0
addi a1,a1,1
j start

End:
   addi a0,zero,10
ecall

Answer: 0xE