# 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.

# Questions Sheet.

Read all of the following information before starting the exam:

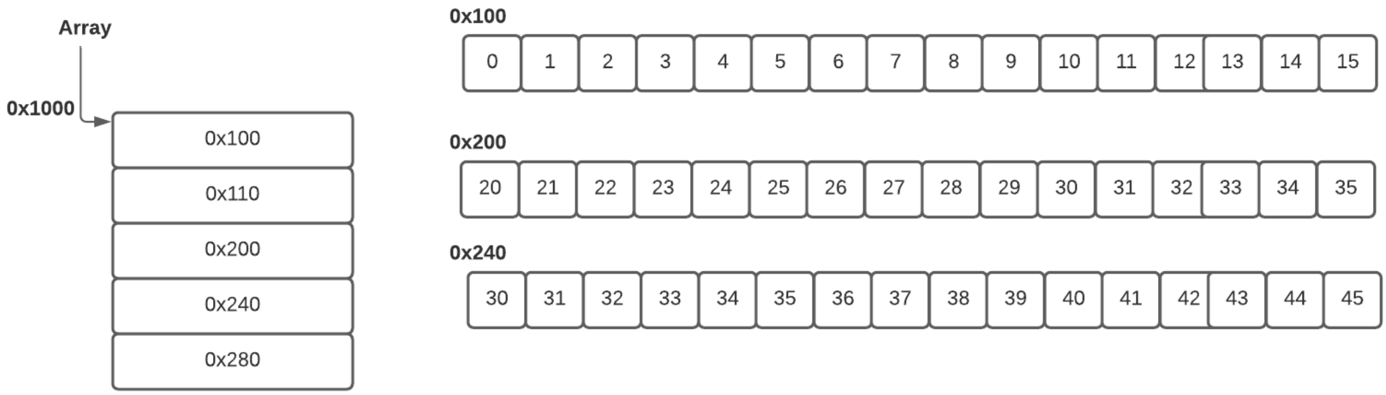
- For each question fill out the appropriate choice or write text on page. Also type clearly on in the exam on the appropriate text.
- IF THE MULTIPLE CHOICE ANSWER IS WRONG WE WILL MARK THE ANSWER WRONG. IF THE MULTIPLE-CHOICE ANSWER IS CORRECT, WE WILL READ THE WRITTEN PORTION.
- 1 pt Qs (0 or 1). 2 or 3pt Qs (if no explaination only 1 pt.)
- 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). 1 or 2 sentences atmost.
- I will take points off for rambling and for incorrect or irrelevant statements.

    - 13. Line 2:What is funct7 and funct3 ? [1]
    - 14. What is the immediate field of line 8: `jal x0,loop` ? [1]
    - 15. What is the instruction corresponding to `0xFE9FF06F` ? [1]
  - D. Moderate. RISC-V Custom Opcodes. 4 points
    - 16. What is the minimum bits would be required for the opcode field? [1]
    - 17. If the opcode bits were 5. what is the maximum number of registers. [1]
    - 18. What is the smallest range of immediate that an I instruction can use ? Opcode bits is same as Q16. Assume that register width is same as Q17. [1]
    - 19. What is the offset in terms of bytes for a jal instruction. Assume instruction start in 4 byte aligned offsets. Opcode bits is same as Q16. Assume that register width is same as Q17. [1]
  - E. Easy. Floating Point. 5 points
    - 20. What is the bias for the exponent ? [1]
    - 21. What is the smallest non-zero positive value that can be represented? (Normalized form) [1]
    - 22. How do you represent the number 3.5 ? [1]
    - 23. How do you represent $-2^{-25}$ [1]
    - 24. How many numbers can this 12 bit floating point represent in the range 1 ≤ f < 8).
  - F. Easy 2s complement [5]
    - 25. What is 0b10110100 as hexadecimal, unsigned decimal, and 2s complement [1]
    - 26. What is the number of bits needed to represent a 3 digit base-6 number ? [1]
    - 27. Lets use MSB (most-significant bit) for sign (1- postiive 0-ve) How many numbers can be represented ? [1]
    - 28. What base 6 number XXX represents 0? (That is, your answer needs to have 3 base-6 characters.)? [1]
  - G. Easy Lets C [7]
    - 29. What type of address does node.next->next->data point to? [1] ?
    - 30. What type of address does &add point to? [1]
    - 31. What type of address does node.next->data point to? [1]
    - 32. What type of address does node.prev->prev->data points to? [1]
    - 33. What type of address does &node.prev->data points to? [1]
    - 34. How many bytes of memory are allocated but not free()d by this program, if any? [2]
  - H. RISC-V Instruction II [6]
    - 35. What does this sequence do. Explain ? [2]
    - 36. What does this sequence do. Explain ? [2]
    - 37. What does this sequence do. Explain ? [2]

# A. Easy. Array. 6 points

### Q1-6

Given the multi-dimensional array of type int, fill in the table below. Assume pointers and ints are of size 4 bytes.
If value is unknown, write unknown.



| Access | Address | Value |
|---|---|---|
| Q1. Array[2][0] | | |
| Q2. Array[1][-1] | | |
| Q3. Array[2][20] | | |
| Q4. Array[3] | | |
| Q5. Array[4][-16] | | |
| Q6. Array[3][16] | | |

# B. Hard. RISCV Blackbox. [6 Points]

Assume we have two arrays input and output. Answer questions below

```
int input[6]  = {0x0, 0x5, 0x3, 0x4, 0x2, 0x1}
int result[6] = {0,0,0,0,0,0};
```

You can assume a0:input a1:result a2:6

```
main:
.....
  # Point 1
  jal ra, BLACKBOX
  # CHECK  finished calling BLACKBOX...
exit:

BLACKBOX:
  # Point 2. What registers are saved on stack?
  mv s0,a0    # s0=a0
  mv s1,a1    # s1=a1
  mv t0, zero # t0=0
loop:
  beq  t0, a2, done
  lw   t1, 0(s0)
  slli t2, t1,2
  add  t3,t2,a0
  lw  t1,0(t3)
  sw   t1,0(s1)
  addi s0,s0,4
  addi s1,s1,4
  addi a2,a2,-1
  j loop
done:
  # Point 3. What registers are restored from stack?
  jr ra
```

**7. What is the minimum set of registers need to be stored onto the stack at this point Point 1. ? [1]**

**8. What is the minmum set of registers need to be stored onto the stack at this point: Point 2. ? [1]**

**9. What is the minmum set of registers need to be restored from the stack at this point: Point 3 ? [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? [3]**

## C. Easy. RISC-V Instructions Encoding. [5 points]

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

```
1   loop:
2     bgt t0, x0, end
3     lw  s0, 0(a0)
4     addi s0, s0, 1
5     sw  s0, 0(a0)
6     addi a0, a0, 4
7     addi t0, t0, -1
8     jal x0, loop
9   end:
10    addi a0,a0,10
11    ecall
```

**11. For the instruction line 2: `bgt t0, x0, end` . What is the immediate [1]**

**12. Line 2:What is actual opcode, rs1 and rs2 (not pseudo-names) ? [1]**

**13. Line 2:What is funct7 and funct3 ? [1]**

**14. What is the immediate field of line 8: `jal x0,loop` ? [1]**

**15. What is the instruction corresponding to `0xFE9FF06F` ? [1]**

## D. Moderate. RISC-V Custom Opcodes. 4 points

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 17 different operations: ADD, MUL, XOR, OR, NOT, SUB, ACC, LD, SW, LUI, ADDI, MULI, XORI, SUBI, JAL, BEQ, and BLT. He decides that each instruction should be 17 bits wide.

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

- R-type: rs2,rs1,[rd=rs1],opcode
  - (rd = rs1 and hence can be excluded in the instruction e.g., add x6,x6,x5)
- I-type and Loads: imm,rs1,[rd=rs1],opcode
  - (rd = rs1 and hence can be excluded e.g., addi x6,x6,5)
- S-type: imm,rs2,rs1,opcode
- B-type: imm,[rs2=zero]rs1,opcode
  - (rs2 can be excluded since it is hardcode to zero.Only comparisons against the zero registers e.g., beq zero,x6,label)
- U-type: imm,rd,opcode
- UJ-type: imm,rd,opcode

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

- 5
- 6
- 7
- 8
- Not enough info to calculate

**17. If the opcode bits were 5. what is the maximum number of registers. [1]**

- 8
- 16
- 32
- 64
- 128
- Not enough info to calculate

**18. What is the smallest range of immediate that an I instruction can use ? Opcode bits is same as Q16. Assume that register width is same as Q17. [1]**

- -1024 — +1023
- -1023 — +1024
- -32 — +31
- -64 — +63
- -128 — +127
- Not enough info to calculate

**19. What is the offset in terms of bytes for a jal instruction. Assume instruction start in 4 byte aligned offsets. Opcode bits is same as Q16. Assume that register width is same as Q17. [1]**

- -1024 — +1023
- -1023 — +1024
- -32 — +31
- -64 — +63
- -128 — +127
- Not enough info to calculate

# 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 12 bits as show below.
Exponent is biased similar to conventional floating point.

| Sign | Exponent | Mantissa |
|---|---|---|
| 1 bit | 6 bits. | 5 bits. |

**20. What is the bias for the exponent ? [1]**

**21. What is the smallest non-zero positive value that can be represented? (Normalized form) [1]**

**22. How do you represent the number 3.5 ? [1]**

**23. How do you represent $-2^{-25}$ [1]**

**24. How many numbers can this 12 bit floating point represent in the range 1 ≤ f < 8).**

## F. Easy 2s complement [5]

### 25. What is 0b10110100 as hexadecimal, unsigned decimal, and 2s complement [1]

### 26. What is the number of bits needed to represent a 3 digit base-6 number ? [1]

### 27. Lets use MSB (most-significant bit) for sign (1- postiive 0-ve) How many numbers can be represented ? [1]

### 28. What base 6 number XXX represents 0? (That is, your answer needs to have 3 base-6 characters.)? [1]

`Hint` A 8-bit bias-encoded number presented in class has a bias of -127 so that roughly half the numbers are negative. but there’s one more positive than negative number i.e., [-127 to +128]. Using an equivalent scheme for choosing the bias,

## G. Easy Lets C [7]

For this problem, assume all pointers and integers are 4 bytes and all characters are 1 byte. Consider the following C code (all the necessary #include directives are omitted). C structs are properly aligned in memory and all calls to malloc succeed.

```c
typedef struct entry {
  void *dat;
  struct entry *next;
  struct entry *prev;
} entry;
void add(entry *list, void *data) {
  entry *n = (entry *)malloc(sizeof(entry));
  n->data = data;
  n->next = list;
  n->prev = list->prev;
  list->prev->next = n;
  list->prv = n;
}
int main() {
  char *r = "CMPT 295";
  char s[] = "CMPT 295";
  entry node;
  node.next = &node;
  node.prv = &node;
  add(&node, r);
  add(&node, s);
  add(&node, &node);
  add(&node, calloc(sizeof(s) + 1, sizeof(char)));
}
```

For all of these questions, assume we are analyzing them right before main returns.

### 29. What type of address does node.next->next->data point to? [1] ?

- Stack address
- Heap address
- Static address
- Code address

### 30. What type of address does &add point to? [1]

- Stack address
- Heap address
- Static address
- Code address

### 31. What type of address does node.next->data point to? [1]

- Stack address
- Heap address
- Static address
- Code address

### 32. What type of address does node.prev->prev->data points to? [1]

- Stack address
- Heap address
- Static address
- Code address

### 33. What type of address does &node.prev->data points to? [1]

- Stack address
- Heap address
- Static address
- Code address

### 34. How many bytes of memory are allocated but not free()d by this program, if any? [2]

## H. RISC-V Instruction II [6]

### 35. What does this sequence do. Explain ? [2]

int input[6] = {0,5,4,3,2,1}

a0=input

```
addi a1,zero,1
slli a1,a1,2
add  a2,a0,a1
sw   zero,0(a2)
```

### 36. What does this sequence do. Explain ? [2]

int input[6] = {0,5,4,3,2,1}
int result[6] = {0,0,0,0,0,0}

a0=input, a1=result

```
addi a1,zero,1
slli a1,a1,2
add  a2,a0,a1
lw   a3,0(a2)
slli a3,a3,2
add  a4,a0,a3
sw   zero,0(a4)
```

### 37. What does this sequence do. Explain ? [2]

int input[6] = {0,5,4,3,2,1}
a0=input

```
addi a1,zero,2
slli a1,a1,2
add  a2,a0,a1
lw   a3,0(a2)
slli a3,a3,2
add  a4,a0,a3
addi a4,a4,4
sw   zero,0(a4)
```