Q: Are all of these enough to get full marks in the exam?

A: NO. This is a practice sheet. Meaning, you can practice all you want using the questions from this sheet. However, doing well in exams depends upon your ability to understand a question, formulate an answer, and express it correctly. You see, these are humane skills which cannot be guaranteed by completing a practice sheet only. But yeah, Best of luck anyway.

**Chapter 2 (Instructions: Language of the Computer)**

## **Question - 1:**

Construct the equivalent RISC-V code of the following C code. Once you have the RISC-V code, identify type of each instruction and encode them accordingly.

A[7] = A[2] + A[B[8]] + 10;

B[i] = A[3] - 8;

Base addresses of array A and B are in register X20 and X21 and i is in register X22

## **Question - 2:** (Skip)

Construct the equivalent RISC-V code of the following C code.

for (i = 8; i > 0 ; i--) {

if ( A[i] == i){

A[2] = A [B[3]] ;

}

}

Base addresses of array A and B are in register X20 and X21 . Also consider i is in register X22.

**Question - 3:**

Construct the equivalent RISC-V code of the following C code.

if ( A[i] < i){

A[2] = A [B[3]] ;

}

Base addresses of array A and B are in register X20 and X21 . Also consider i is in register X22.

**Question - 4:**

Construct the equivalent RISC-V code of the following C code.

if ( A[3] != A[6]){

if (A[3] == 0) {

A[3] = A[3] + 2;

}else{

A[6] = A[6] / 16;

}

}else{

A[6] = A[6] \* 8

}

Base addresses of array A and B are in register X20 and X21 .

**Question - 5:** (Skip)

Construct the equivalent RISC-V code of the following C code.

Main () {

int x = 0;

int y = 9;

int z = addition(x, y);

}

int addition (int a, int b) {

int c = a + b;

return c;

}

Variables x, y, z are stored in X20,X21 and X22 registers. Argument x, y are passed using register X13, X14

Variable c from the addition function also uses register X21

**Question - 6:**

Write RISC-V assembly code that checks if the number stored in register **X25** is **even** or not. If **even** then store **1** in register **X26** otherwise store **0**.

**Question - 7:**

ADD X25 , X25 , X0 . Can you make this instruction faster? If yes, Write the updated instruction?

**Question - 8:**

| Memory Location | Code | Line Number | Machine Code |
| --- | --- | --- | --- |
|  | ADDI **X5** , X**0** , 5 | 1 |  |
|  | ADDI **X6** , X**0** , 1 | 2 |  |
|  | ADDI **X25** , X**0** , 0 | 3 |  |
|  | Loop: BLT **X5** , **X6** , loopBreak | 4 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_XXX\_\_\_\_\_\_XXXXXXX |
|  | ADDI X**25** , X**25** , 1 | 5 |  |
| #7080 | ADDI X**5** , X**5** , -1 | 6 |  |
|  | BEQ **X0** , **X0** , Loop | 7 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_XXX\_\_\_\_\_\_XXXXXXX |
|  | loopBreak: | 8 |  |

1. What is the value of **PC** while executing line2? Answer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Fill up the machine codes corresponding to line4 and line7 in the table above.

**Question - 9:**

| Memory Location | Code | Line Number |
| --- | --- | --- |
|  | Loop: |  |
|  | SLLI **X10** , X**22** , 3 | 1 |
|  | ADD **X10** , X**10** , X**25** | 2 |
|  | LD **X9** , 0(X**10**) | 3 |
|  | BNE **X9** , **X24** , Exit | 4 |
| #80016 | ADDI X**22** , X**22** , 1 | 5 |
|  | BEQ **X0** , **X0** , Loop | 6 |
|  | Exit: |  |

1. Fill up the memory locations.
2. Find the SB-type instructions from the above code and encode them accordingly.

Given,

I. opcode = (103)10, funct3 = (000)2 opcode for BEQ

II. opcode = (103)10, funct3 = (001)2 opcode for BNE

**Question - 10:**

Write necessary RISC-V instructions to store the value (1111 1111 0000 1111 11)2 in X20 register.

**Question - 11:**

Show how the value 0xabcdef12 would be arranged in memory in RISC-V machine.

**Question - 12:**

For the RISC-V assembly instructions below, what is the corresponding C/high level statement?

| slli x30, x5, 3  add x30, x10, x30  slli x31, x6, 3  add x31, x11, x31  ld x5, 0(x30)  addi x12, x30, 8  ld x30, 0(x12)  add x30, x30, x5  sd x30, 0(x31) | Assume that the  variables f, g, h, i, and j are assigned to registers x5, x6, x7,  x28, and x29, respectively. Assume that the base address of the  Arrays A and B are in registers x10 and x11, respectively. |
| --- | --- |