

Name: _____

1. (1 pt.)

- **Read all material carefully.**
- *If in doubt whether something is allowed, ask, don't assume.*
- You may refer to **your** books, papers, and notes during this test.
(No sharing of material.)
- **E-books** may be used **subject to the restrictions** noted in class. (Briefly, do only those things with an e-book that are just as easily done with a physical book.)
- **Computers of any kind** (including tablets, phones, and similar devices) are **not permitted** except when used exclusively as e-book readers.
- **Network access** of any kind (cell, voice, text, data, ...) is **not permitted**.
- Write, and draw, carefully. Ambiguous or cryptic answers receive zero credit.
- Use class and textbook **conventions** for notation, algorithmic options, etc.
- Unless otherwise indicated, phrases such as *assembly instructions* and *machine code* refer to **RISC-V architecture** and related details as described in the class and the textbooks, and as executed in the RARS(M) environment.
- Questions that ask for **explanations** allocate a sizable fraction of points to those.
(Answers without explanations will score very poorly.)
- Budget your **time**, noting that *number of points = number of minutes*.

Write your name in the space provided above.

Do not write anything else on this page.

WAIT UNTIL INSTRUCTED TO CONTINUE TO REMAINING QUESTIONS.

(Do not view any other pages.)

Do not write on this page.
(It is for use in grading only.)

Q	Full Score
1	1
2	19
3	25
total	45

2. (19 pts.) Provide a *binary* representation of the *data segment* generated corresponding to the following assembly code fragment (by the RARS(M) assembler), assuming that the memory address of the start of the data segment is, in hexadecimal, 10010000. Your answer must clearly indicate the address of each byte (corresponding to this fragment).

```
1  .data
2  tenk: .word 10000
3  dogs: .asciz "dog"
```

Explain your answer by indicating how that representation is derived. (Reminders: Textbook/class conventions; no computer use.) For your reference, the UTF-8 (and ASCII) codes for the characters `d`, `g`, and `o`, are (in decimal) 100, 103, and 111, respectively.

[additional space for earlier material]

3. (25 pts.) Provide a *complete assembly language program* that corresponds *as closely as possible* to the following C program. (Assume that `ints` are 32 bits wide. The required assembly language program, when run in the RARS(M) environment, must exhibit the same input-output behavior as the given C program.) **Explain your program.** Identify any missing parts or bugs for better partial credit.

```
1  #include <stdio.h>
2  int main() {
3      int a, b, c, r = 10000;
4      scanf("%d%d%d", &a, &b, &c);
5      if (a <= b) {
6          if (b <= c) r += 1;
7          else r += 2;
8      }
9      else {
10         if (a <= c) r += 3;
11         else r += 4;
12     }
13     printf("%d\n", r);
14     return 0;
15 }
```

[additional space for earlier material]