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	2	
3	7	
4	10	
5	20	
total	40	

2. (2 pts.) State the exact output produced by the following C program. (If there is an error of any sort then explain the error instead.) No explanation is needed unless consideration for partial credit is desired.

```
1  #include <stdio.h>
2  int
3  main() {
4    int fnum = 6;
5    fnum += 28;
6    int hnum = fnum << 2;
7    printf("%d %d\n", fnum, hnum);
8    return 0;
9  }
```

3. (7 pts.) Provide an assembly language code fragment that corresponds *as closely as possible* to the code in lines 4, 5, and 6 of the C program of Question 2. Assume that `int`s are 32 bits wide and that `fnum` and `hnum` mapped to two *suitable* registers (which the answer must clearly identify). Explain briefly why your program is correct.

4. (10 pts.) Provide a binary representation of the machine code generated for the following instruction. Explain your answer by indicating how that representation is derived. (Reminders: Textbook/class conventions; no computer use.)

```
addi x22, x22, 42
```

5. (20 pts.) Provide a *complete assembly language program* that corresponds *as closely as possible* to the code following C program. (Assume that `int`s 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. [Hint: 8888 > 2047 is significant.]

```
1  #include <stdio.h>
2  int
3  main() {
4    int i, j, k;
5    scanf("%d%d", &i, &j);
6    k = i + 2 * j + 8888;
7    printf("%d%d%d\n", i, j, k);
8    return 0;
9  }
```

[additional space for earlier material]