DEPARTMENT OF
COMPUTER SCIENCE

COLORADO SCHOOL OF MINES
EARTH • ENERGY • ENVIRONMENT

CSCI 341: Computer Organization
WS 4: Decision Making Instructions

**1** Conditional statements in higher level languages can become very complex, just look at the code below for an example. (Even this example can be considered simple.)

```
if (rv!=0 && self->str!=NULL && other->str!=NULL){
     return strcmp(self->str,other->str);
}
```

RISC-V only allows a comparison between two register values. Knowing this, how would you create a complex conditional statement in assembly? Note that this question is not asking you to translate the code shown above in assembly, it's more asking how something like this would be done in principle.

**2** Turn this C code into RISC-V, generating labels as needed. Map a,b,c,d,e to s0, s1, s2, s3, s4.

```
if (a == b) c = d + e;
else c = d - e;
```

**3** What is the RISC-V for this C code? `i` in `s3`, `k` in `s5`, address of `save` in `s6`.

```
   while (save[i] == k)
           i += 1;
```

4 Write a program that prints "na na na na na na na na Batman!" but only has two calls to print the strings “na “ and “Batman!”.

1. Write it in C/C++/Java/Python first, so you have the high-level design
2. Now, write it in assembly. Hint: you will need to use a branch to loop, and count the number of times you've looped).
3. Finally, how did your solution differ from the one provided in the course shell: batman.s?

```
for (int i=0; i<8; i++) printf(“na “);
printf(“Batman!”);
```