

Name _____

COMP 303

HW #4

1. Convert the following C program to MIPS assembler. Assume that the address of array "a" is in register "r1", and that the variable "sum" is in register "r2."

```
int a[5];
int sum = 0;

sum = a[0] + a[1] + a[2] + a[3] + a[4];
```

2. Take a look at the following program.

```
int a[5], b[5], c[5];
int sum = 0;
for(i = 0; i < 5; ++i) {
    a[i] = 0;
    b[i] = i+1;
    c[i] = 10 * i + 10;
}
/* add all the elements of array b */
for(i = 0; i <= 5; ++i) {
    sum = sum + b[i];
}
```

- a. The program contains a bug. Point out what it is. Hint: check the programmer's intentions, as described in comments, against the code.
- b. What does the buggy program compute for "sum" when it terminates? Assume that the compiler has placed array "a" in memory starting at location 0x100, array "b" in memory starting at location 0x124, and array "c" in memory starting at location 0x138.
- c. Suppose that the compiler has placed array "a" at memory location 0x138, "b" at memory location 0x124, and "c" at memory location 0x100. What does the same buggy program compute for "sum" when it terminates?
- d. Fix the bug and point out what the (now corrected) program computes when it terminates.

3. Convert the following C program to MIPS assembler. Assume that the addresses of arrays "a," "b," and "c" are in registers r1, r2 and r3, respectively, and that the variable "sum" is in register r4.

```
int a[20], b[20], c[20];
c[0] = a[0] + b[0];
c[1] = a[1] + b[1];
c[2] = a[2] + b[2];
c[3] = a[3] + b[3];
c[4] = a[4] + b[4];
```

4. Convert the following C program to MIPS assembler. Assume that the variable "ptr" is in register r4, and use the register r5 for variable "sum."

```
struct Element {
    int value;
    struct Element *next;
};
struct Element *ptr;
int sum = 0;

sum = ptr->value + ptr->next->value +
      ptr->next->next->value;
```