Exam 2 Practice Problems

Problem 1

Create a struct “student” that has a name, age, and gpa. Two students are equal if they have the same name, age, and gpa. Create a function to create a new student, as well as a function that checks to see if two students are equal and the function to free a student. The code you write would then work with the following code snippet.

```
int main() {
  struct student* s1 = create_student("Bob", 19, 3.5);
  struct student* s2 = create_student("Sam", 20, 2.7);
  struct student* s3 = create_student("Bob", 19, 3.5);

  if(student_equal(s1, s2)) {
    printf("Student 1 and Student 2 are equal\n");
  }

  if(student_equal(s1, s3)) {
    printf("Student 1 and Student 3 are equal\n");
  }

  free_student(s1);
  free_student(s2);
  free_student(s3);
}
```

Problem 2

For each analyzer in the compilation process, write an example piece of code that would be caught by that analyzer.

Lexical Analyzer

Syntax Analyzer

Semantic Analyzer

Problem 3

Write a C-preprocessor macro which will give you sum of three numbers.

Problem 4

Make a dependency tree from the following Makefile

```
CC=gcc
CC_FLAGS=-Wall -Wextra -pedantic -std=c99

main: main.o list.o person.o
  $(CC) $(CC_FLAGS) -o main.out main.o list.o person.o
main.o: main.c list.h person.h
  $(CC) $(CC_FLAGS) -c main.c
list.o: list.c list.h
  $(CC) $(CC_FLAGS) -c list.c
person.o: person.c person.h
  $(CC) $(CC_FLAGS) -c person.c
```

Problem 5

Compute the results of the following bitwise operations

```
0011 & 1100
1010 | 1100
1011 ^ 1100
```