1. You coded a set of statistics functions in a file called stats.c. The header file stats.h contains the prototypes of the functions.

   Which of the following commands will produce an object file

   a. gcc stats.c
   b. gcc –c stats.c
   c. gcc –g stats.c
   d. gcc –o stats.c
   e. gcc –o stats.c stats.h

2. The program printMain prints the following "This is a test!!". Given the following code, how many times is the sentence "This is a test!!" printed?

```
#include "stdio.h"
#include "uinstd.h"

int main()
{
    int cpid = 0;
    char *args[2]={"printMain", NULL};

    cpid = fork();
    execv(",/printMain", args);
    cpid = fork();
    execv("./printMain", args);
    sleep(10);
    return(0);
}
```

   a. 1
   b. 2
   c. 3
   d. 4
   e. none of the above

3. Given the following code, how many times is the sentence "This is a test!!" printed? Explain why.

```
#include "stdio.h"
#include "unistd.h"

int main()
{
   int cpid = 0;
```

```
    int i;
    for(i = 1; i< 3; i++) {
       cpid = fork();
       cpid = fork();
       cpid = fork();
    }
    printf("This is a test \n");
    return(0);
}
```

4. Given the following code. Assume that each the process id of the main program is 100 and that each child id is one thahas an id of What will be printed?

```
#include "stdio.h"
#include "unistd.h"

int main()
{
   int cpid = 0;
   int status;

   printf("AA = %d\n",getpid());
   cpid = fork();
   if (cpid != 0) {
      wait(&status);
      printf("DD = %d\n",getpid());
   }

   cpid = fork();
   if (cpid != 0) {

      wait(&status);
      printf("EE = %d\n",getpid());
   }

   printf("FF = %d\n",getpid());

   return(0);
}
```

Answers:

1. What is the purpose of the preprocessing step?
2. What is the purpose of the ***make*** program?
3. What is the structure of a ***makefile*** that is used by the ***make*** program?
4. True/false
    a. An #include directive can appear anywhere in a code or header file?

b. A #define directive can appear anywhere in a code or header file
c. If the same #define directive (with the same name e.g, MAX(x,y))appears multiple times in the same code file then the file compilation will fail
d. An include file (e.g., myHeader.h) can contain only data type declarations and functions prototypes
e. A parent process can send information to a child process (e.g., number of second to sleep) using the kill() function
f. A function that is registered with the signal() function can be activated by the operating system as necessary and in doing so disrupts the normal execution of the program.

5. The file employee.h contains the following

```
struct employee {
    unsigned int age;
    float salary;
};
```

The file person.h contains the following

```
#include "employee.h"
struct person {
    char name[64];
    struct employee emp;
};
```

The file main.c contains the following

```
include "stdio.h"
#include "person.h"
#include "employee.h"

int main()
{
    struct person p;
    printf("in main \n");
}
```

You are compiling the file main,.c and there is a compiling error. What is the error? Why did it occur? How can you fix it? Note, that the problem is not a syntax error such as a missing “;”.

Coding questions:

1. Write a main program using a manager- workers paradigm that creates where the main thread is the manger and the children are the workers. The number of workers should be three. Each worker will call a function workerFunction() that was coded by another programmer in the company.

   The manager needs to wait until all workers completed the work.

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>

int workerFun();

int main()
{
  int workers[3];









}
```

2. The following program raises a floating point exception (SIGFPE). Why is it raised? Add code so that the program will trap the SIGFPE, print the following message "The program

```
#include <stdio.h>
#include <unistd.h>
#include <signal.h>




int main()
{
   int i, j = 3;

   for (i = 7; i > 0; i--) {
      printf("%d %% %d  = %d \n",i, i-j, i % (i-j));
   }
   return(0);
}
```

}