1. Consider this simple C program:

```c
int fibonacci(int *n2, int *n1, int x)
{
    /* printf("fib(%d,%d,%d)\n", *n2, *n1, x); */
    int n0 = *n1 + *n2;
    *n2 = *n1;
    *n1 = n0;
    if (x <= 3) return n0;
    else return fibonacci(n2, n1, --x);
}

int main()
{
    int n1 = 1;
    int n2 = 1;
    printf("%d\n", fibonacci(&n2, &n1, 4));
}
```

The Intel x86 assembly language for this program can be found at http://www.cs.purdue.edu/homes/brylow/cs352/Fall2003/Homework/hw2-p1.s

Draw the stack of activation records for this program, showing the final contents of each location when the stack is at its largest point. Label each word in the stack that you can identify. Some locations in the stack may be undefined or unlabeled.

The first three words of the stack will look like this:

```
?  MAIN stack frame
|----|
|  | RET_OS Return address of Operating System
| | BP_OS Frame Pointer in MAIN
|  | ...
|  | Local variable "n1"
|  | ...
```

Fill in the rest.
2. The source code from problem 1 can be downloaded from
http://www.cs.purdue.edu/homes/brylow/cs352/Fall2003/Homework/hw2-p1.c
Run the Mips compiler provided in the ∼cs352/bin directory to generate the Mips
assembly language for hw2-p1.c.

mips-gcc -S hw2-p1.c

Draw the stack of activation records for this program, showing the final contents of
each location when the stack is at its largest point. Label each word in the stack that
you can identify. Some locations in the stack may be undefined or unlabeled.
The first three words of the stack will look like this:

```
?    MAIN stack frame
RET_OS Return address of Operating System
FP_OS Frame Pointer in MAIN
... Local variable "n2"
```

Fill in the rest.

3. (a) By convention, the Mips machine prefers to pass function arguments in designated
argument registers. Why does the program in question 2 store the values of
fibonacci’s arguments on the stack during execution?

(b) Give an example of a C program that requires the Mips machine to spill function
arguments into the stack frame. Verify your example with the Mips compiler.

(c) Each Mips routine begins with a line like, “subu $sp,$sp,x”, where x is the size
of the stack frame to allocate. What three factors contribute to the size of x?