#1. In C, a(n) _________________  _________________  _________________ variable has local/function scope and has memory allocated in the Data segment of the run time environment.

A(n) _________________ variable has local/function scope and has memory allocated in the Stack (stack frame) of the run time environment.

The lifetime of any program part in the BSS is _________________.

The _________________ is responsible for ensuring the BSS segment in memory is zero-filled before the program starts executing.

_______________ is the keyword to denote a variable/function declaration. The real definition is expected to be found in another source module.

_______________ is the keyword in most modern languages (C/C++/Java/C#) that turns off any type of optimization.

List the 2 main functions of the linkage editor:

1)

2)

How many bits are available in a SPARC Format 2 instruction to encode a branch displacement?

Which type of cache is more efficient for programs that may perform many writes to the same temporary short-lived variables?

#2. What gets printed if the following function is invoked as `recurse(3, 2)`?

```c
#include <stdio.h>

int recurse( int a, int b ) {
    int local = a - b;
    int result;
    if ( a < 8 )
    {
        result = local + recurse( a + 2, b );
    }
    else
    {
        result = local;
    }
    printf( "%d\n", result );
    return result;
}
```

(over)
#3. Given the following program, reorder the printf lines so that the values that are printed are sorted from smallest to largest if compiled and run on a Sun SPARC architecture. These lines print out the address of the different parts of the program (not the values assigned) with the printf() format specifier %p.

```c
int a;

int main( int argc, char *argv[] ) {
    int b = 420;
    static int c = 404;

    /* 1 */ (void) printf( b --> %08p
", &b );
    /* 2 */ (void) printf( c --> %08p
", &c );
    /* 3 */ (void) printf( a --> %08p
", &a );
    /* 4 */ (void) printf( main --> %08p
", main );
    /* 5 */ (void) printf( argc --> %08p
", &argc );
    /* 6 */ (void) printf( malloc --> %08p
", malloc(50) );
}
```

This line number would print the smallest value

This line number would print the largest value

What question would you like to see on the Final Exam? (1 pt)