PROBLEM 5

*extern* – specifies external linkage. The object or function is actually defined in another file. Declarations of variables and functions at file scope are external by default. For example in a file called test.c there is an int x = 10; within the main() function, x can be redeclared using extern int x. The extern keyword can be left out in this case, as it is implied that x is external.

*const* – specifies that an object cannot be modified after it is defined. An example of a *const* is the number of days in a year, it will never change, and should therefore never be modified.

const int year = 365;

*register* – tells the compiler that the object will be used very frequently and should be kept in a processor register if possible.

register int i = 0;

while(1) ++i;

*static –* inside a function, static keeps the object value between function calls, it “remembers”. A static variable or function is only seen in the file it is defined in.

test(){

static x = 0;

printf(“%d”, ++i);

}

calling test() the first time will output 1, the second call will output 2, etc.

*continue* – passes control to the next iteration of a do, for, or while loop;

for (i=0; i< 5; ++i){

if (i==1)

continue;

printf(“%d”,i);

}

this code will output the values: 0,2,3,4 (it will skip 1)