

Closed book, closed notes.

1. (10 points) Given the following makefile, called makefile. After I edit one.h, when I type make, which files will be changed by gcc?

```
whole.out : one.o, two.o, three.o          two.o, one.o, and whole.out
    gcc -o whole.out one.o two.o three.o

one.o : one.c whole.h one.h
    gcc -c one.c

two.o : two.c whole.h one.h
    gcc -c two.c

three.o : three.c whole.h three.h
    gcc -c three.c
```

2. (20 points)

```
char *strcat(char *s1, const char *s2)
{
    char * cat = malloc ( (strlen(s1)+strlen(s2))*sizeof(char) ); // reserve memory
    char * dummy = cat; // point dummy at the same location as cat

    while ( *cat++ = *s1++ ); // copy s1 into cat, including '\0'
    cat--; // get rid of '\0'
    while ( *cat++ = *s2++ ); // copy s2 into cat, starting at the end of s1

    return dummy;
}
```

3. (15 points) Consider the following recursive function. What does the function do as whole? (Do not describe what each line does.)

```
void pb(int n) {
    if (n != 0) {
        pb(n / 2);
        putchar('0' + n % 2);
    }
}
```

It prints out the binary representation of n starting with the leftmost 1.
For example, if n = 43 then pb would print 101011.

4. (10 points) What will be the values of the strings s1 and s2 after the following statements have been executed?

```
strcpy(s1, "computer");
strcpy(s2, "science");
if (strcmp(s1, s2) < 0)
    strcat(s1, s2);
else
    strcat(s2, s1);
s2[strlen(s2) - 6] = '\0';
```

s1 computerscience

s2 s

5. (30 points) Show what the following program will look like after preprocessing. Some lines of the program may cause compilation errors; underline all such lines.

#define N = 10	
#define INC(x) x + 1	
#define SUB (x, y) x - y	
#define SQR(x) ((x) * (x))	
#define CUBE(x) (SQR(x) * (x))	
main() {	main() {
int a[N], i, j, k, m;	<u>int a[= 10], i, j, k, m; /* = 10 is an error */</u>
#ifdef N	
i = j;	i = j;
#else	
j = i;	
#endif	
i = 10 * INC(j);	i = 10 * j + 1 ;
i = SUB(j, k);	<u>i = (x, y) x - y (j, k); /* x and y undeclared */</u>
i = SQR(SQR(j++));	i = ((((j++) * (j++))) * ((j++) * (j++))) ;
i = CUBE(j);	i = (((j) * (j)) * (j)) ;
#undef SQR	
i = SQR(j);	<u>i = SQR(j); /* SQR undefined */</u>
#define SQR	
i = SQR(j);	i = (j);
return 0;	return 0;
}	}

6. (68 points) Write a program that counts how many times each of the 26 letters is used in the file test.txt and then prints out the result. Your program must contain two functions besides main: 1) count_letters reads from the file and counts the number of occurrences of each letter, and 2) show_results prints to the screen the results of the counting.

If test.txt is:

There has been an alarming increase
in the number of things I know nothing about.

The output would be:

a b c d e f g h i j k l m n o p q r s t u v w x y z
6 3 1 0 8 1 3 5 6 0 1 1 2 10 4 0 0 4 3 5 2 0 1 0 0 0

Pts	
1	#include <stdio.h>
1	#include <ctype.h>
4	void count_letters(int count[])
	{
	char c;
2	FILE *fp;
	int i;
4	for (i = 0; i < 26; i++)
2	count[i] = 0;
4	fp = fopen("test.txt", "r");
6	while(fscanf(fp,"%c", &c) != EOF)
2	if (isalpha(c))
6	++count[tolower(c) - 'a'];
	fclose(fp);
	}
4	void show_results(int count[])

	{
	int i;
5	for(i = 'a'; i <= 'z'; i++)
2	printf(" %c ", i);
1	printf("\n");
5	for(i = 0; i < 26; i++)
6	printf("%2d ", count[i]);
1	printf("\n");
	}
2	int main()
	{
2	int count[26];
2	count_letters(count);
2	show_results(count);
2	return 0;
	}

