

The questions in this document may have appeared in midterms in previous years when this class was taught by different instructors.

1. (10 points) You are 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
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) Write a function `char * strcat (char *, char *)`, which takes as input two strings and returns a string which is the concatenation of the two. For full points you must use pointers and not arrays (solutions with arrays will get 50% deduction of points).

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);
    }
}
```

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_____

s2_____

5. (30 points) On the lines at the right, 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() {
```

```
    int a[N], i, j, k, m;
```

```
#ifdef N
```

```
    i = j;
```

```
#else
```

```
    j = i;
```

```
#endif
```

```
i = 10 * INC(j);
```

```
i = SUB(j, k);
```

```
i = SQR(SQR(j++));
```

```
i = CUBE(j);
```

```
#undef SQR
```

```
    i = SQR(j);
```

```
#define SQR
```

```
    i = SQR(j);
```

```
    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. Do not distinguish between lower and uppercase letters.

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