*Midterm Exam*

csci-e113 17 November 1997

Your Name:

*Instructions:* You have two hours for this exam. You may use any notes or texts you like. Please put your answers in the spaces provided on this test sheet. The point values of the questions are listed in the table below. Watch the time, write clearly, good luck.

| *#* | *points* | *score* | *totals* |
|---|---|---|---|
| 1 | 4 | | |
| 24 | | | |
| 34 | | | |
| 44 | | | |
| 54 | | | |
| 64 | | | |
| | | | |
| 75 | | | |
| 85 | | | |
| 95 | | | |
| | | | |
| 10 | 12 | | |
| 11 | 6 | | |
| | | | |
| 12A | 9 | | |
| 12B | 9 | | |
| 12C 9 | | | |
| | | | |
| 13A | 8 | | |
| 13B | 8 | | |
| | | | |

*Problems 1-6: Compare and Contrast.* For each of the following pairs of six fragments, describe the difference in value, effect, and/or utility. Please be concise and stay in the space provided.

1. A.
```
break;
```
B.
```
exit(0);
```

2. A.
```
char *p1, *p2;
p1 = p2 ;
```
B.
```
char *p1, *p2;
strcpy(p1, p2);
```

3. A.
```
static int i;
```
B.
```
extern int i;
```

4. A.
```
char table[12][20];
```
B.
```
char *table[12];
```

5. A.
```
while( c = *p++ )
    putchar(c);
```
B.
```
while( c == *p++ )
    putchar(c);
```

6. A.
```
int n;
f(n);
```
B.
```
int n;
f(&n);
```

*Problems 7-9: Syntax/Usage.* Something is wrong with each of the following C fragments. State the problem AND correct the error.

7.

```
char *s = "xyz";
while( s != NULL ){
      putchar(*s);
      s++;
}
```

8.

```
if ( x >= MIN || x <= MAX )
     printf("Number is in range\n");
```

9.

```
newnode = (struct node *) malloc( sizeof(struct node) );
newnode.next = NULL;
```

*Problem 10: Arrays and Pointers:* Given the following definitions and initial assignments,

```
int list[] = { 4, 7, 3, 6, 5, 9 };
int *p, *q;
p = list; q = &list[3];
```

state the value of each of the expressions a) - d):

a) `p[1] = ___` b) `q[1] = ___` c) `*q - *p = ___` d) `q - p = ___`

e) what is the *type* of the expression `list + 2`? _____

*Problem 11: Storage Classes:* The following program shows only the function and variable definitions. The code is not shown.

```
int x, y;

void f(int y)
{
   int c;
}

void g(void)
{
   int d;
}

main()
{
   int x;
}
```

What variables and/or parameters are visible in each of these functions? (Some names appear in more than one place; be sure you identify the variable clearly.)

a) `main`

b) `f`

c) `g`

*Problem 12: Arrays and Loops* Concerned about this global warming business, you decide to investigate for yourself. You get an electronic thermometer, put it outside your window and attach the cable to your computer. Each hour your computer records the temperature and stores that integer value in an array defined as: `int temp[7][24]`. After seven days, you have a record of the hour-by-hour temperature. You are now prepared to analyze the data.

**12A** Write a function called `avg_temp(int data[7][24], int day)` that returns as a floating point value the average temperature on the day specified by the argument called 'day', an integer in the range 0..6.

**12B** Write a function called `scorcher(int data[7][24])` that returns the day number (an integer in the range 0..6) of the day with the greatest average temperature. You must call the function described in part **A** above. If there is more than one day with the greatest average temperature, return the day number of the first one you find.

**12C** Write a function called `freezing(int data[7][24])` that prints to standard output the day and hours of all times when the temperature was 32 degrees or lower. The output should look like:

```
Monday 2
Monday 3
Monday 4
Monday 5
Wednesday 15
Wednesday 16
Wednesday 17
 ...
```

Day 0 is Monday.

*Problem 13: Linked Lists* Consider a linked list like the one we used in `wlfiler4.c` to store words and their frequencies. Each node contains a pointer to a string, a count, and also contains a pointer to the next node in the list. Here are two functions that might be useful in analyzing or processing the data in this list.

**13-A** Write a function called `word_position(struct word *head, char str[])` that takes as arguments a pointer to the head of the list and a string. This function returns the position in the list of the link containing `str`. The first item in the list is item number 0, the next one is item number 1, etc. If the word does not appear in the list, this function returns -1.

**13-B** Write a function called `swap_top2(struct word *head)` that takes as input the pointer to the head node of the list exchanges the first two element in the list. Assume the list has at least two elements. Here are 'before' and 'after' pictures: