Student Name:... NID:...........................

COP 3223 Sec 2: Spr'09 C Programming Test 2 Practice (50 points)

1. (15 points) Write down what the printed output of this program is. Assume input is aaBcc*cC7

```
#include <stdio.h>
#include <ctype.h>
int main() {
  int index, freq[26], c, stars, maxfreq;
  for (index=0; index<26; index++)
    freq[index] = 0;
  while ( (c = getchar()) != '7') {
    if (isalpha(c))
      freq[tolower(c)-'a']++;
  }
  maxfreq = freq[25];
  for (index = 24; index >= 0; index--) {
    if (freq[index] > maxfreq)
      maxfreq = freq[index];
  }
  printf("a b c d e f\n");
  for (index=0; index<5; index++) {
    for (stars=0; stars< (maxfreq - freq[index]); stars++)
       printf(" ");
    for (stars=0; stars< (freq[index]); stars++)
       printf("*");
    printf("%c  \n", ('A' + index) );
    }
    printf(" \n");
  }
  return 0;
}
```

```
          THIS OUTPUT WILL BE GRADED
           -------------------------------------------------------
           |
Out Line 1 |
           |
           |
Out Line 2 |
           |
           |
Out Line 3 |
           |
           |
Out Line 4 |
           |
           |
Out Line 5 |
           |
           |
Out Line 6 |
```

2. (15 points) Write down what the printed output of this program is:

```
#include <stdio.h>

int main()

{
        int a[6] = {0, 3, 5, 7, 4, -1};
        int b[6] = {1, 10, 20, 40, 100, 2000};
        int i,x,temp;

        for(i = 0; i < 6; i++)
                printf("%d ", a[i]);
        printf("\n");

        for(i = 1; i < 6; i++)
        {
                a[i] += a[i-1];
        }

        for(i = 0; i < 6; i++)
                printf("%d ", a[i]);
        printf("\n");

        printf("Result is: %f\n", a[5] / 6.);

        x = 5;

        for(i = 0; i < x; i++)
        {
                temp = b[x];
                b[x] = b[i];
                b[i] = temp;
                x--;
        }

        for(i = 0; i < 6; i++)
                printf("%d ", b[i]);
        printf("\n");

        return 0;
}
```

3. (8 points) Write a C program that will read in two integers. The program should first add the two integers, then multiply that sum by 40, and then print out the answer. Assume correct input.

4. (12 points) Write a C program that uses a FOR-loop to read in 9 numbers. For each of the 9 numbers, if it is greater than 20, it should be added into a sum. After the loop, print out the sum. Assume correct input.

5. (12 points) Write a C program that will read in integers until it encounters the integer –99. Note that –99 could be the first integer. For each other (earlier) integer, if present, you need to compute its

squared value, and then print out the integer followed by its squared value, one pair per line with two blanks between the pair.. Assume the other integers (besides the –99) are positive, and do not print a line for –99.

6. (15 points) Write a program that will read in and decide if a given integer is prime (a prime number is divisible by only 1 and itself).

7. (20 points)
   Write a C program that asks the user to enter one float at a time. As long as the user continues to enter descending floats (the newest float is less than the previous one entered) the program should report the product of the current and previous number (except for the first float entered). If the user enters a number that is not smaller than their last response, the program should tell them so and exit. You can assume that for each run of the program, the user will enter at least two floats, and will not try to trick the program; so, you do not need to check for errors in what the user types in.

8. (12 points) (This is from the textbook) Write a program that prompts the user to enter an integer n, then prints all even squares between 1 and n. For example, if the user enters 100, the program should print: 4 16 36 64 100

9. (20 points) (This is from the textbook) Write a program that prints a one-month calendar. The user specifies the number of days in the month and the day of the week on which the month begins:

```
Enter number of days in month: 31
Enter starting day of the week (1=Sun, 7=Sat): 3
The month calendar is below:
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

   Hint: The most important part of the solution is a *for* statement that uses a variable i to count from 1 to n, where n is the number of days in the month, printing each value of i. Inside the loop, an *if* statement tests whether i is the last day in a week; if so, it prints a new line character.