1. (5 points) **Fill in the blank:**
To use printf or scanf, you must include the header file ________________.

2. (5 points) **TRUE/FALSE**
When compiling nested IF ELSE statements, the compiler uses the indentation to match each ELSE with an IF.

**Circle one:**

TRUE    FALSE
3. In question 3, there are no deliberate syntax/typographical errors. Please mark your final answer clearly. In order to receive partial credit, you must show your work.

a) (8 points) Show the output and describe the behavior of the following statements for each of the cases when x initially equals 0, 1 and –1:

```c
if ( x == 0 )
    printf(“A
”);
else if (x != 1)
    printf(“B
”);
else if (x < 1)
    printf(“C
”);
else if (x = -1)
    printf(“D
”);
else if (x)
    printf(“E
”);
```

b) (8 points) Show the output and describe the behavior of the following statements for each of the cases when x initially equals 0, 1 and –1:

```c
int y;
for  (y=x; y>=0; y=x+1)
{
    printf("%d ",y);
    x--;
}
```

c) (8 points) Show the output and describe the behavior of the following statements for each of the cases when x initially equals 0, 1 and –1:

```c
do
{
    y=1;
    while ( ( y > 0 ) && ( x <= 0 ) )
    {
        printf("x is equal to %d\n",x);
        y--;  
    }
    x++;  
} while ( x < 1 );
```
4. (20 points) Design a program that will help Max the Mathematician manipulate numbers. He needs a program to perform the following tasks. First, ask a user for a positive integer input, say n. Then, sum up all positive integers up to and including the user input, and tell the user the answer; i.e., $1 + 2 + \ldots + n$. Next, compute the sum of the squares of these integers and tell the user the answer; i.e., $1^2 + 2^2 + \ldots + n^2$. Finally, show the user the factorial of each integer up to and including the user input; i.e., $1!, 2!, \ldots n!$.

Perform the first three steps of the software design process, up to and including defining the algorithm(s). No code is required for this problem. Remember, modularity is a virtue.
5. (16 points) What is the output of this program? No partial credit will be awarded without an explanation of your thought process. A trace would be acceptable.

#include <stdio.h>

int main(void)
{
    int y = 0;
    int z = 0;

    int total = 0;
    int iterated = 0;

    for (y=3; y >= 1; y--)
    {
        for (z=2; z < y*2; z = z+2)
        {
            total = total + 2;
        } /* end inner for */
        iterated++;
    } /* end outer for */

    printf("The final value of iterated is:\n%d\n", iterated);

} /* end main */
6. (15 points) Write a function that produces the following output:

*  
**  
***  
...

A passed parameter defines the number of lines to print. For example, if the parameter is 4, the output will look like this:

*  
**  
***  
****

The function's prototype is

```c
void draw_stars (int num);
```
7. Given the code:

```c
#include <stdio.h>

int aa = 3;
int f(int dd)
{
    aa=5;
    return aa+dd;
}

int main(void)
{
    int aa=6, bb=5, cc=4, dd;
    cc = aa + bb;
    {
        int cc=3, dd=2;
        dd += aa + cc;
        aa %= dd;
    }
    dd = f(3);
    aa = cc + bb;
    printf("aa = %d\n",aa);
    return 0;
}
```

a) (5 points) What is the value of `aa` printed when line 24 executes? Recall that `x += 1` translates into `x = x + 1`, and `x %= 3` translates into `x = x % 3`.

b) (10 points) On each of the four lines whose numbers are given below, a variable is changed. For each line indicate:
   - which variable is being changed
   - on which line that variable is declared
   - for which lines the variable is in scope
   - for which lines the variable is visible
   - for which lines the variable is invisible
   - the value that is assigned to the variable:

<table>
<thead>
<tr>
<th>Line</th>
<th>Variable changed</th>
<th>Line where DECLARED</th>
<th>Lines where in SCOPE</th>
<th>Lines where VISIBLE</th>
<th>Lines where INVISIBLE</th>
<th>assigned VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>