Write a Fortran and a C program to calculate and to produce a table in an output file which contains 5 columns of data as follows:

<table>
<thead>
<tr>
<th>x</th>
<th>x**(-2.3)</th>
<th>x**(-0.5)</th>
<th>x**0.8</th>
<th>x**3.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>.20</td>
<td>40.51641</td>
<td>2.23607</td>
<td>.27595</td>
<td>.00259</td>
</tr>
<tr>
<td>.40</td>
<td>8.22739</td>
<td>1.58114</td>
<td>.48045</td>
<td>.03370</td>
</tr>
<tr>
<td>.60</td>
<td>3.23782</td>
<td>1.29099</td>
<td>.66454</td>
<td>.15106</td>
</tr>
<tr>
<td>.80</td>
<td>1.67068</td>
<td>1.11803</td>
<td>.83651</td>
<td>.43796</td>
</tr>
<tr>
<td>1.00</td>
<td>1.00000</td>
<td>1.00000</td>
<td>1.00000</td>
<td>1.00000</td>
</tr>
<tr>
<td>1.20</td>
<td>.65748</td>
<td>.91287</td>
<td>1.15703</td>
<td>1.96323</td>
</tr>
<tr>
<td>1.40</td>
<td>.46122</td>
<td>.84515</td>
<td>1.30889</td>
<td>3.47275</td>
</tr>
<tr>
<td>1.60</td>
<td>.33925</td>
<td>.79057</td>
<td>1.45645</td>
<td>5.69173</td>
</tr>
<tr>
<td>1.80</td>
<td>.25875</td>
<td>.74536</td>
<td>1.60036</td>
<td>8.80052</td>
</tr>
<tr>
<td>2.00</td>
<td>.20306</td>
<td>.70711</td>
<td>1.74110</td>
<td>12.99604</td>
</tr>
<tr>
<td>2.20</td>
<td>.16309</td>
<td>.67420</td>
<td>1.87905</td>
<td>18.49115</td>
</tr>
<tr>
<td>2.40</td>
<td>.13351</td>
<td>.64550</td>
<td>2.01451</td>
<td>25.51418</td>
</tr>
<tr>
<td>2.60</td>
<td>.11106</td>
<td>.62017</td>
<td>2.14773</td>
<td>34.30848</td>
</tr>
<tr>
<td>2.80</td>
<td>.09366</td>
<td>.59761</td>
<td>2.27891</td>
<td>45.13202</td>
</tr>
<tr>
<td>3.00</td>
<td>.07991</td>
<td>.57735</td>
<td>2.40822</td>
<td>58.25707</td>
</tr>
</tbody>
</table>

The independent variable $x$ in the first column is varied from 0.20 to 3.00 at an increment of 0.20, the other columns are to be calculated without the fortran operator ** or the C function "pow" or any of the provided “Intrinsic” functions by both languages, e.g., exp, alog, pow, etc. To perform the calculations, you would need to write some functions as follows:

```fortran
real function power(x,a)
  real x,a
  power=x**a
  return
end
```

```fortran
real function natlog(x)
  real x
  natlog=-log(1-x)
  return
end
```

```fortran
real function e(x)
  real x
  e=exp(x)
  return
end
```

The function `power(x,a)` calculates $x^a$ with $a$ being a floating point number, it utilizes the function `natlog` and `e`. The function `natlog(x)` calculates the natural logarithm, $\log_e x$ using the mathematical series

$$
\log_e(1-y) = -\left[ y + \frac{y^2}{2} + \frac{y^3}{3} + \frac{y^4}{4} + \frac{y^5}{5} + \ldots \right], \quad [y^2 < 1 \text{ and } y = -1]
$$
The function \( e(x) \) calculates \( e^x \), or the anti-log to \( \log_e x \), using the mathematical series

\[
e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \ldots
\]

The \texttt{natlog} and \texttt{e} functions are needed because

\[
x^a = e^{a \log_e x}
\]

• The computer program should open an output file named “table.dat” to store the results. The output data should be written with several well designed format statement.

• Create a UNIX script file with the programs and the sample executions in it and send it through e-mail to ce108@usc.edu.

• If you do the program only in fortran, your grade will be 80% of the maximum.

To do the C program, the definition of the functions are

```c
#include <stdio.h>
#include <math.h>
float power(float x, float a);
float natlog(float x);
float e(float x);

main()
{
    ......
}

float power(float x, float a)
{
    float answer;
    ......
    return answer;
}

float natlog(float x)
{
    float answer;
    ......
    return answer;
}

float e(float x)
{
    float answer;
    ......
    return answer;
}
```