**Blue Waters Petascale Semester Curriculum v1.0**

**Unit 4: OpenMP**

**Lesson 1: Race Conditions**

**Sample Assessment**

*Developed by David P. Bunde for the Shodor Education Foundation, Inc.*



*Except where otherwise noted, this work by The Shodor Education Foundation, Inc. is licensed under CC BY-SA 4.0. To view a copy of this license, visit*[*https://creativecommons.org/licenses/by-sa/4.0*](https://creativecommons.org/licenses/by-sa/4.0)

*Browse and search the full curriculum at*[*http://shodor.org/petascale/materials/semester-curriculum*](http://shodor.org/petascale/materials/semester-curriculum)

*We welcome your improvements! You can submit your proposed changes to this material and the rest of the curriculum in our GitHub repository at*[*https://github.com/shodor-education/petascale-semester-curriculum*](https://github.com/shodor-education/petascale-semester-curriculum)

*We want to hear from you! Please let us know your experiences using this material by sending email to* [*petascale@shodor.org*](mailto:petascale@shodor.org)

1. Define the term “race condition” and give an example of where one occurs outside the computer in a domain other than cooking.
2. A *power series* is one way to compute (or estimate) the value of a function as the sum of an infinite sequence of terms. For example, the value of ex can be computed as

* 1. Show how to use OpenMP to parallelize the following code for computing terms of this into the array t[i] assuming that the powers of x and the factorials are in the arrays xs[i] and f[i] respectively.

for(int i=0; i < k; i++) {

t[i] = xs[i] / f[i];

}

* 1. Write OpenMP code to calculate the sum of the first k terms of this series assuming that the t[i] values have already been computed as in the previous part.

1. The code fragment below worked in serial, but not once the OpenMP pragma was added. Explain the problem and show how to fix it.

#pragma omp parallel for

for(int i=0; i < n; i++) {

cnt = 0;

val = i;

while(val > 1) {

if(val % 2 == 0)

val = val / 2;

else

val = 3\*val + 1;

cnt++;

}

ans[i] = cnt;

}