Yakshita B Rakholiya
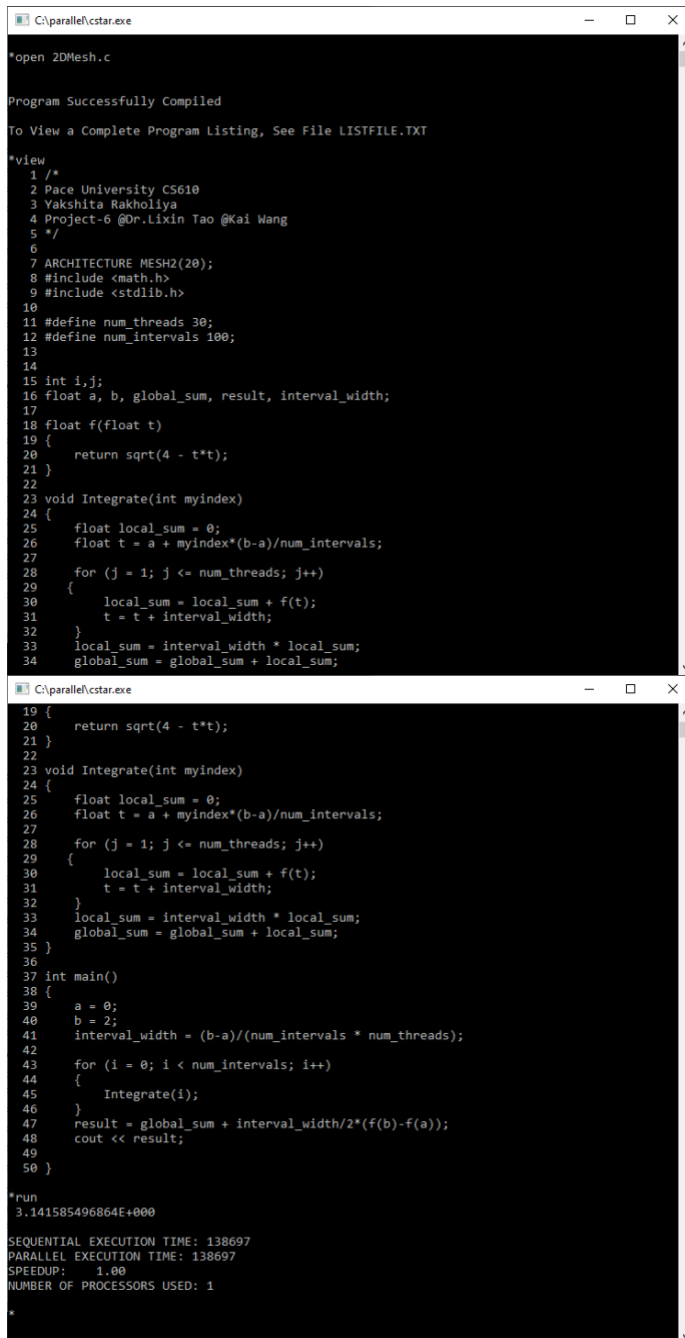**yr92282n@pace.edu**
Student ID: U01875270
Course: CS-610-22756
Project-6

o Developing an efficient parallel numerical integration program on a 2-D mesh, as described in textbook Chapter 8 Programming Projects, page 302.

```
C:\parallel\cstar.exe                                        —   □   ×

*open 2DMesh.c

Program Successfully Compiled

To View a Complete Program Listing, See File LISTFILE.TXT

*view
   1 /*
   2 Pace University CS610
   3 Yakshita Rakholiya
   4 Project-6 @Dr.Lixin Tao @Kai Wang
   5 */
   6
   7 ARCHITECTURE MESH2(20);
   8 #include <math.h>
   9 #include <stdlib.h>
  10
  11 #define num_threads 30;
  12 #define num_intervals 100;
  13
  14
  15 int i,j;
  16 float a, b, global_sum, result, interval_width;
  17
  18 float f(float t)
  19 {
  20     return sqrt(4 - t*t);
  21 }
  22
  23 void Integrate(int myindex)
  24 {
  25     float local_sum = 0;
  26     float t = a + myindex*(b-a)/num_intervals;
  27
  28     for (j = 1; j <= num_threads; j++)
  29     {
  30         local_sum = local_sum + f(t);
  31         t = t + interval_width;
  32     }
  33     local_sum = interval_width * local_sum;
  34     global_sum = global_sum + local_sum;
```

```
C:\parallel\cstar.exe                                        —   □   ×

  19 {
  20     return sqrt(4 - t*t);
  21 }
  22
  23 void Integrate(int myindex)
  24 {
  25     float local_sum = 0;
  26     float t = a + myindex*(b-a)/num_intervals;
  27
  28     for (j = 1; j <= num_threads; j++)
  29     {
  30         local_sum = local_sum + f(t);
  31         t = t + interval_width;
  32     }
  33     local_sum = interval_width * local_sum;
  34     global_sum = global_sum + local_sum;
  35 }
  36
  37 int main()
  38 {
  39     a = 0;
  40     b = 2;
  41     interval_width = (b-a)/(num_intervals * num_threads);
  42
  43     for (i = 0; i < num_intervals; i++)
  44     {
  45         Integrate(i);
  46     }
  47     result = global_sum + interval_width/2*(f(b)-f(a));
  48     cout << result;
  49
  50 }

*run
 3.141585496864E+000

SEQUENTIAL EXECUTION TIME: 138697
PARALLEL EXECUTION TIME: 138697
SPEEDUP:    1.00
NUMBER OF PROCESSORS USED: 1

*
```