VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CSE4001 Parallel and Distributed Computing

Digital Assignment-2 (ELA)

Name: Nakul Jadeja

Reg no.: 19BCE0660

School of Computer Science and Engineering

Course Code: CSE4001

Slot: L11+L12

Winter Semester 2021-22

Professor: Narayanamoorthi M

Question 1)

Write an open MP program using C for the following 1-D array of n-elements and manipulate the following.

(Find the total time taken for each)

AIM: -

-To initialize and print the array elements

-To add / multiply a value to each element and print the same

-To find the sum / odd sum / even sum of the array elements:

-To count the odd and even elements in the array

-To find the sum of square's and cubes of array elements

-To find the maximum and minimum in an array

-To print and count the prime numbers and their sum in the array

-To find the mean / standard deviation and variance

CODE: -

```
#include<stdio.h>
#include<omp.h>
#include<math.h>
#define N 20
int main()
{
int A[20]= {
10,122,124,178,155,255,489,687,355,148,100,144,178,9,4,7,5,6,35,22 };
int i;
double start, end, total;
omp_set_num_threads(10);
#pragma omp parallel private(A)
```

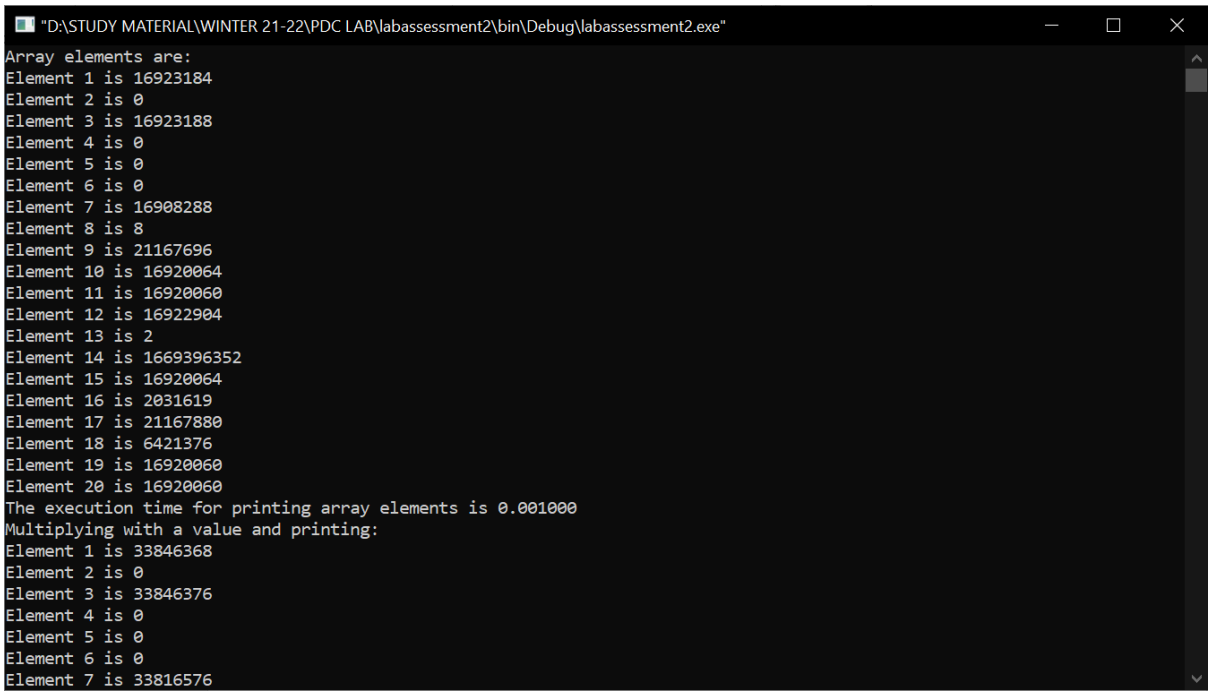
```
{
 start = omp_get_wtime();
 printf("Array elements are:\n");
 for (i = 0; i < N; i++)
{
printf("Element %d is %d\t\n", i+1,A[i]);
}
end = omp_get_wtime(); total =
end - start;
printf("The execution time for printing array elements is %f",total);
start = omp_get_wtime();
printf("\n");
printf("Multiplying with a value and printing:\n");
for (i = 0; i < N; i++)
{
printf("Element %d is %d\t\n", i+1, (2*A[i]));
}
end = omp_get_wtime();
total = end - start;
printf("The execution time for multipluing and printing array elements is
%f\n",total);
start = omp_get_wtime();
int evensum=0, oddsum=0, totalsum=0;
for (int i = 0; i < N; i++)
{
if (A[i] % 2 == 0)
{
evensum += A[i];
}
```

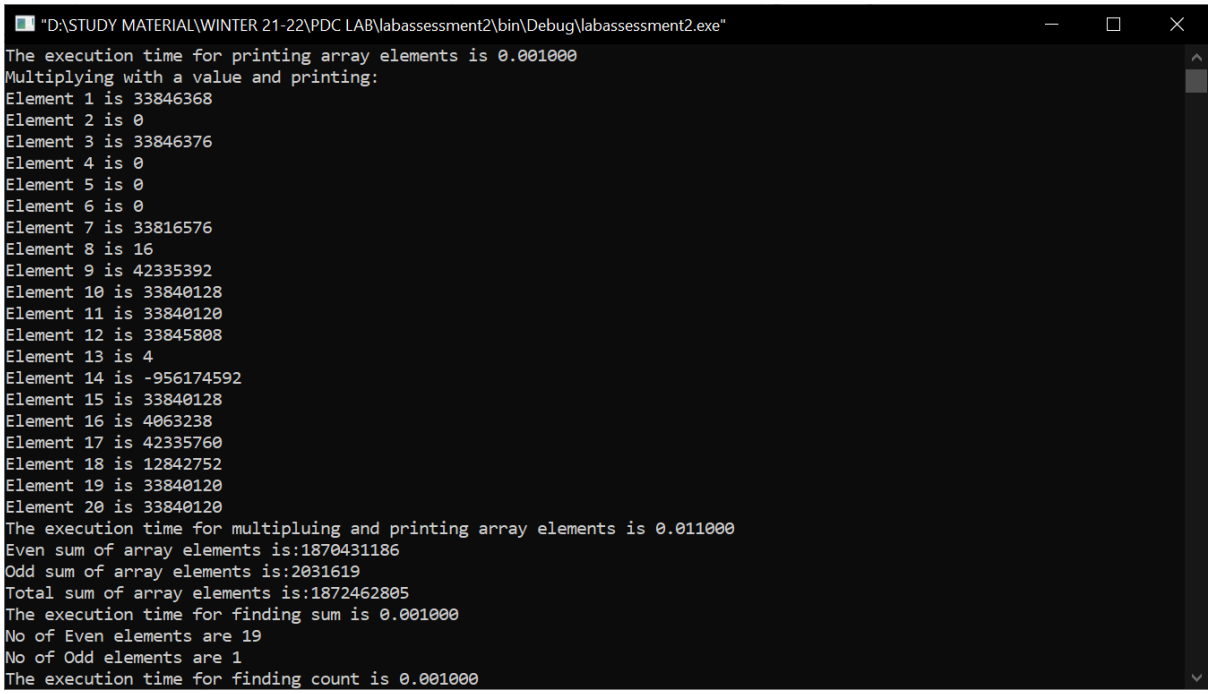
```
else
{
oddsum += A[i];
}
}
totalsum = evensum + oddsum;
printf("Even sum of array elements is:%d\n", evensum);
printf("Odd sum of array elements is:%d\n", oddsum);
 printf("Total sum of array elements is:%d\n", totalsum);
 end =omp_get_wtime();
total = end - start;
printf("The execution time for finding sum is %f\n",total);
start = omp_get_wtime();
int evencount=0,
oddcount = 0;
for (int i = 0; i < N; i++)
{
if (A[i] % 2 == 0)
{
evencount += 1;
}
else
{
oddcount += 1;
}
}
printf("No of Even elements are %d\n", evencount);
printf("No of Odd elements are %d\n", oddcount);
end = omp_get_wtime();
```

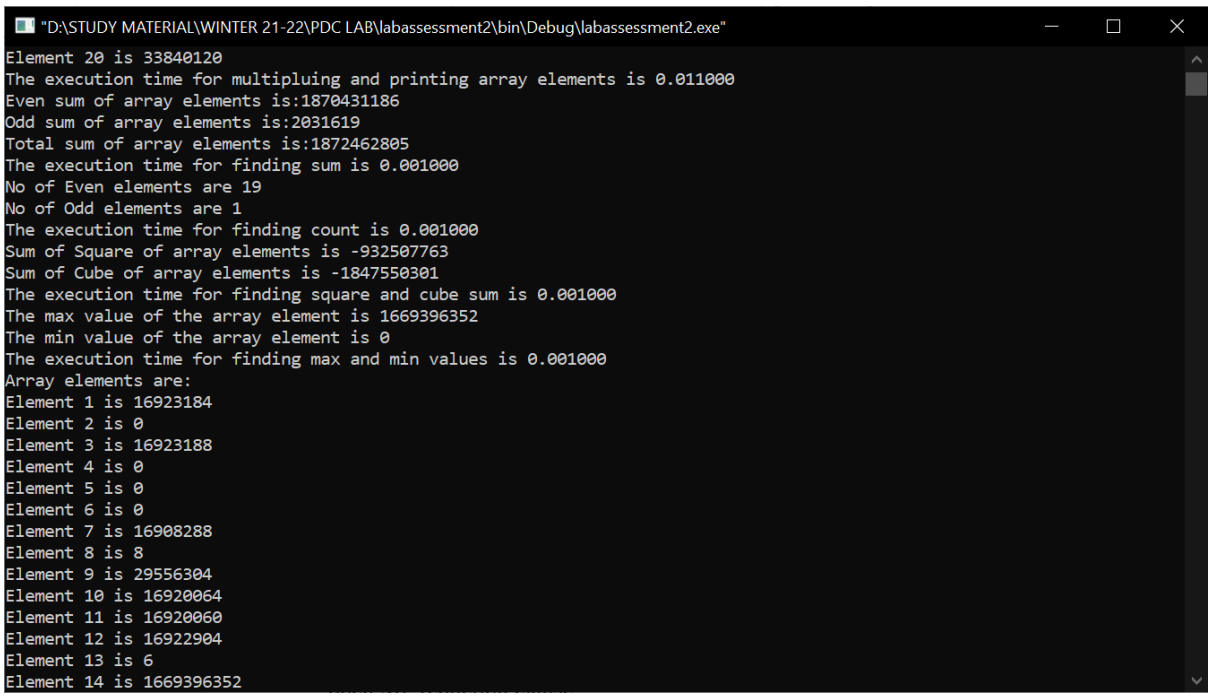
```
total =end - start;
printf("The execution time for finding count is %f\n", total);
start = omp_get_wtime();
int squaresum=0, cubesum=0,temp,temp1;
for (int i= 0; i < N; i++)
{
temp = 0;
temp1 = 0;
temp = A[i] * A[i];
temp1 = temp * A[i];
squaresum += temp;
cubesum += temp1;
}
printf("Sum of Square of array elements is %d\n", squaresum);
printf("Sum of Cube of array elements is %d\n", cubesum);
end =omp_get_wtime();
total = end - start;
printf("The execution time for finding square and cube sum is %f\n", total);
start =omp_get_wtime();
int max,min;
max = A[0];
for (i = 1; i < N; i++)
{
if (A[i] > max)
{
max = A[i];
}
}
printf("The max value of the array element is %d\n", max);
```

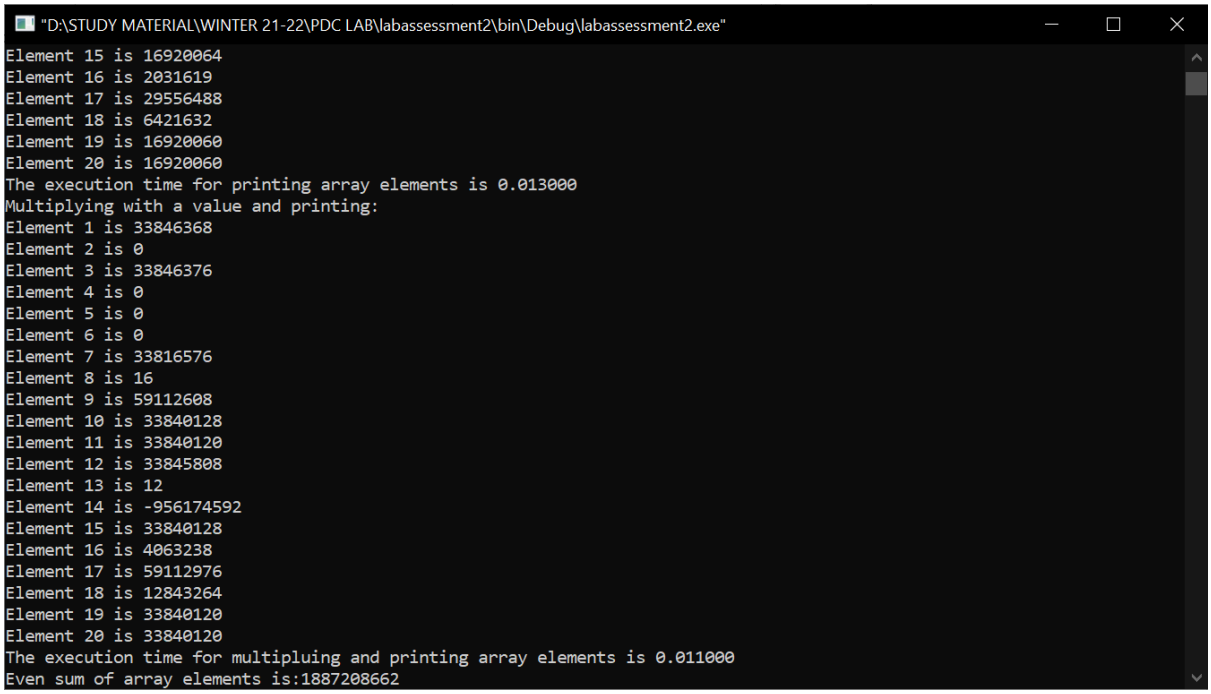
```
min = A[0];
for (i = 1; i < N; i++)
{
if (A[i] < min)
{
min = A[i];
}
}
 printf("The min value of the array element is %d\n", min);
 end =omp_get_wtime();
total = end - start;
printf("The execution time for finding max and min values is %f\n", total);
}
}
```

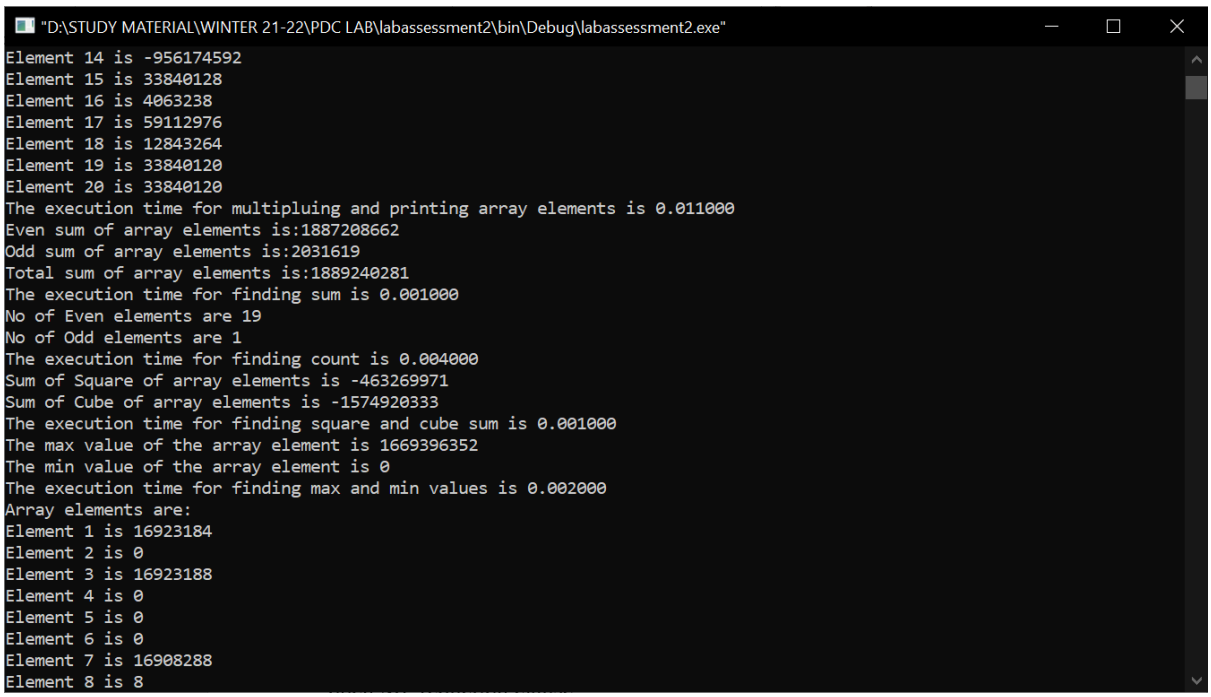
SCREENSHOT OF THE OUTPUTS: -

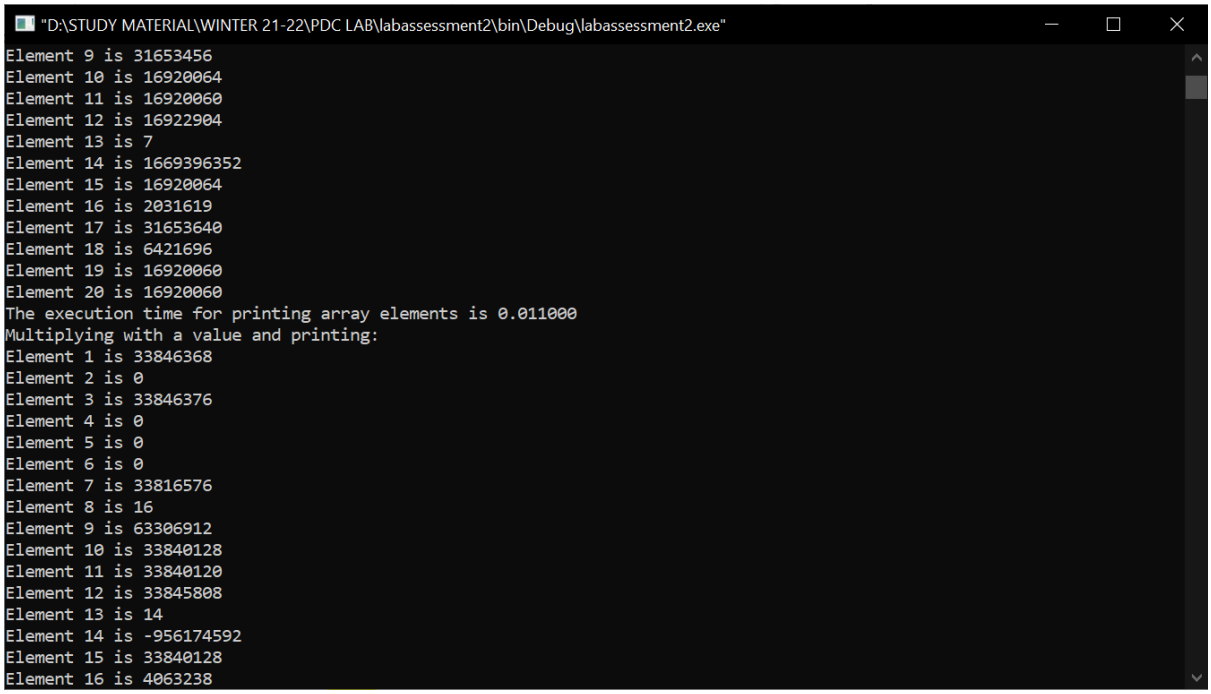


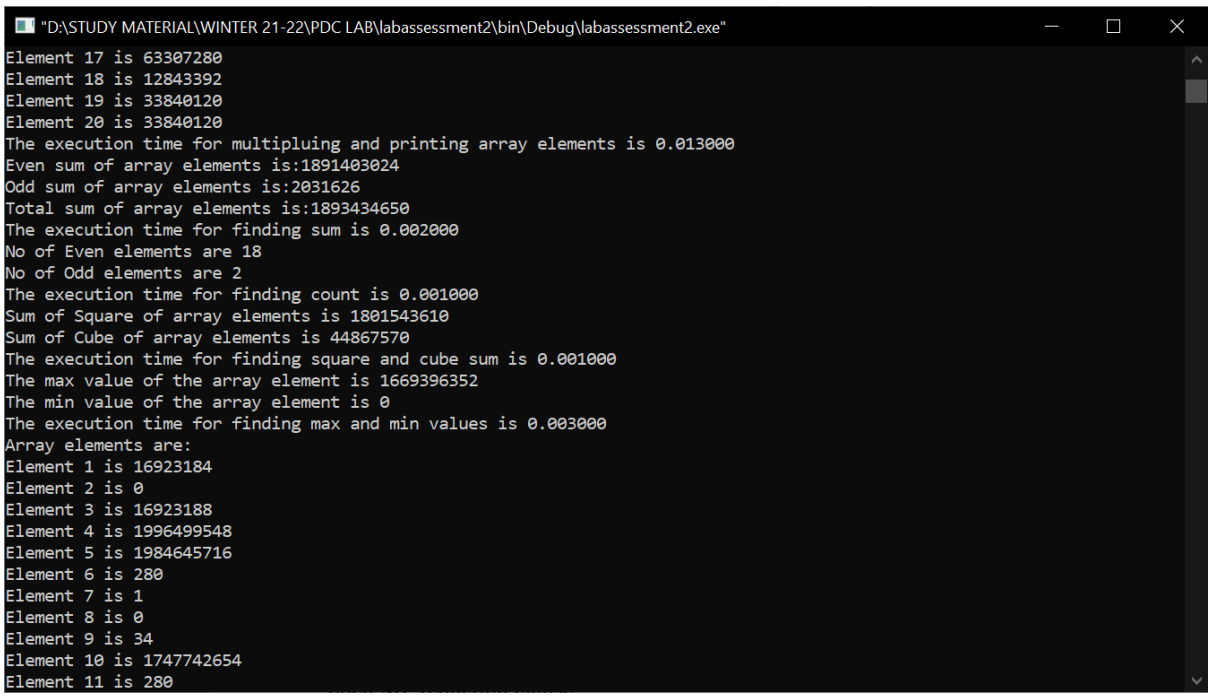

"D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\bin\Debug\labassessment2.exe"
The execution time for printing array elements is 0.001000
Multiplying with a value and printing:
Element 1 is 33846368
Element 2 is 0
Element 3 is 33846376
Element 4 is 0
Element 5 is 0
Element 6 is 0
Element 7 is 33816576
Element 8 is 16
Element 9 is 42335392
Element 10 is 33840128
Element 11 is 33840120
Element 12 is 33845808
Element 13 is 4
Element 14 is -956174592
Element 15 is 33840128
Element 16 is 4063238
Element 17 is 42335760
Element 18 is 12842752
Element 19 is 33840120
Element 20 is 33840120
The execution time for multipluing and printing array elements is 0.011000
Even sum of array elements is:1870431186
Odd sum of array elements is:2031619
Total sum of array elements is:1872462805
The execution time for finding sum is 0.001000
No of Even elements are 19
No of Odd elements are 1
The execution time for finding count is 0.001000

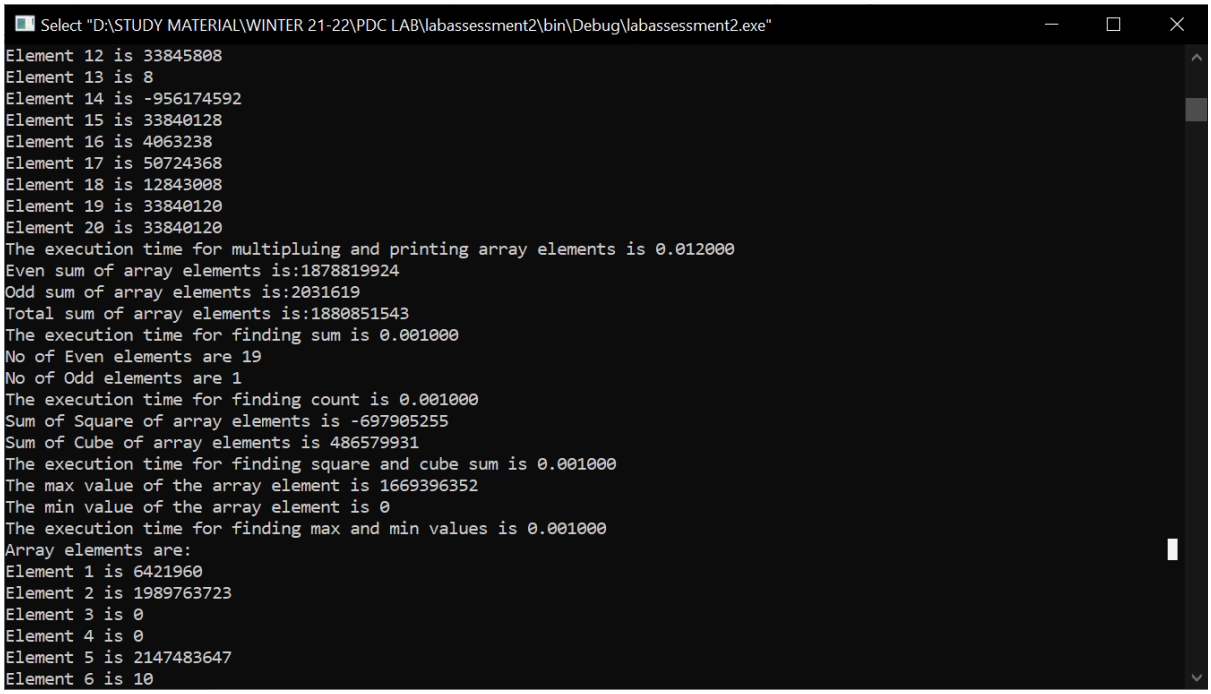

"D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\bin\Debug\labassessment2.exe"
Element 20 is 33840120
The execution time for multipluing and printing array elements is 0.011000
Even sum of array elements is:1870431186
Odd sum of array elements is:2031619
Total sum of array elements is:1872462805
The execution time for finding sum is 0.001000
No of Even elements are 19
No of Odd elements are 1
The execution time for finding count is 0.001000
Sum of Square of array elements is -932507763
Sum of Cube of array elements is -1847550301
The execution time for finding square and cube sum is 0.001000
The max value of the array element is 1669396352
The min value of the array element is 0
The execution time for finding max and min values is 0.001000
Array elements are:
Element 1 is 16923184
Element 2 is 0
Element 3 is 16923188
Element 4 is 0
Element 5 is 0
Element 6 is 0
Element 7 is 16908288
Element 8 is 8
Element 9 is 29556304
Element 10 is 16920064
Element 11 is 16920060
Element 12 is 16922904
Element 13 is 6
Element 14 is 1669396352

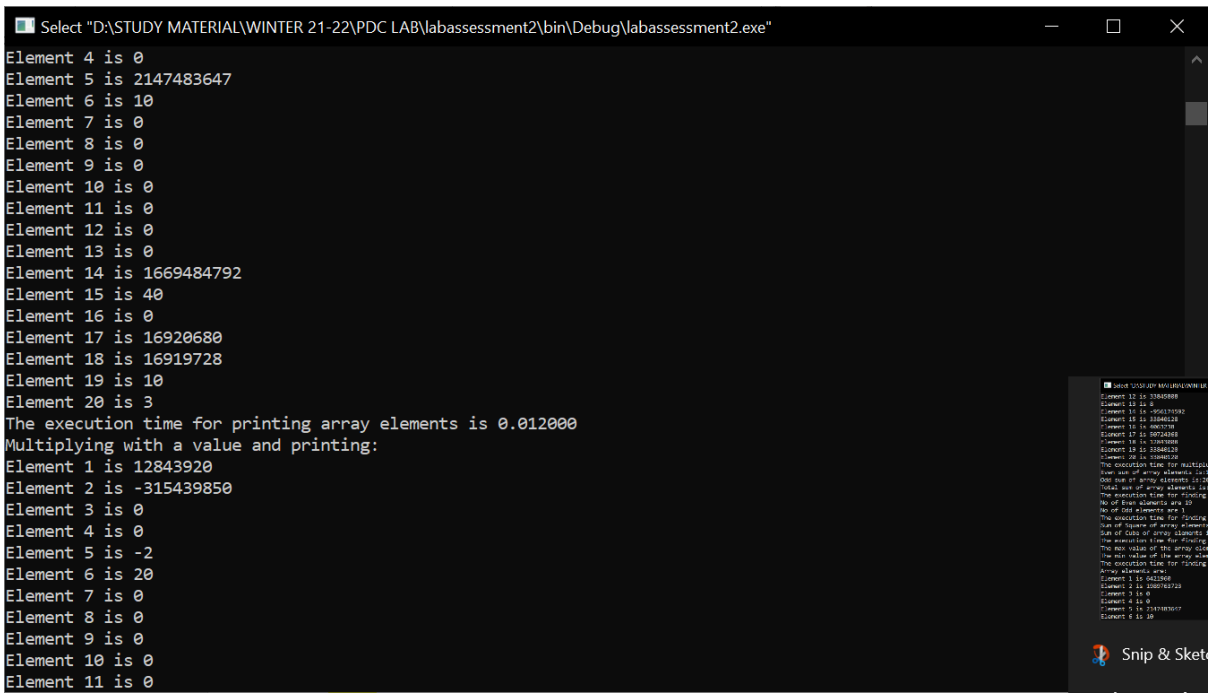

"D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\bin\Debug\labassessment2.exe"
Element 15 is 16920064
Element 16 is 2031619
Element 17 is 29556488
Element 18 is 6421632
Element 19 is 16920060
Element 20 is 16920060
The execution time for printing array elements is 0.013000
Multiplying with a value and printing:
Element 1 is 33846368
Element 2 is 0
Element 3 is 33846376
Element 4 is 0
Element 5 is 0
Element 6 is 0
Element 7 is 33816576
Element 8 is 16
Element 9 is 59112608
Element 10 is 33840128
Element 11 is 33840120
Element 12 is 33845808
Element 13 is 12
Element 14 is -956174592
Element 15 is 33840128
Element 16 is 4063238
Element 17 is 59112976
Element 18 is 12843264
Element 19 is 33840120
Element 20 is 33840120
The execution time for multipluing and printing array elements is 0.011000
Even sum of array elements is:1887208662

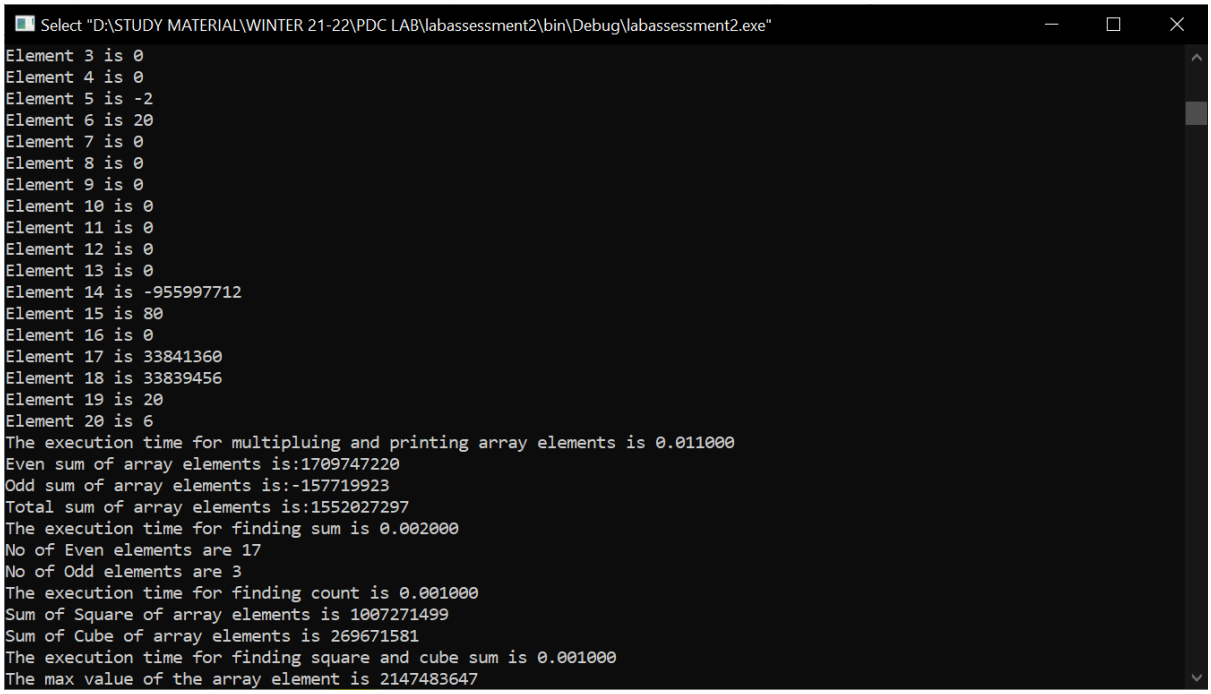

"D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\bin\Debug\labassessment2.exe"
Element 14 is -956174592
Element 15 is 33840128
Element 16 is 4063238
Element 17 is 59112976
Element 18 is 12843264
Element 19 is 33840120
Element 20 is 33840120
The execution time for multipluing and printing array elements is 0.011000
Even sum of array elements is:1887208662
Odd sum of array elements is:2031619
Total sum of array elements is:1889240281
The execution time for finding sum is 0.001000
No of Even elements are 19
No of Odd elements are 1
The execution time for finding count is 0.004000
Sum of Square of array elements is -463269971
Sum of Cube of array elements is -1574920333
The execution time for finding square and cube sum is 0.001000
The max value of the array element is 1669396352
The min value of the array element is 0
The execution time for finding max and min values is 0.002000
Array elements are:
Element 1 is 16923184
Element 2 is 0
Element 3 is 16923188
Element 4 is 0
Element 5 is 0
Element 6 is 0
Element 7 is 16908288
Element 8 is 8

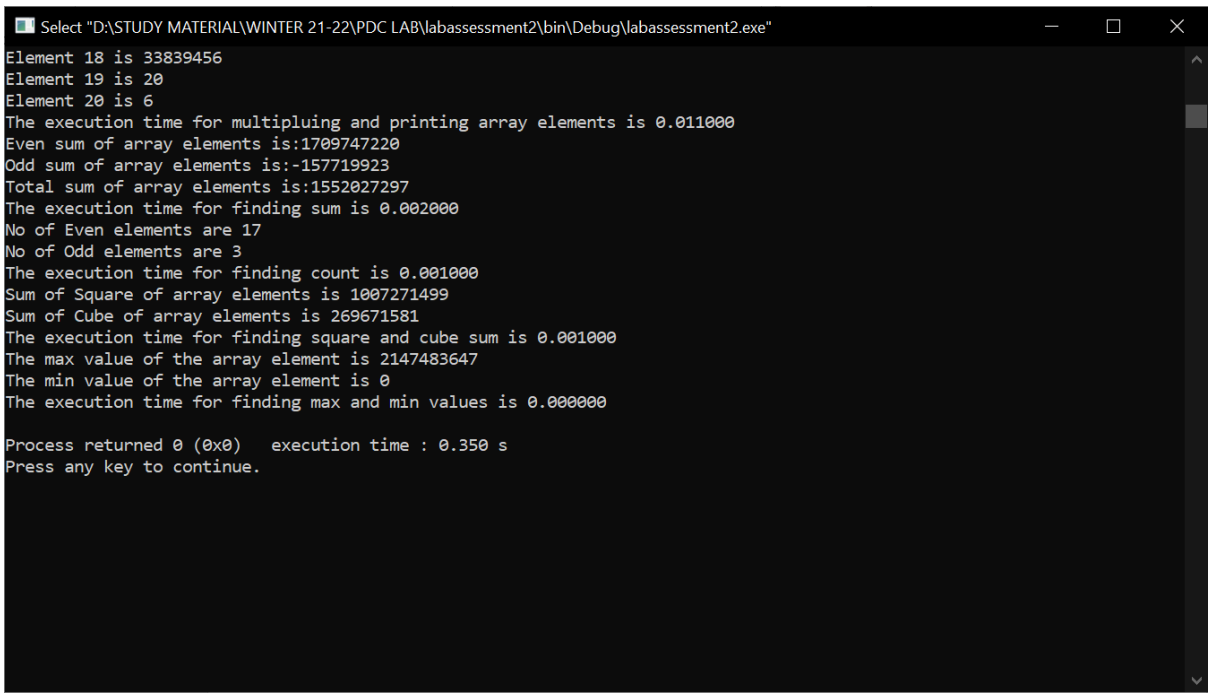

"D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\bin\Debug\labassessment2.exe"
Element 9 is 31653456
Element 10 is 16920064
Element 11 is 16920060
Element 12 is 16922904
Element 13 is 7
Element 14 is 1669396352
Element 15 is 16920064
Element 16 is 2031619
Element 17 is 31653640
Element 18 is 6421696
Element 19 is 16920060
Element 20 is 16920060
The execution time for printing array elements is 0.011000
Multiplying with a value and printing:
Element 1 is 33846368
Element 2 is 0
Element 3 is 33846376
Element 4 is 0
Element 5 is 0
Element 6 is 0
Element 7 is 33816576
Element 8 is 16
Element 9 is 63306912
Element 10 is 33840128
Element 11 is 33840120
Element 12 is 33845808
Element 13 is 14
Element 14 is -956174592
Element 15 is 33840128
Element 16 is 4063238

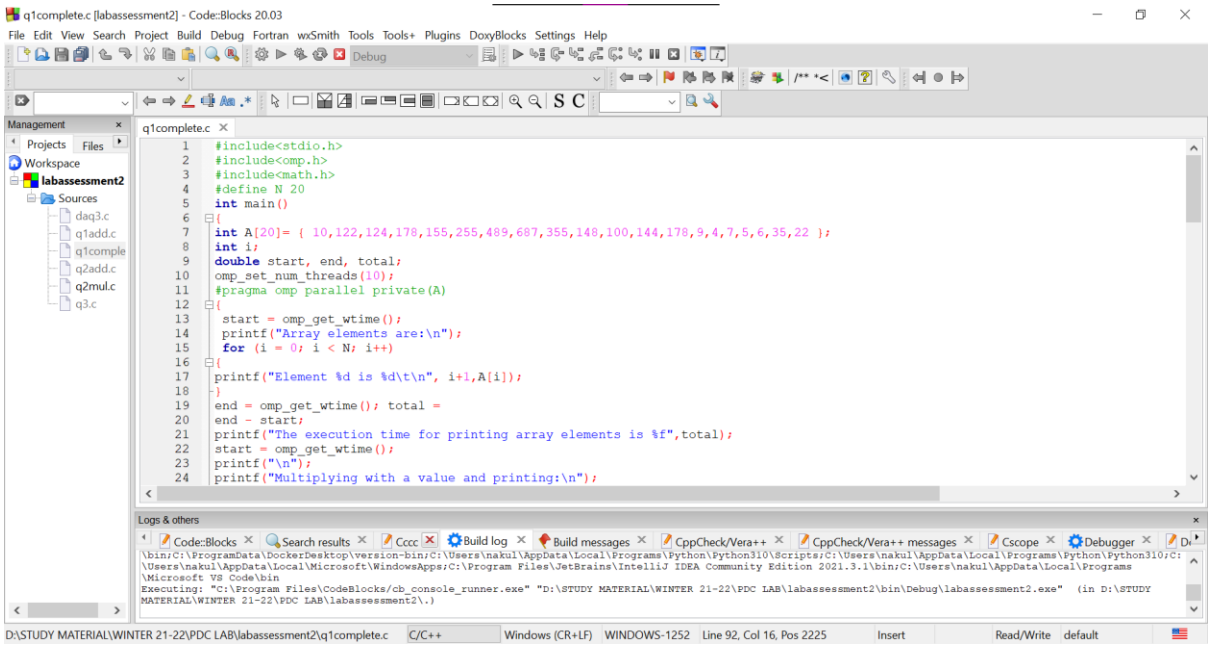

"D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\bin\Debug\labassessment2.exe"
Element 17 is 63307280
Element 18 is 12843392
Element 19 is 33840120
Element 20 is 33840120
The execution time for multipluing and printing array elements is 0.013000
Even sum of array elements is:1891403024
Odd sum of array elements is:2031626
Total sum of array elements is:1893434650
The execution time for finding sum is 0.002000
No of Even elements are 18
No of Odd elements are 2
The execution time for finding count is 0.001000
Sum of Square of array elements is 1801543610
Sum of Cube of array elements is 44867570
The execution time for finding square and cube sum is 0.001000
The max value of the array element is 1669396352
The min value of the array element is 0
The execution time for finding max and min values is 0.003000
Array elements are:
Element 1 is 16923184
Element 2 is 0
Element 3 is 16923188
Element 4 is 1996499548
Element 5 is 1984645716
Element 6 is 280
Element 7 is 1
Element 8 is 0
Element 9 is 34
Element 10 is 1747742654
Element 11 is 280


Select "D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\bin\Debug\labassessment2.exe"
Element 12 is 33845808
Element 13 is 8
Element 14 is -956174592
Element 15 is 33840128
Element 16 is 4063238
Element 17 is 50724368
Element 18 is 12843008
Element 19 is 33840120
Element 20 is 33840120
The execution time for multipluing and printing array elements is 0.012000
Even sum of array elements is:1878819924
Odd sum of array elements is:2031619
Total sum of array elements is:1880851543
The execution time for finding sum is 0.001000
No of Even elements are 19
No of Odd elements are 1
The execution time for finding count is 0.001000
Sum of Square of array elements is -697905255
Sum of Cube of array elements is 486579931
The execution time for finding square and cube sum is 0.001000
The max value of the array element is 1669396352
The min value of the array element is 0
The execution time for finding max and min values is 0.001000
Array elements are:
Element 1 is 6421960
Element 2 is 1989763723
Element 3 is 0
Element 4 is 0
Element 5 is 2147483647
Element 6 is 10


Select "D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\bin\Debug\labassessment2.exe"
Element 4 is 0
Element 5 is 2147483647
Element 6 is 10
Element 7 is 0
Element 8 is 0
Element 9 is 0
Element 10 is 0
Element 11 is 0
Element 12 is 0
Element 13 is 0
Element 14 is 1669484792
Element 15 is 40
Element 16 is 0
Element 17 is 16920680
Element 18 is 16919728
Element 19 is 10
Element 20 is 3
The execution time for printing array elements is 0.012000
Multiplying with a value and printing:
Element 1 is 12843920
Element 2 is -315439850
Element 3 is 0
Element 4 is 0
Element 5 is -2
Element 6 is 20
Element 7 is 0
Element 8 is 0
Element 9 is 0
Element 10 is 0
Element 11 is 0


Select "D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\bin\Debug\labassessment2.exe"
Element 3 is 0
Element 4 is 0
Element 5 is -2
Element 6 is 20
Element 7 is 0
Element 8 is 0
Element 9 is 0
Element 10 is 0
Element 11 is 0
Element 12 is 0
Element 13 is 0
Element 14 is -955997712
Element 15 is 80
Element 16 is 0
Element 17 is 33841360
Element 18 is 33839456
Element 19 is 20
Element 20 is 6
The execution time for multipluing and printing array elements is 0.011000
Even sum of array elements is:1709747220
Odd sum of array elements is:-157719923
Total sum of array elements is:1552027297
The execution time for finding sum is 0.002000
No of Even elements are 17
No of Odd elements are 3
The execution time for finding count is 0.001000
Sum of Square of array elements is 1007271499
Sum of Cube of array elements is 269671581
The execution time for finding square and cube sum is 0.001000
The max value of the array element is 2147483647


Select "D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\bin\Debug\labassessment2.exe"
Element 18 is 33839456
Element 19 is 20
Element 20 is 6
The execution time for multipluing and printing array elements is 0.011000
Even sum of array elements is:1709747220
Odd sum of array elements is:-157719923
Total sum of array elements is:1552027297
The execution time for finding sum is 0.002000
No of Even elements are 17
No of Odd elements are 3
The execution time for finding count is 0.001000
Sum of Square of array elements is 1007271499
Sum of Cube of array elements is 269671581
The execution time for finding square and cube sum is 0.001000
The max value of the array element is 2147483647
The min value of the array element is 0
The execution time for finding max and min values is 0.000000
Process returned 0 (0x0) execution time : 0.350 s
Press any key to continue.



Question 2)

AIM: -Find the sum and product of n – elements in an array using open MP reduction clause.

ALGORITHM:

1. Start
2. Read variables i, n and sum
3. Read the arrays a and b
4. Initialize n=100
5. Run a for loop and initialize the arrays
6. Initialize sum=0
7. Start the parallel loop and store the value of parallel executions into the variable sum

NOTE: #pragma omp parallel for turns the loop into a parallel loop. The reduction(+:sum)

declares that we're reducing the input array by summing into the variable sum, so after the partial loops are done, their results are summed into this variable.

8. Display sum
9. Stop

SOURCE CODE:

```
#include <omp.h>
#include <stdio.h>
#include <stdlib.h>
int main (int argc, char *argv[])
{
int i, n;
float a[100], b[100], sum;
n = 100;
for (i=0; i < n; i++)
a[i] = b[i] = i * 1.0;
sum = 0.0;
#pragma omp parallel for reduction(+:sum)
for (i=0; i < n; i++)
sum = sum + (a[i] * b[i]);
printf(" Sum = %f\n",sum);
}
```

AND

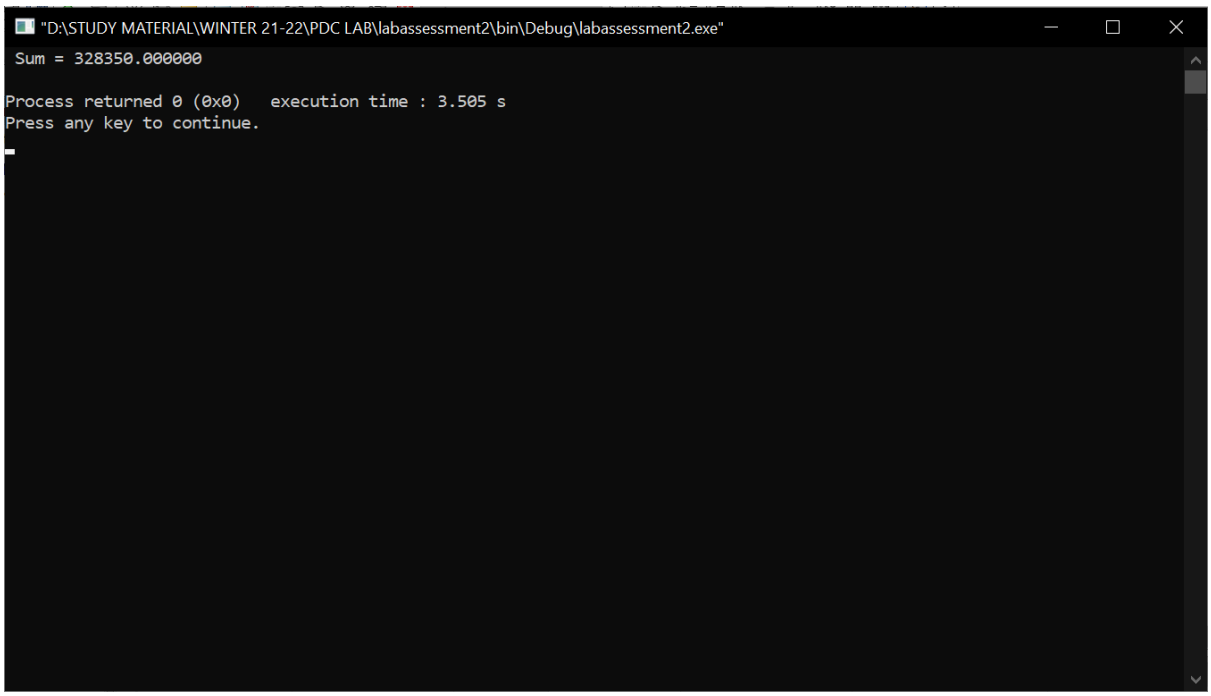
```
#include<stdio.h>
#include<omp.h>
#include <stdlib.h>

/* Main Program */
int main()
{
        float  *array_A, sum, prod, *checkarray, serialsum, serialprod;
        int  arraysize, i, k, Noofthreads;

        printf("Nakul Jadeja \n 19BCE060 \n\n");
```

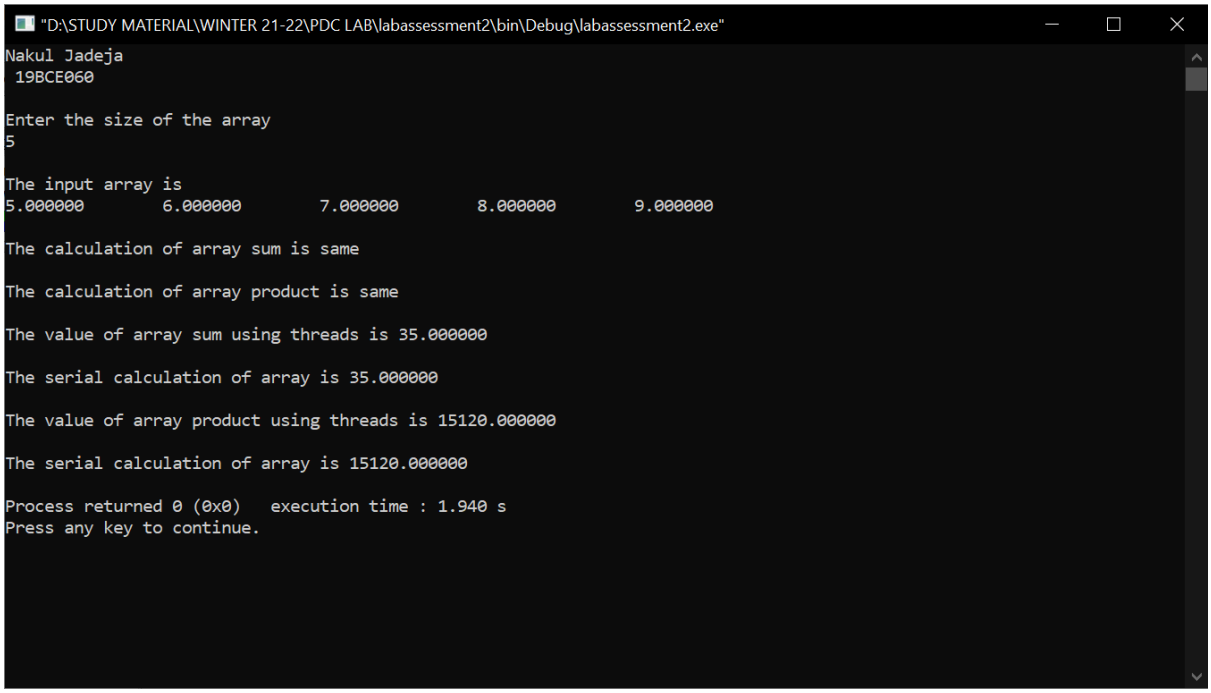
```
        printf("Enter the size of the array \n");
        scanf("%d", &arraysize);

        if (arraysize <= 0) {
                printf("Positive Number Required\n");
                exit(1);
        }

        /* Dynamic Memory Allocation */
        array_A = (float *) malloc(sizeof(float) * arraysize);
        checkarray = (float *) malloc(sizeof(float) * arraysize);

        for (i = 0; i < arraysize; i++) {
                array_A[i] = i + 5;
                checkarray[i] = array_A[i];
        }

        printf("\nThe input array is \n");
        for (i = 0; i < arraysize; i++)
                printf("%f \t", array_A[i]);
        printf("\n");

        sum = 0.0;
        prod= 1.0;

        /* OpenMP Parallel For With Reduction Clause */

#pragma omp parallel for reduction(+ : sum)
        for (i = 0; i < arraysize; i++)
```

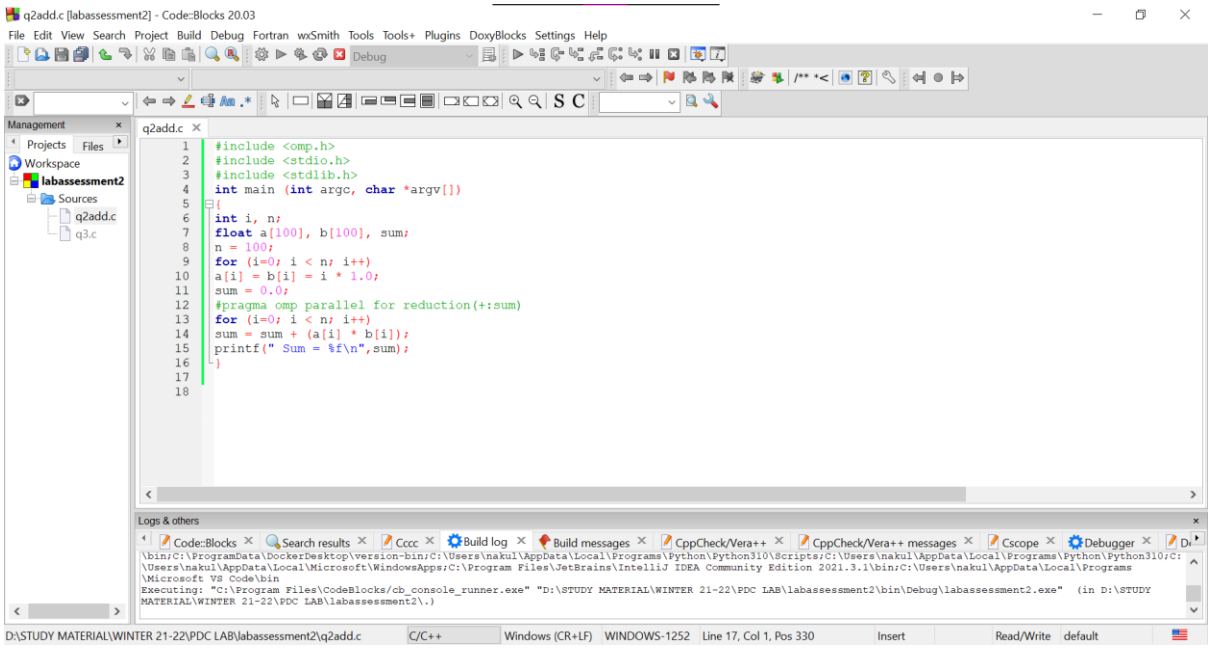
```
            sum = sum + array_A[i];

#pragma omp parallel for reduction(* : prod)
        for (i = 0; i < arraysize; i++)
            prod = prod * array_A[i];

        /* Serial Calculation */

        serialsum = 0.0;
        serialprod = 1.0;
        for (i = 0; i < arraysize; i++)
        {
            serialsum = serialsum + array_A[i];
            serialprod = serialprod * array_A[i];
        }

        /* Output Checking */

        if (serialsum != sum) {
            printf("\nThe calculation of array sum is different \n");
            exit(1);
        } else
            printf("\nThe calculation of array sum is same\n");

        if (serialprod != prod) {
            printf("\nThe calculation of array product is different \n");
            exit(1);
        } else
            printf("\nThe calculation of array product is same\n");
```

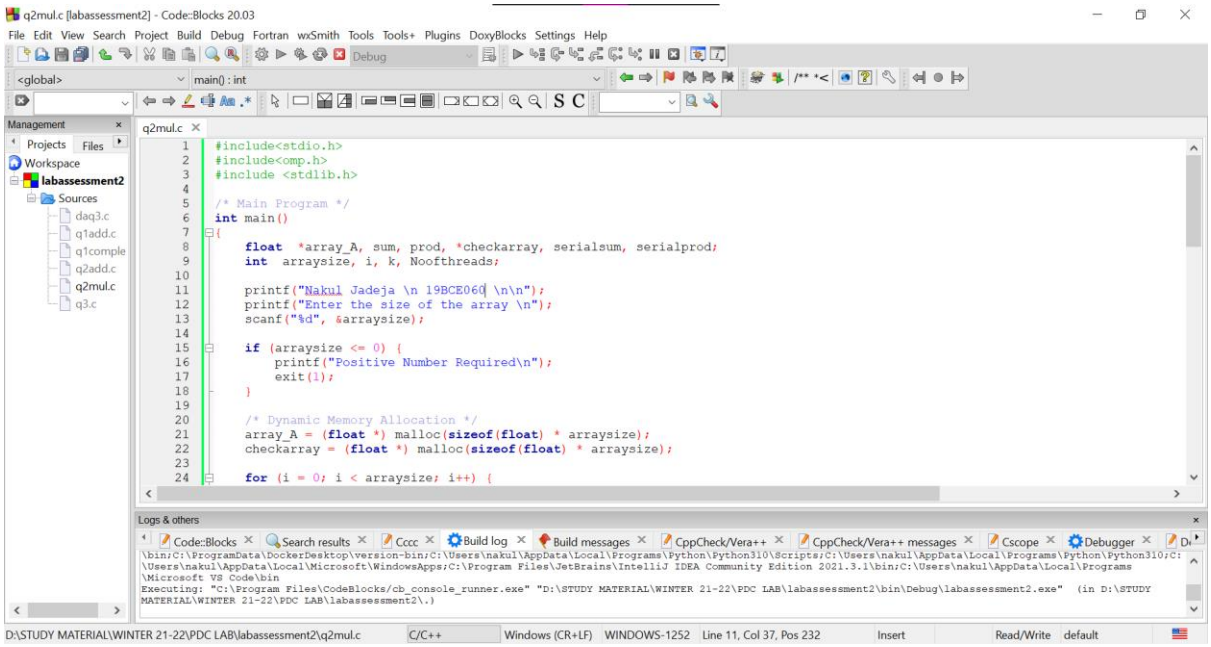
```
    /* Freeing Memory Which Was Allocated */

    free(checkarray);
    free(array_A);

    printf("\nThe value of array sum using threads is %f\n", sum);
    printf("\nThe serial calculation of array is %f\n\n", serialsum);
    printf("The value of array product using threads is %f\n", prod);
    printf("\nThe serial calculation of array is %f\n", serialprod);
}
```

SCREENSHOT: -




"D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\bin\Debug\labassessment2.exe"
Nakul Jadeja
19BCE060
Enter the size of the array
5
The input array is
5.000000 6.000000 7.000000 8.000000 9.000000
The calculation of array sum is same
The calculation of array product is same
The value of array sum using threads is 35.000000
The serial calculation of array is 35.000000
The value of array product using threads is 15120.000000
The serial calculation of array is 15120.000000
Process returned 0 (0x0) execution time : 1.940 s
Press any key to continue.


q2add.c [labassessment2] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects Files
Workspace
labassessment2
Sources
q2add.c
q3.c
q2add.c
#include <omp.h>
#include <stdio.h>
#include <stdlib.h>
int main (int argc, char *argv[])
{
int i, n;
float a[100], b[100], sum;
n = 100;
for (i=0; i < n; i++)
a[i] = b[i] = i * 1.0;
sum = 0.0;
#pragma omp parallel for reduction(+:sum)
for (i=0; i < n; i++)
sum = sum + (a[i] * b[i]);
printf(" Sum = %f\n",sum);
}
Logs & others
Code::Blocks
Search results
Cccc
Build log
Build messages
CppCheck/Vera++
CppCheck/Vera++ messages
Cscope
Debugger
Executing: "C:\Program Files\CodeBlocks/cb_console_runner.exe" "D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\bin\Debug\labassessment2.exe" (in D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\.)
D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\q2add.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 17, Col 1, Pos 330
Insert
Read/Write
default



Question 3)

AIM: -Write c code using Open MP to add two vectors and measure the time.

ALGORITHM: -

1. First, we begin with mentioning the header files and define the array size whose elements

will be added together.

2. Then, define the number of threads used for vector addition and take the input values.

3. Initialize a and b with integer values and determine elements each will work on.

4. Print he output and clear the memory.

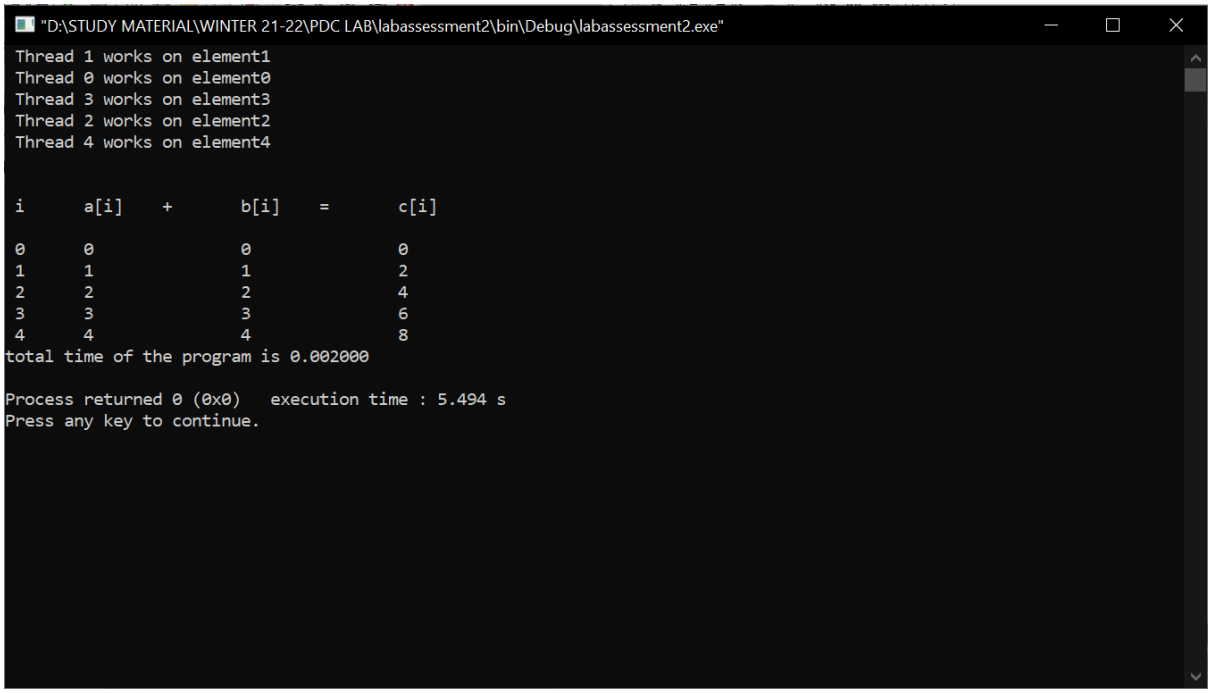
SOURCE CODE: -

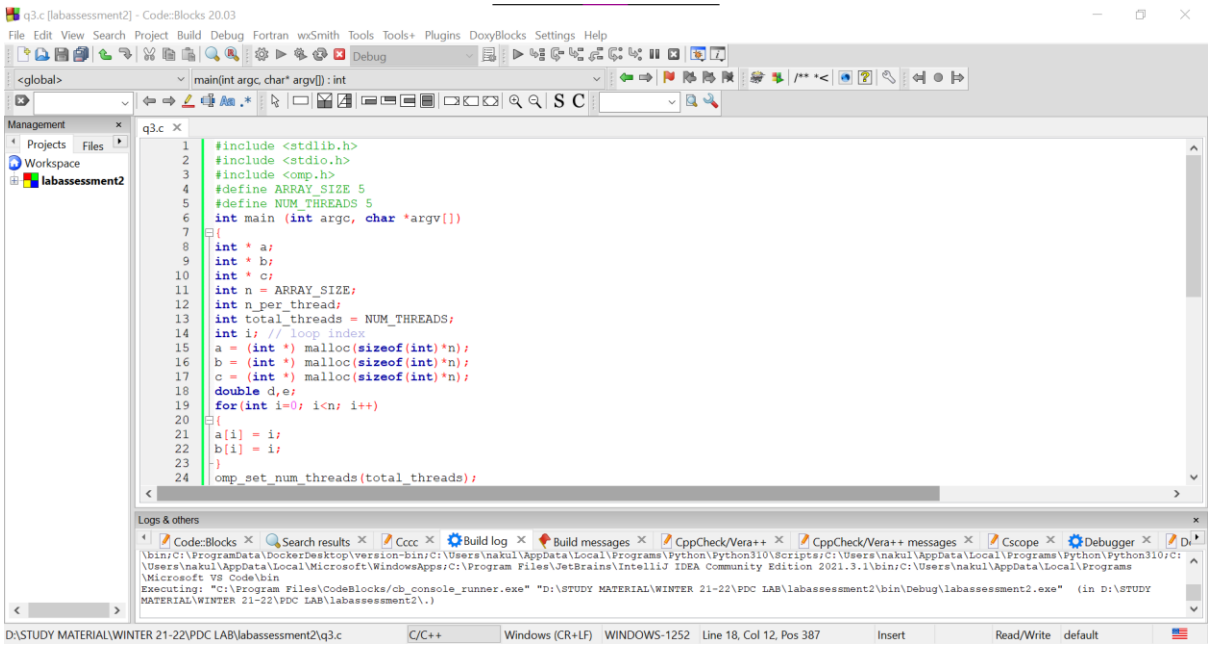
```
#include <stdlib.h>

#include <stdio.h>

#include <omp.h>

#define ARRAY_SIZE 5
```

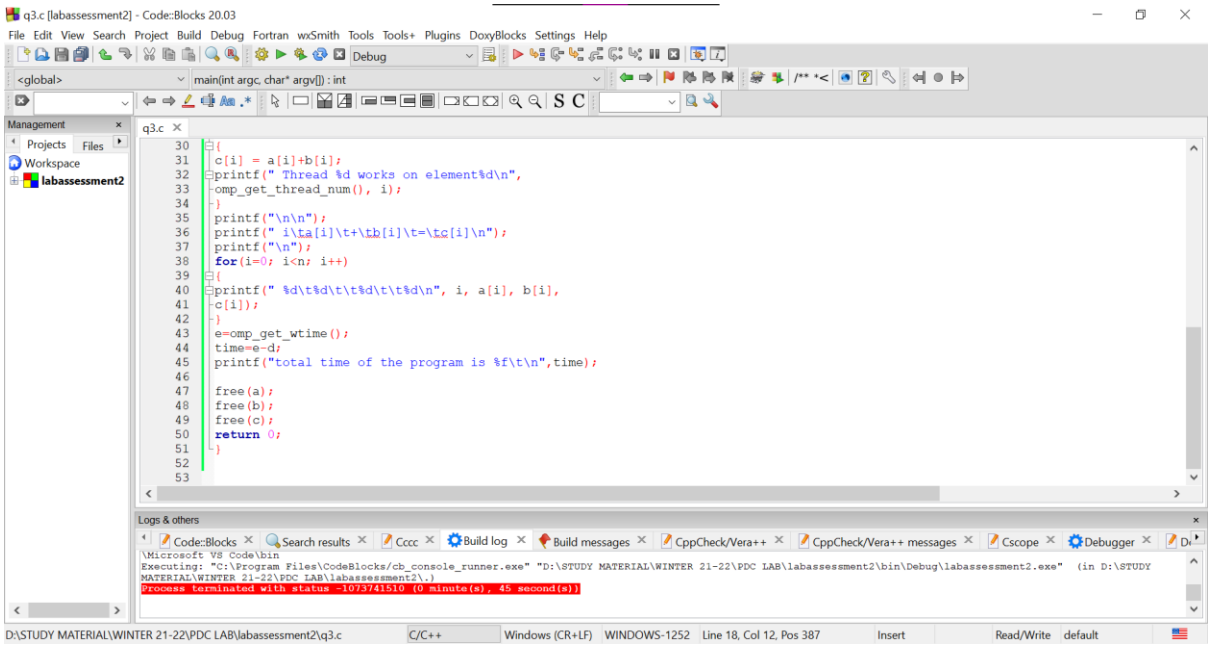
```
#define NUM_THREADS 5
int main (int argc, char *argv[])
{
int * a;
int * b;
int * c;
int n = ARRAY_SIZE;
int n_per_thread;
int total_threads = NUM_THREADS;
int i; // loop index
a = (int *) malloc(sizeof(int)*n);
b = (int *) malloc(sizeof(int)*n);
c = (int *) malloc(sizeof(int)*n);
double d,e;
for(int i=0; i<n; i++)
{
a[i] = i;
b[i] = i;
}
omp_set_num_threads(total_threads);
n_per_thread = n/total_threads;
double time;
d=omp_get_wtime();
#pragma omp parallel for shared(a, b, c) private(i) schedule(static,
n_per_thread)
for(i=0; i<n; i++)
{
c[i] = a[i]+b[i];
```

```
printf(" Thread %d works on element%d\n",
omp_get_thread_num(), i);
}
printf("\n\n");
printf(" i\ta[i]\t+\tb[i]\t=\tc[i]\n");
printf("\n");
for(i=0; i<n; i++)
{
printf(" %d\t%d\t\t%d\t\t%d\n", i, a[i], b[i],
c[i]);
}
e=omp_get_wtime();
time=e-d;
printf("total time of the program is %f\t\n",time);

free(a);
free(b);
free(c);
return 0;
}
```

SCREENSHOTS:


"D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\bin\Debug\labassessment2.exe"
Thread 1 works on element1
Thread 0 works on element0
Thread 3 works on element3
Thread 2 works on element2
Thread 4 works on element4
i a[i] + b[i] = c[i]
0 0 0 0
1 1 1 2
2 2 2 4
3 3 3 6
4 4 4 8
total time of the program is 0.002000
Process returned 0 (0x0) execution time : 5.494 s
Press any key to continue.


q3.c [labassessment2] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects Files
Workspace
labassessment2
q3.c
#include <stdlib.h>
#include <stdio.h>
#include <omp.h>
#define ARRAY_SIZE 5
#define NUM_THREADS 5
int main (int argc, char *argv[])
{
int * a;
int * b;
int * c;
int n = ARRAY_SIZE;
int n_per_thread;
int total_threads = NUM_THREADS;
int i; // loop index
a = (int *) malloc(sizeof(int)*n);
b = (int *) malloc(sizeof(int)*n);
c = (int *) malloc(sizeof(int)*n);
double d,e;
for(int i=0; i<n; i++)
{
a[i] = i;
b[i] = i;
}
omp_set_num_threads(total_threads);
Logs & others
Code::Blocks Search results Cccc Build log Build messages CppCheck/Vera++ CppCheck/Vera++ messages Cscope Debugger
Executing: "C:\Program Files\CodeBlocks/cb_console_runner.exe" "D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\bin\Debug\labassessment2.exe" (in D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\.)
D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\q3.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 18, Col 12, Pos 387
Insert
Read/Write
default


q3.c [labassessment2] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Debug
<global>
main(int argc, char* argv[]) : int
Management
Projects
Files
Workspace
labassessment2
q3.c
30 {
31 c[i] = a[i]+b[i];
32 printf(" Thread %d works on element%d\n",
33 omp_get_thread_num(), i);
34 }
35 printf("\n\n");
36 printf(" i\ta[i]\t+\tb[i]\t=\tc[i]\n");
37 printf("\n");
38 for(i=0; i<n; i++)
39 {
40 printf(" %d\t%d\t\t%d\t\t%d\n", i, a[i], b[i],
41 c[i]);
42 }
43 e=omp_get_wtime();
44 time=e-d;
45 printf("total time of the program is %f\t\n",time);
46
47 free(a);
48 free(b);
49 free(c);
50 return 0;
51 }
52
53
Logs & others
Code::Blocks
Search results
Cccc
Build log
Build messages
CppCheck/Vera++
CppCheck/Vera++ messages
Cscope
Debugger
\Microsoft VS Code\bin
Executing: "C:\Program Files\CodeBlocks/cb_console_runner.exe" "D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\bin\Debug\labassessment2.exe" (in D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\.)
Process terminated with status -1073741510 (0 minute(s), 45 second(s))
D:\STUDY MATERIAL\WINTER 21-22\PDC LAB\labassessment2\q3.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 18, Col 12, Pos 387
Insert
Read/Write
default