**Class:** Final Year B.Tech(Computer Science and Engineering)

**Year:** 2025-26 **Semester:** 1

**Course:** High Performance Computing Lab

**PRN : 22510021 Batch : B7**

**Practical No. 2**

**Exam Seat No:**

**Title of practical: Study and implementation of basic OpenMP clauses**

Implement following Programs using OpenMP with C:

1. Vector Scalar Addition
2. Calculation of value of Pi

Analyse the performance of your programs for different number of threads and Data size.

**Problem Statement 1:** Vector Scalar Addition

**Screenshots:**

1. **Code:**

**#include <stdio.h>**

**#include <omp.h>**

**int main(){**

**int n;**

**printf("Enter desired number of threads \n");**

**scanf("%d", &n);**

**int arr1[5] = {3, 6, 9, 11, 21};**

**int arr2[5] = {21, 11, 9, 6, 3};**

**int res[5];**

**omp\_set\_num\_threads(n);**

**#pragma omp parallel for**

**for(int i=0; i<5; i++){**

**res[i] = arr1[i] + arr2[i];**

**printf(" addition of %dth elements is done by thread number %d \n", i, omp\_get\_thread\_num());**

**}**

**printf(" Result array is " );**

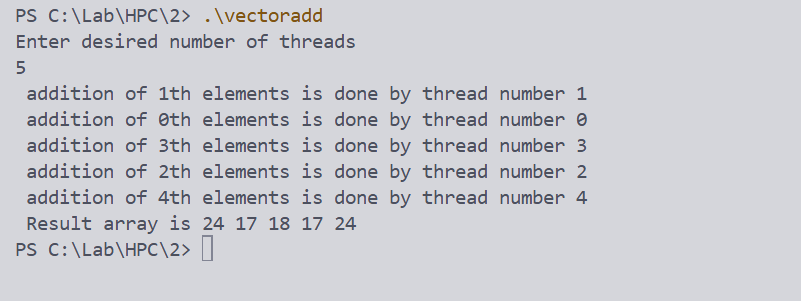
**for(int i=0; i<5; i++){**

**printf("%d ", res[i]);**

**}**

**}**

1. **Result:**

****

**Information:**

**Analysis:**

**Problem Statement 2:**

**Screenshots:**

**Code:**

**#include <stdio.h>**

**#include <omp.h>**

**int main() {**

**printf("1 Billion Slices \n");**

**long num\_slices = 1000000000; *// Number of rectangles***

**double width = 1.0 / (double)num\_slices; *// Width of each rectangle***

**double sum = 0.0;**

**double pi;**

**int i;**

**double height;**

**#pragma omp parallel for reduction(+:sum) private(height)**

**for (i = 0; i < num\_slices; i++) {**

**double midpoint = (i + 0.5) \* width; *// Midpoint of current rectangle***

**height = 4.0 / (1.0 + midpoint \* midpoint); *// Height of rectangle at midpoint***

**sum += height;**

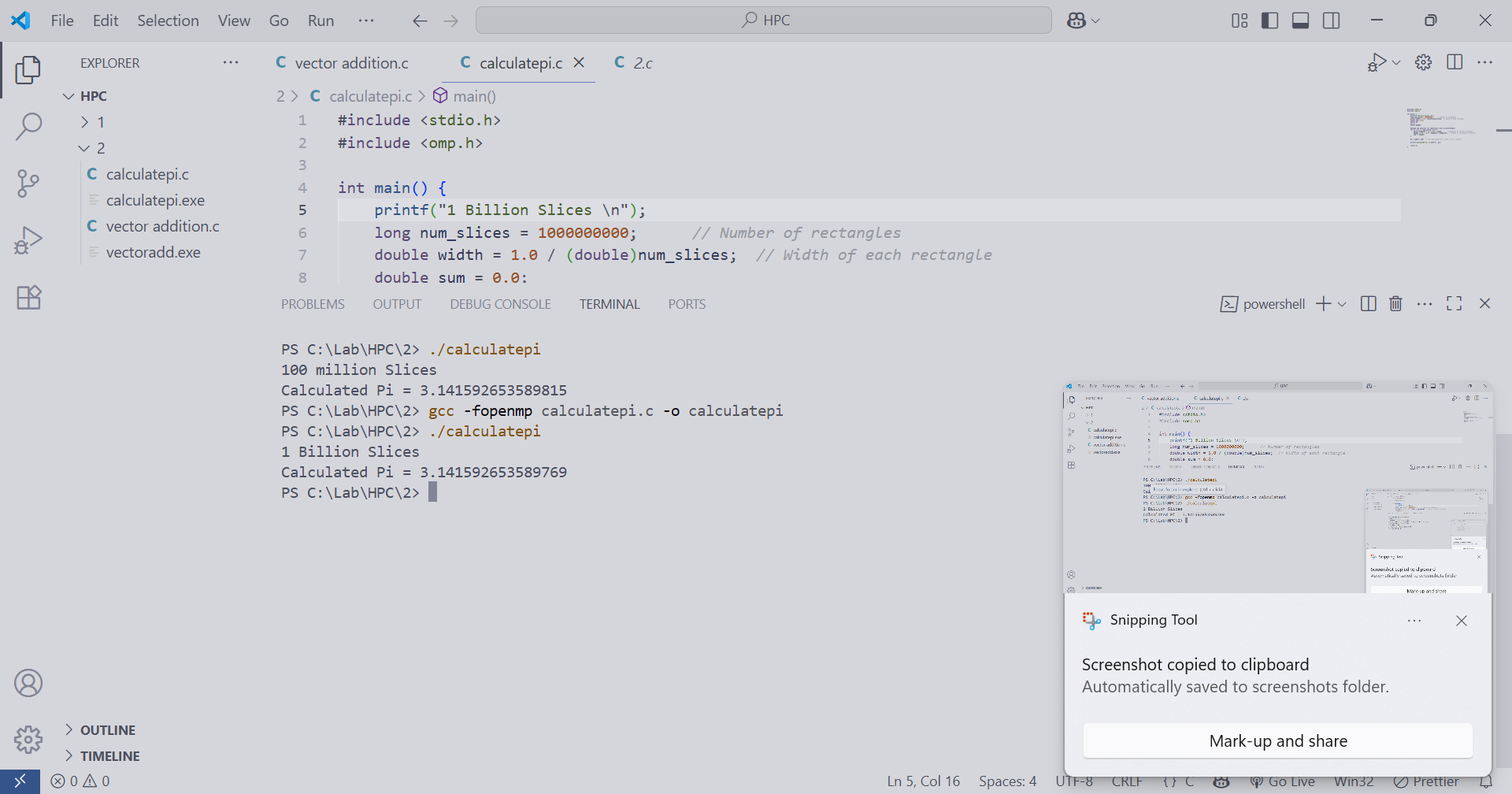
**}**

**pi = width \* sum; *// Area approximation = width \* sum of heights***

**printf("Calculated Pi = %.15lf\n", pi);**

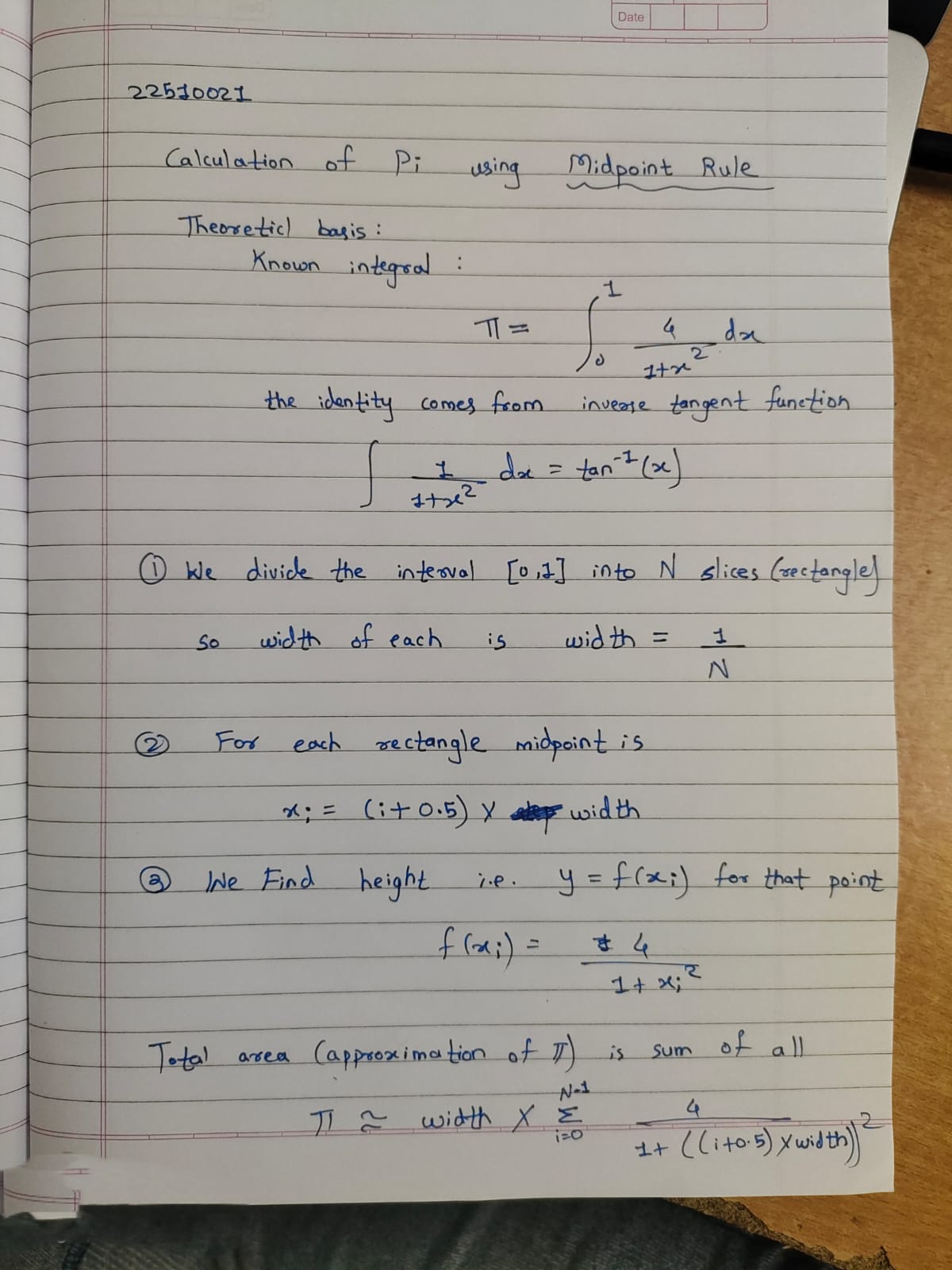
**return 0;**

**}**

**Result : **

**Information:**

**Analysis:**

****

**Github Link:**

<https://github.com/22510021-Shrikrishna/HPC.git>