OS LAB 12

Q1.

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <omp.h>

#define N 100

void print(int matrix[][N])

{

int i, j;

for(i=0; i<N; i++)

{

for(j=0; j<N; j++)

{

printf("%d ", matrix[i][j]);

}

printf("\n");

}

printf("\n\n");

}

int main()

{

srand(time(0));

int matrix1[N][N], matrix2[N][N], i, j;

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

matrix1[i][j] = rand() % 100 + 1;

matrix2[i][j] = rand() % 100 + 1;

}

}

int n=100;

print(matrix1);

print(matrix2);

int sum[N][N];

#pragma omp parallel private(i, j) shared(sum)

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

#pragma omp critical

sum[i][j] = matrix1[i][j] + matrix2[i][j];

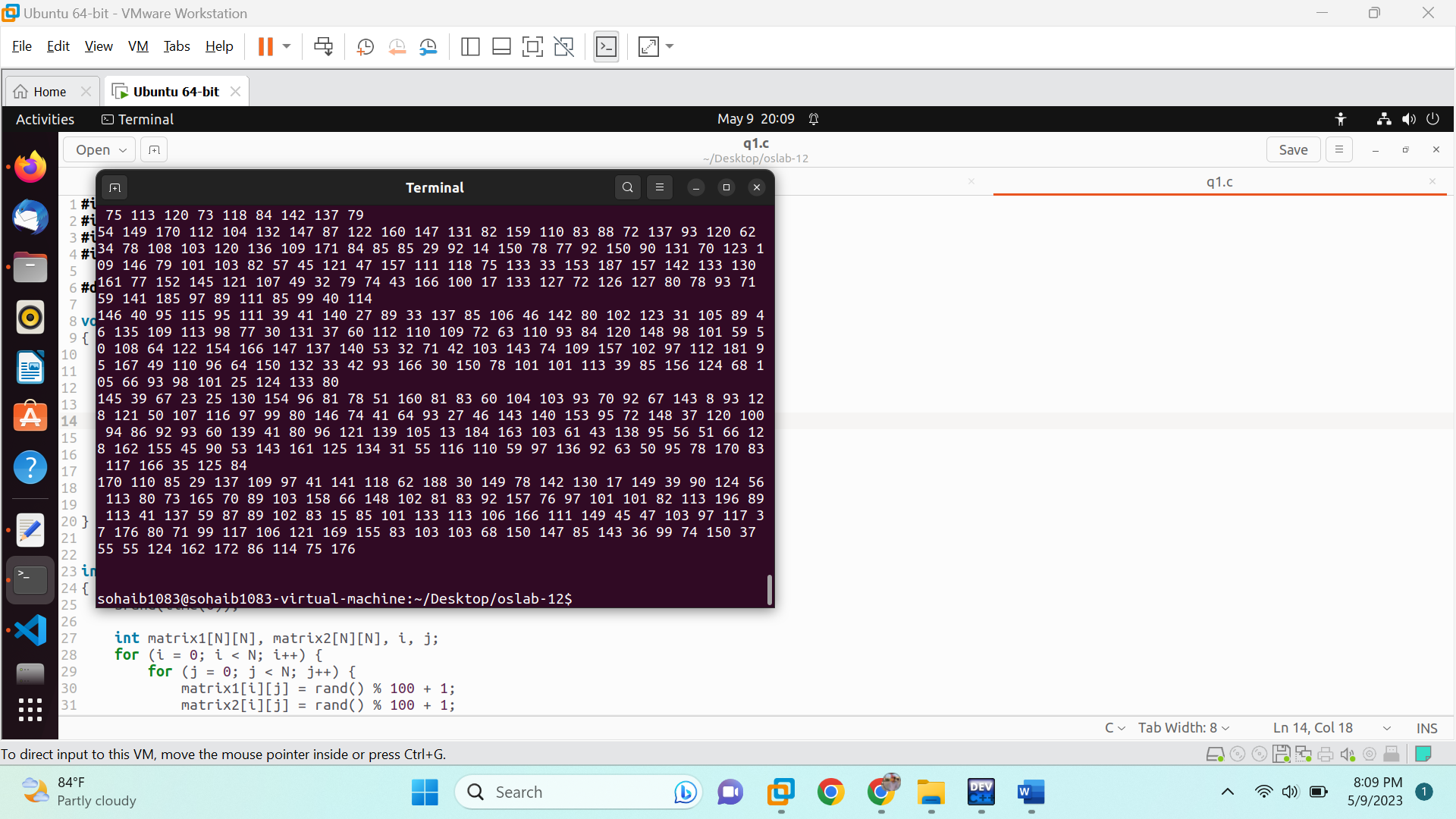
}

}

print(sum);

return 0;

}



Q2.

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

#include <omp.h>

#define n 1000

int main (void)

{

srand(0);

int arr[n];

int max = rand() % 99 + 1;

arr[0]=max;

int mx\_index=0, i;

#pragma omp parallel for shared(default) private(i)

for (i=1; i<n; i++){

arr[i] = rand() % 100 + 1;

#pragma omp critical

if (arr[i] > max){

max=arr[i];

mx\_index=i;

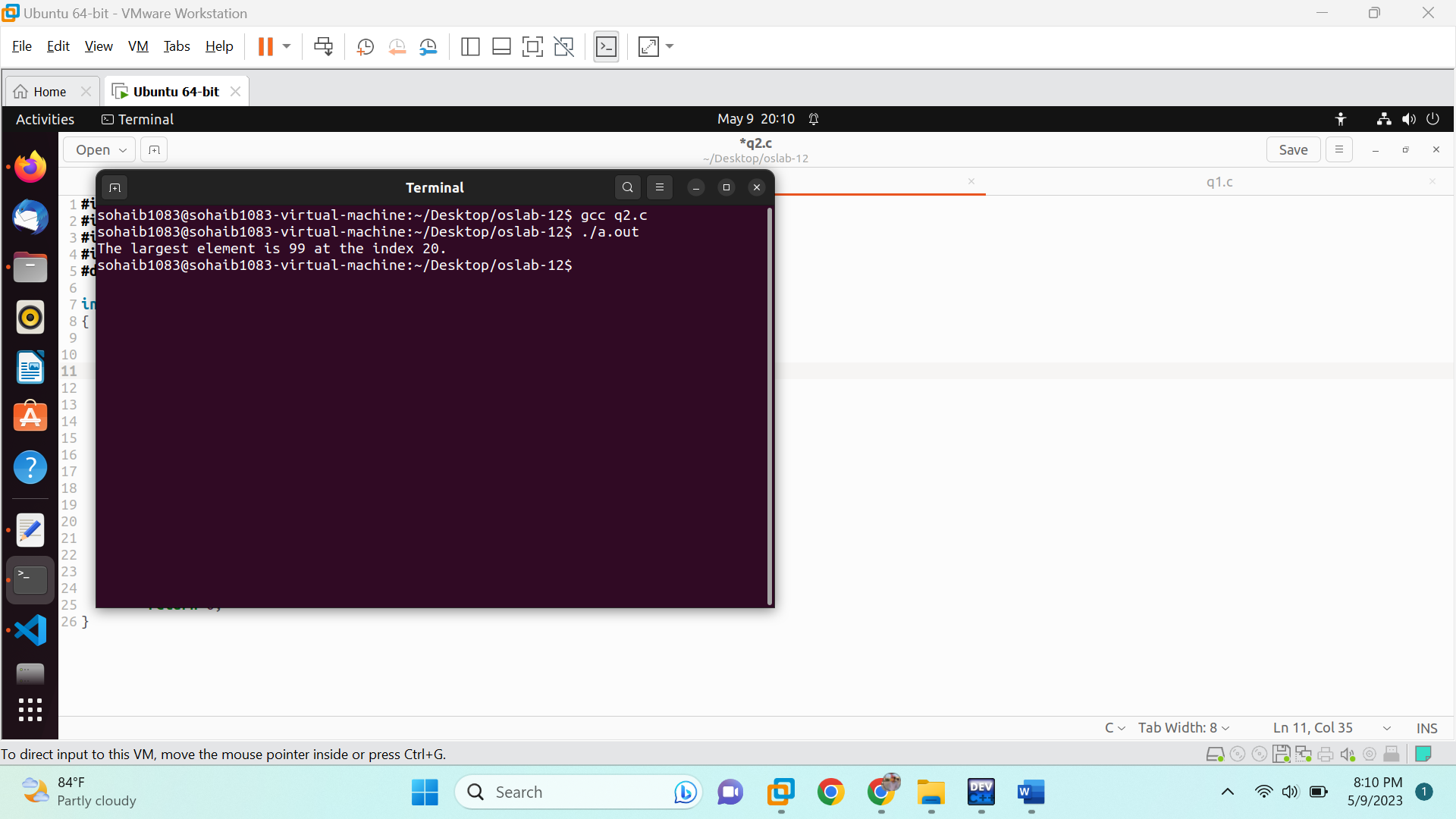
}

}

printf("The largest element is %d at the index %d.\n", max, mx\_index);

return 0;

}



Q3.

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <omp.h>

#define n 1000

int main (void){

int arr[n];

int i;

srand(time(0));

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

{

arr[i] = rand() % 100 + 1;

}

int temp, j;

#pragma omp parallel for shared(default) private(i,j)

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

{

for(j=i+1; j<n; j++)

{

#pragma omp critical

if(arr[j] < arr[i])

{

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

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

{

printf("%d ",arr[i]);

}

printf("\n");

return 0;

}

A screenshot of a computer

Description automatically generated