**Họ và Tên: Trần Thị Ngọc Diệp**

**MSSV: 1827005**

**Lớp: B2HK182 – Cấu Trúc Dữ Liệu và Giải Thuật**

**(TUT 1 & LAB 1)**

------------------------

**TUT 1**

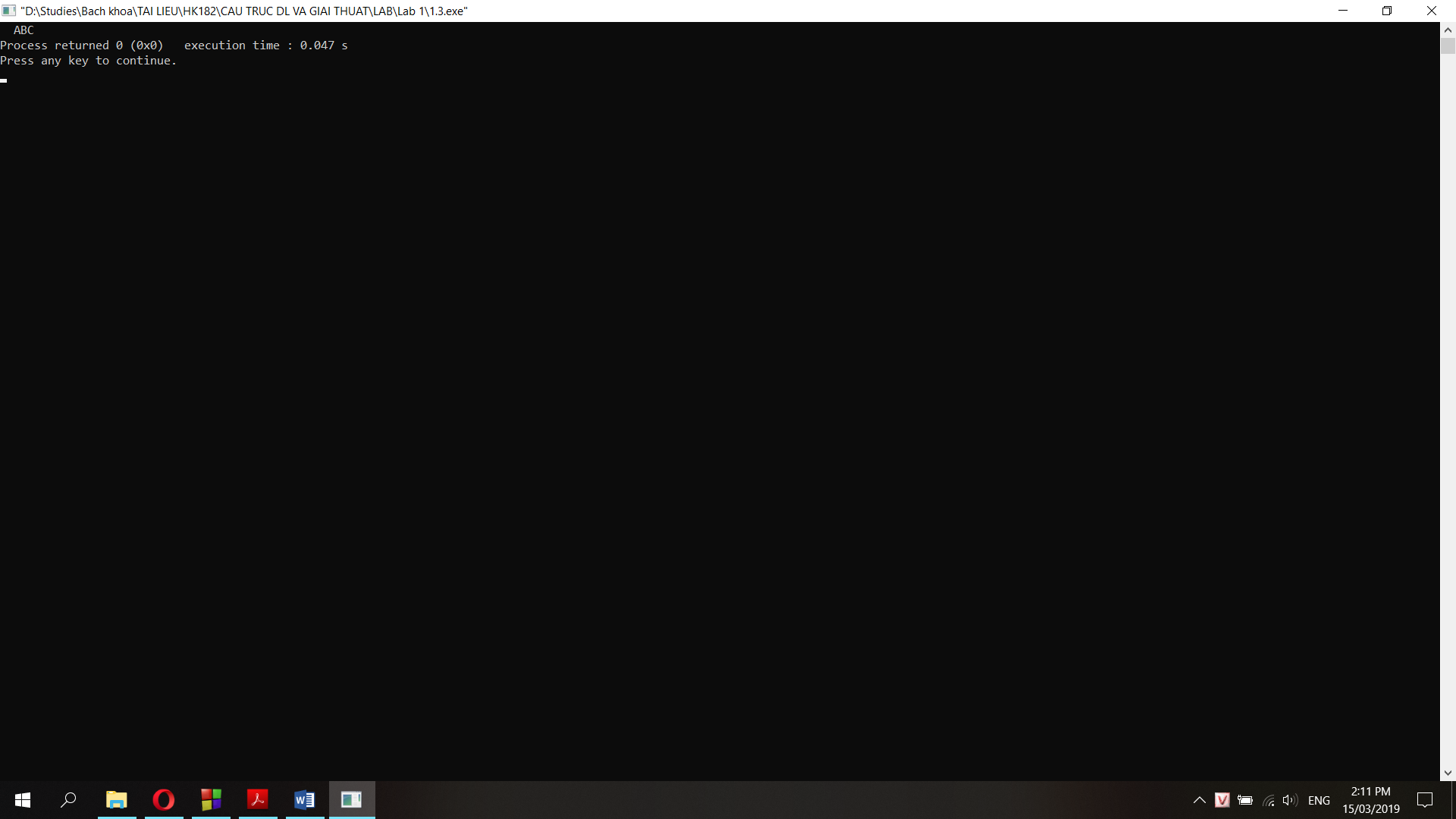
**Câu 1:**

char str[5] = "ABC";

cout << str[3] << str[4];

cout << str;

Kết quả của đoạn chương trình:



Chuỗi (string) str là một mảng có độ dài là 5, chứa các phần từ kiểu char. Nhập giá trị “ABC” cho chuỗi nghĩa là nhập giá trị cho phần tử từ 0 đến 2. Các vị trí còn lại mang giá trị NULL nên khi in ra sẽ in khoảng trống. Sau đấy lệnh cout<<str sẽ in ra toàn bộ chuỗi.

**Câu 2:**

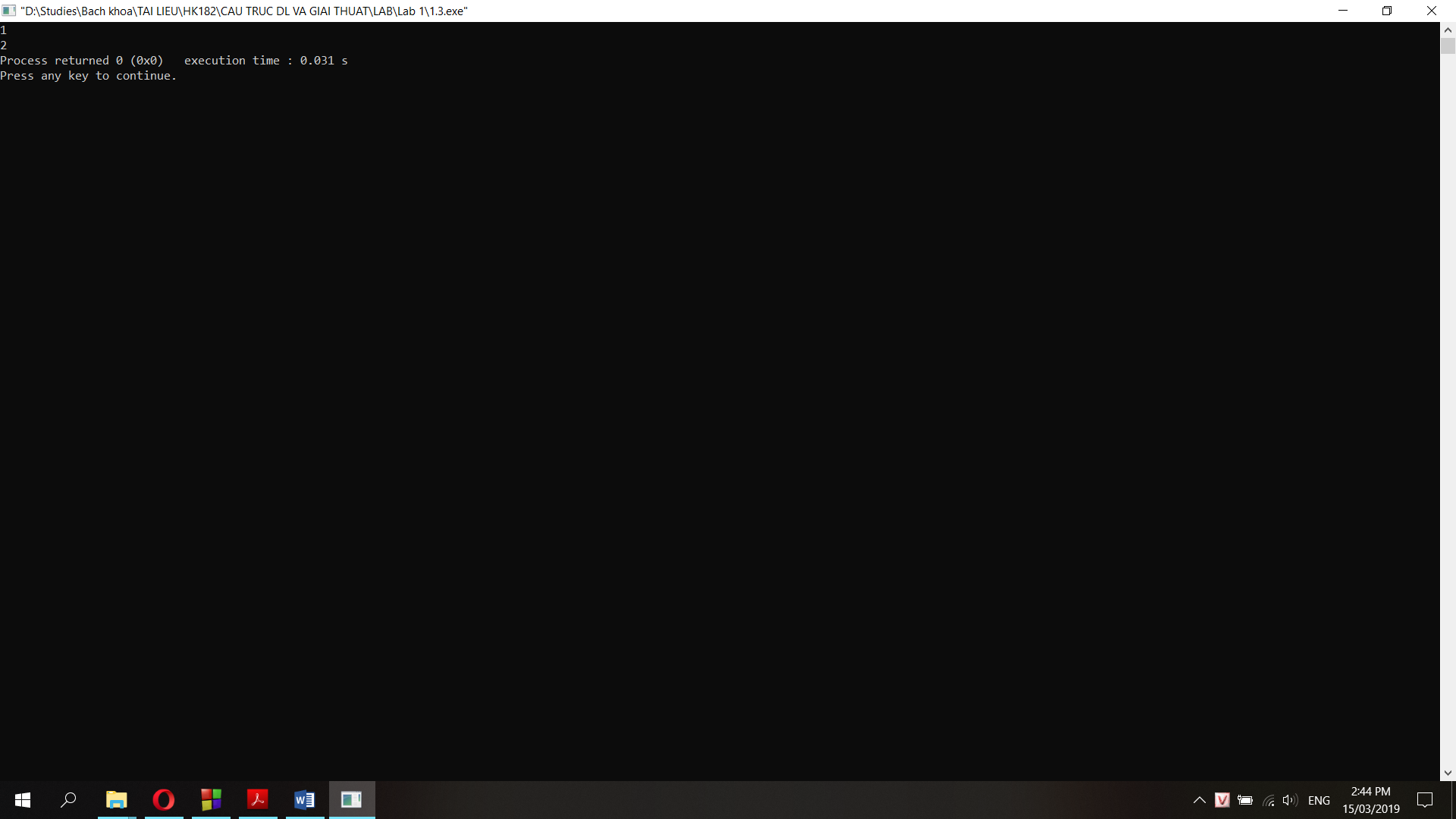
int a = 1, b = 2, c = 3;

int \*arr[3] = { &a, &b, &c };

cout << \*arr[(\*arr[2])-- - 3] << "\n";

cout << c;

Kết quả của đoạn chương trình:



\*arr[2] là giá trị của phần tử thứ 2 trong array chỉ đến, ở đây là giá trị của c (3). Toán tử -- là postfix nên sẽ thay đổi giá trị của c sau khi thực hiện lệnh cout.

Như vậy cout << \*arr[(\*arr[2])-- - 3]; đơn giản là cout<<\*arr[0]; in ra giá trị của a.

Giá trị của c lúc này đã bị giảm 1, nên khi in ra có giá trị là 2.

**Câu 3:**

a. int arr[10];

int \*ptr;

ptr = arr;

b. Không có gì xảy ra nếu delete một NULL pointer. Tuy nhiên, delete một pointer 2 lần sẽ dẫn đến kết quả không xác định được, có thể có hoặc không có lỗi xảy ra.

**Câu 4:**

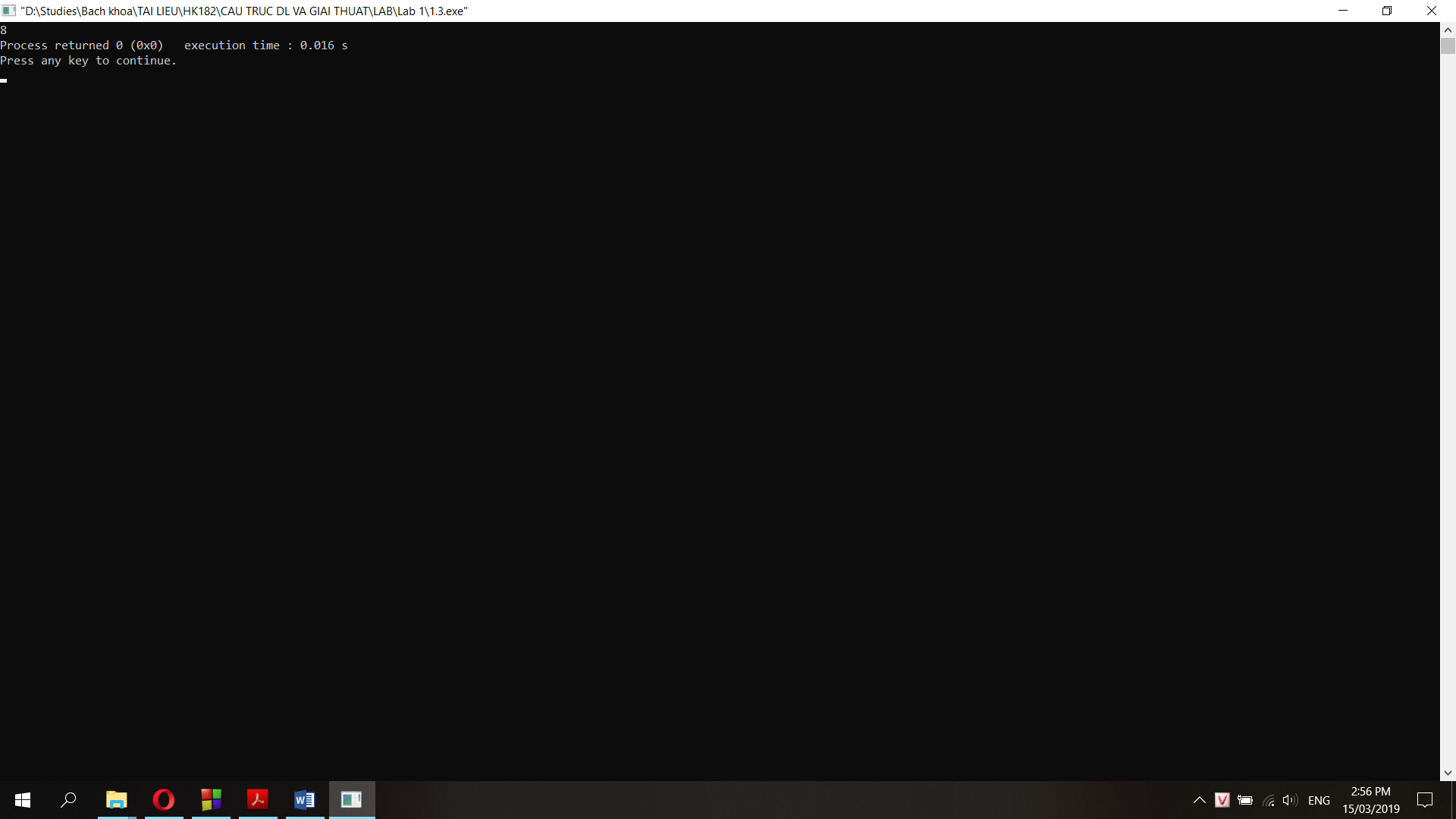
int fun(int x, int y) {

if (x == 0) return y;

return fun(x - 1, x + y);

}

Kết quả chương trình khi gọi hàm fun(3,2):



B1: Hàm gọi tiếp fun(2,5)

B2: Hàm gọi tiếp fun(1,7)

B3: Hàm gọi tiếp fun(0,8)

B4: Hàm chạy vào nhánh if vì x=0, hàm trả về giá trị của y.

**Câu 5:**

a. int count = 1;

for (; count <= 5; count++) {

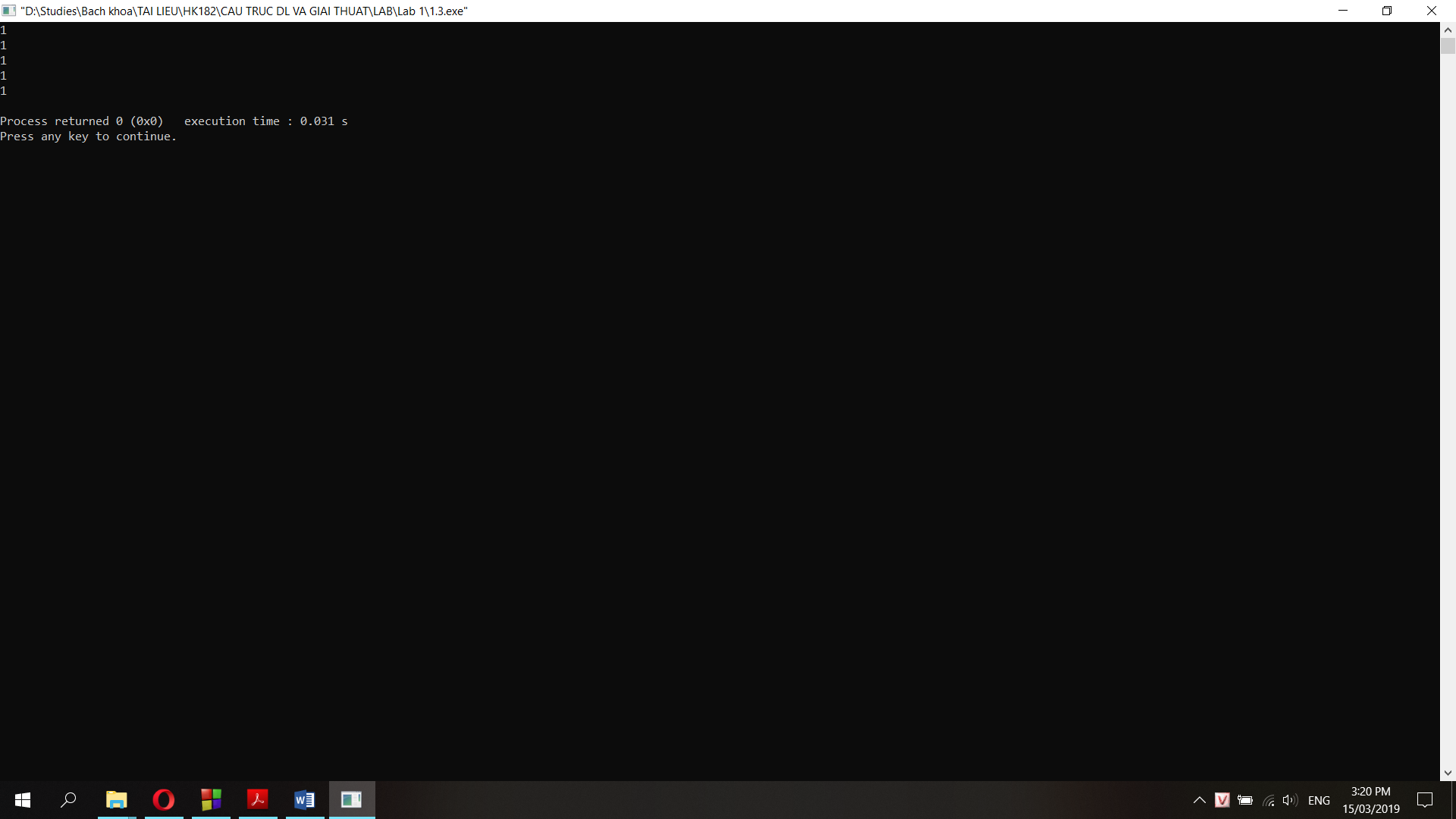
int count = 1;

cout << count << "\n";

}

return 0;

Kết quả vòng lặp for:



Do biến count được gọi và gán giá trị bên ngoài vòng lặp đồng thời phép toán count++ được gọi ngay sau đó sẽ cộng giá trị cho biến count này. Biến count được gọi lại trong vòng lặp và chỉ có giá trị cục bộ, phép toán count++ không ảnh hưởng đến giá trị biến này. Khi biến count ở ngoài vòng lặp tăng lên đến 5 thì dừng vòng lặp for.

b. int count = 1;

while (count <= 5) {

int count = 1;

cout << count << "\n";

count++;

}

return 0;

Vòng lặp while chạy vô hạn vì phép toán count++ tăng giá trị cho biến count trong vòng lặp. Biến count ngoài vòng lặp không bị ảnh hưởng đến giá trị và luôn bằng 1.

**Câu 6:**

void f(int x) {

x++;

return;

}

void g(int &x) {

x++;

return;

}

void h(const int &x) {

x++;

return;

}

Hàm f() không thay đổi giá trị của biến vì đây là cách truyền tham trị.

Hàm g() thay đổi giá trị của biến (tăng lên 1) vì đây là cách truyền tham chiếu.

Hàm h() bị lỗi vì sử dụng phép toán lên biến const.

**Câu 7:**

#include <iostream>

using namespace std;

class Test {

int x;

Test() { x = 5;}

};

int main() {

Test \*t = new Test();

cout << t->x;

}

Chương trình chạy bị lỗi vì biến thành viên x là private (mặc định) của class, không thể truy cập x trong hàm main được. Sửa class thành:

class Test {

public:

int x;

Test() { x = 5;} };

**Câu 8:**

#include <iostream>

using namespace std;

int main() {

int \*\*p = new int\*[5];

int \*q = new int[5];

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

q[i] = i;

}

p[1] = q;

delete p; //1

delete q; //2

delete [] q; //3

}

Giá trị của q[2] và p[1][2] lần lượt là:

* Sau khi chạy câu lệnh delete 1: 2, địa chỉ bất kỳ vì p đã bị xóa nên p[1][2] không có quy luật nào tùy vào mỗi hệ thống khác nhau.
* Sau khi chạy câu lệnh delete 2: 2, địa chỉ bất kỳ. Mặc dù q vẫn còn giá trị, nhưng giá trị in ra không tuân theo quy luật nào.
* Sau khi chạy câu lệnh delete 2: error

**Câu 9:**

#include <iostream>

using namespace std;

int fun(int n, int \*f\_p) {

int t, f;

if (n <= 1){

\*f\_p = 1;

return 1;

}

t = fun(n- 1,f\_p);

f = t+ \* f\_p;

\*f\_p = t;

return f;

}

int main() {

int x = 15;

cout<<fun(5, &x);

}

**Kết quả in ra: giá trị của hàm fun(5,&x) là 8.**

Hàm hồi quy gọi 5 lần, fun(n,f\_p) cho đến khi n = 1, return 1. Giá trị x truyền vào là tham chiếu, nên giá trị của x sau mỗi lần gọi hàm sẽ bị thay đổi.

Lần gọi hàm thứ 5 (n=1): f\_p = 1, return 1 vào giá trị t.

Lần gọi hàm thứ 4 (n=2): t = 1; f\_p = 1; f = 2; f\_p = t (=1); return f = 2 vào giá trị t.

Lần gọi hàm thứ 3 (n=3): t = 2; f\_p = 1; f = 2+1 = 3; f\_p = t (=2); return f = 3 vào giá trị t.

Lần gọi hàm thứ 2 (n=4): t = 3; f\_p = 2; f = 3+2 = 6; f\_p = t (=3); return f = 5 vào giá trị t.

Lần gọi hàm thứ 1 (n=5): t = 5; f\_p = 3; f = 5+3 = 8; f\_p = t (=5); return f = 8.

Vậy hàm return giá trị 8 khi gọi fun(5, &x)

**LAB 1**

**Câu 1:**

#include <iostream>

using namespace std;

**bool isLeapYear(int y)**{

if (((y%4==0)&&(y%100!=0))||(y%400==0)){

return true;

}

else return false;

}

int main(){

int y;

cout<<"Nhap nam: ";

cin>>y;

if (isLeapYear(y)==1){

cout<<"Nam Nhuan";

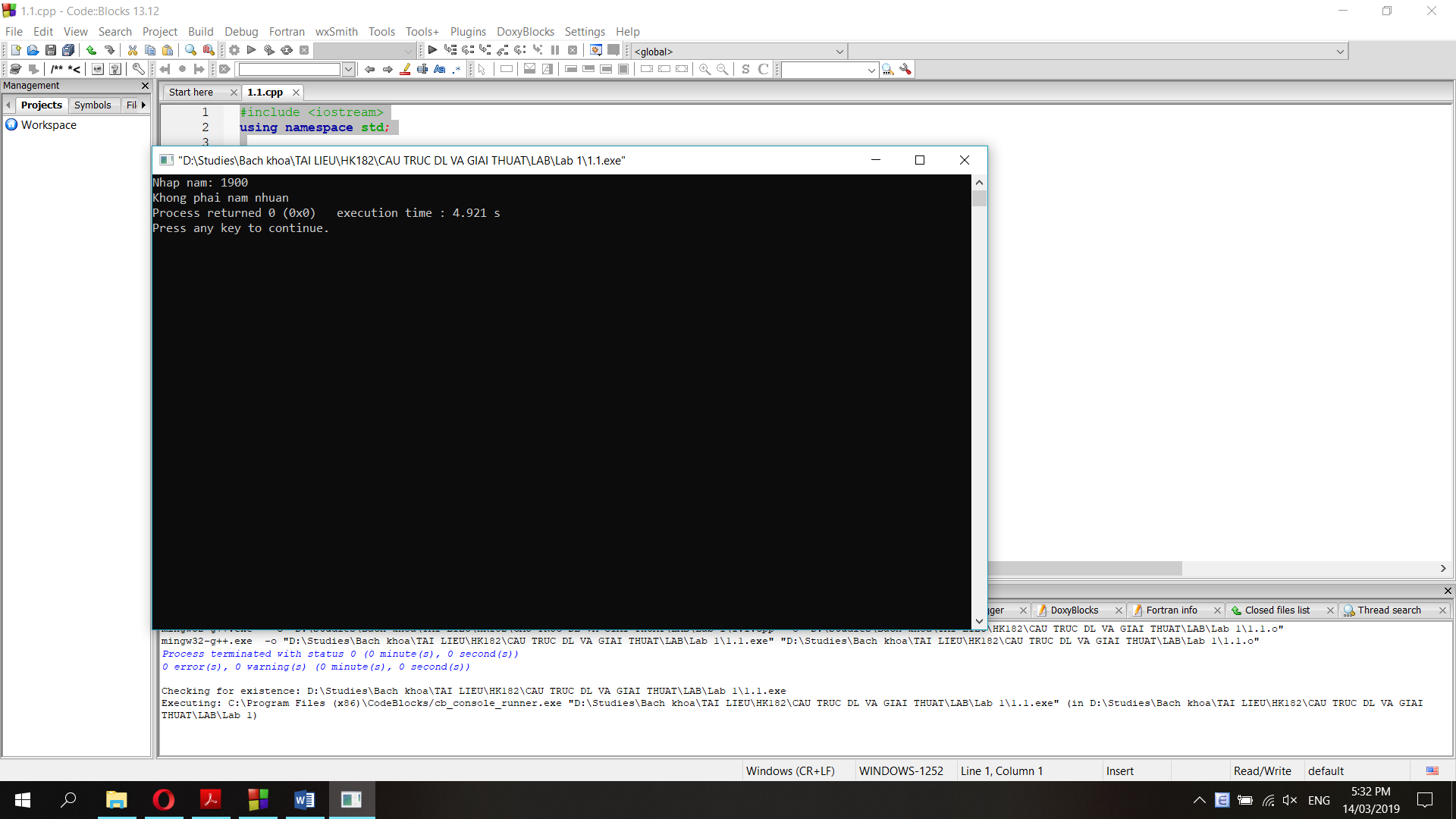
}

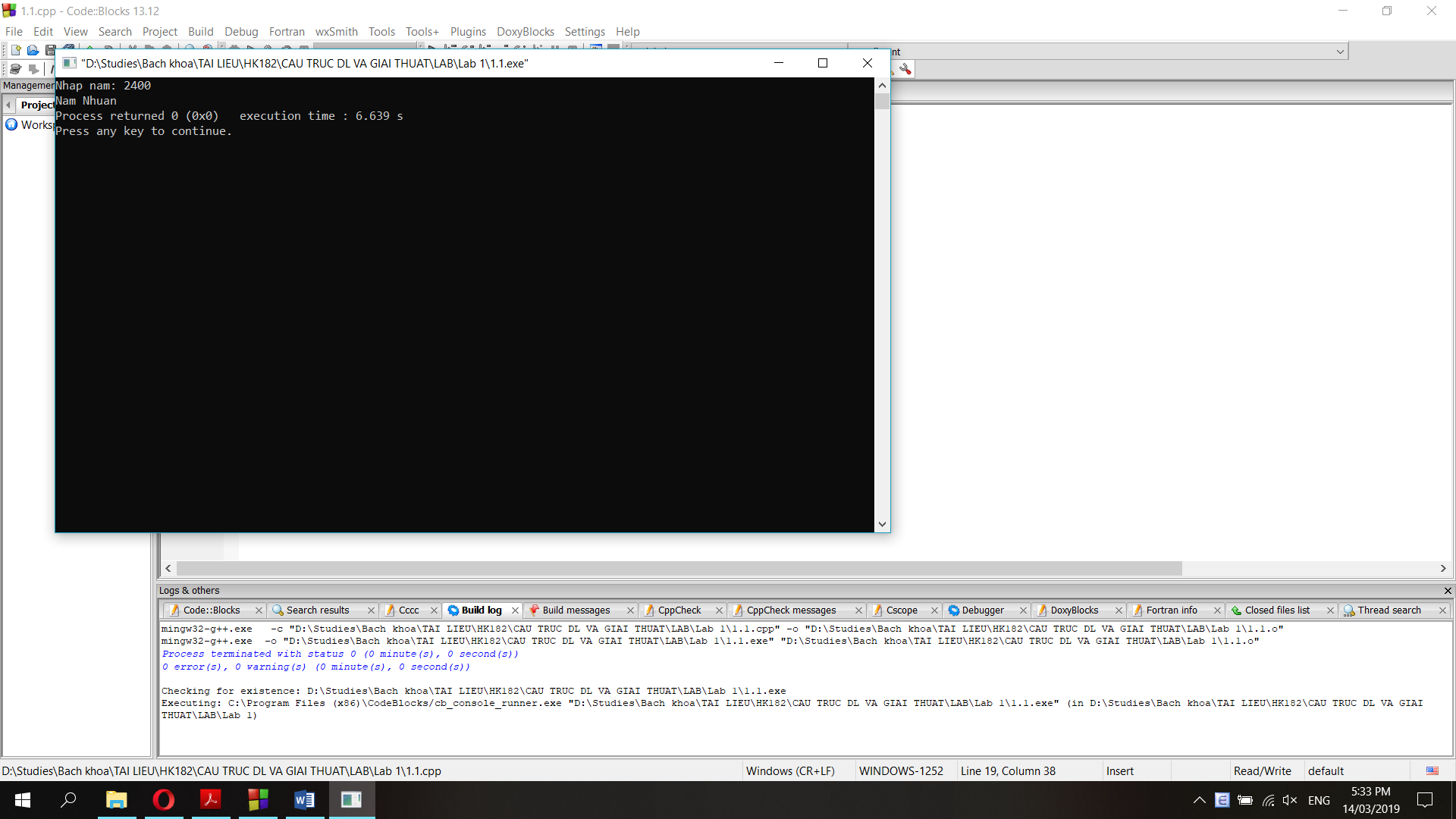
else {

cout<<"Khong phai nam nhuan";

}

}





**Câu 2:**

**a.**

#include<stdlib.h>

#include<stdio.h>

#include <limits.h>

#include<iostream>

#include<time.h>

#include <cstdlib>

using namespace std;

#define Max 20

void mul2Matrix(int R1, int C1, int R2, int C2, int matrixA[][Max], int matrixB[][Max], int matrixC[][Max]){

int i, j, k;

//Gan gia tri cho cac phan tu cua ma tran C = 0

for (i=0; i<R1; i++){

for (j=0; j<C2; j++){

matrixC[i][j] = 0;

}

}

//in cac phan tu cua ma tran A va B

printf("\nCac phan tu cua ma tran A: \n\n");

for (i=0; i<R1; i++){

for (j=0; j<C1; j++){

if (matrixA[i][j]>=0) printf(" %d\t", matrixA[i][j]);

else printf("%d\t", matrixA[i][j]);

}

printf("\n");

}

printf("\nCac phan tu cua ma tran B: \n\n");

for (i=0; i<R2; i++){

for (j=0; j<C2; j++){

if (matrixB[i][j]>=0) printf(" %d\t", matrixB[i][j]);

else printf("%d\t", matrixB[i][j]);

}

printf("\n");

}

//Nhan 2 ma tran A va B, luu ket qua vao C

for (i=0; i<R1; i++){

for (j=0; j<C2; j++){

int sum = 0;

for (k=0; k<C1; k++)

sum += matrixA[i][k] \* matrixB[k][j];

matrixC[i][j] = sum;

}

}

//In ma tran ket qua C

printf("\nCac phan tu cua ma tran ket qua C: \n\n");

for (i=0; i<R1; i++){

for (j=0; j<C2; j++){

if (matrixC[i][j]>=0) printf(" %d\t", matrixC[i][j]);

else printf("%d\t", matrixC[i][j]);

}

printf("\n");

}

}

int main () {

srand(time(0));

int Matrix\_A[Max][Max], Matrix\_B[Max][Max];

int r1, c1, r2, c2;

printf("Vui long nhap so hang va so cot cua ma tran A: ");

scanf("%d %d", &r1, &c1);

printf("Vui long nhap so hang va so cot cua ma tran B: ");

scanf("%d %d", &r2, &c2);

while (c1!=r2){

printf("Error: Phai thoa man dieu kien C1 = R2\n");

printf("Vui long nhap so hang va so cot cua ma tran A: ");

scanf("%d %d", &r1, &c1);

printf("Vui long nhap so hang va so cot cua ma tran B: ");

scanf("%d %d", &r2, &c2);

}

//Cap phat bo nho ma tran C trong stack

int Matrix\_C[Max][Max];

//Random cac gia tri cua 2 ma tran A va B

for (int i=0; i<r1; i++){

for (int j=0; j<c1; j++){

Matrix\_A[i][j] = -40 + (rand()/(RAND\_MAX/91));

}

}

for (int i=0; i<r2; i++){

for (int j=0; j<c2; j++){

Matrix\_B[i][j] = -40 + (rand()/(RAND\_MAX/91));

}

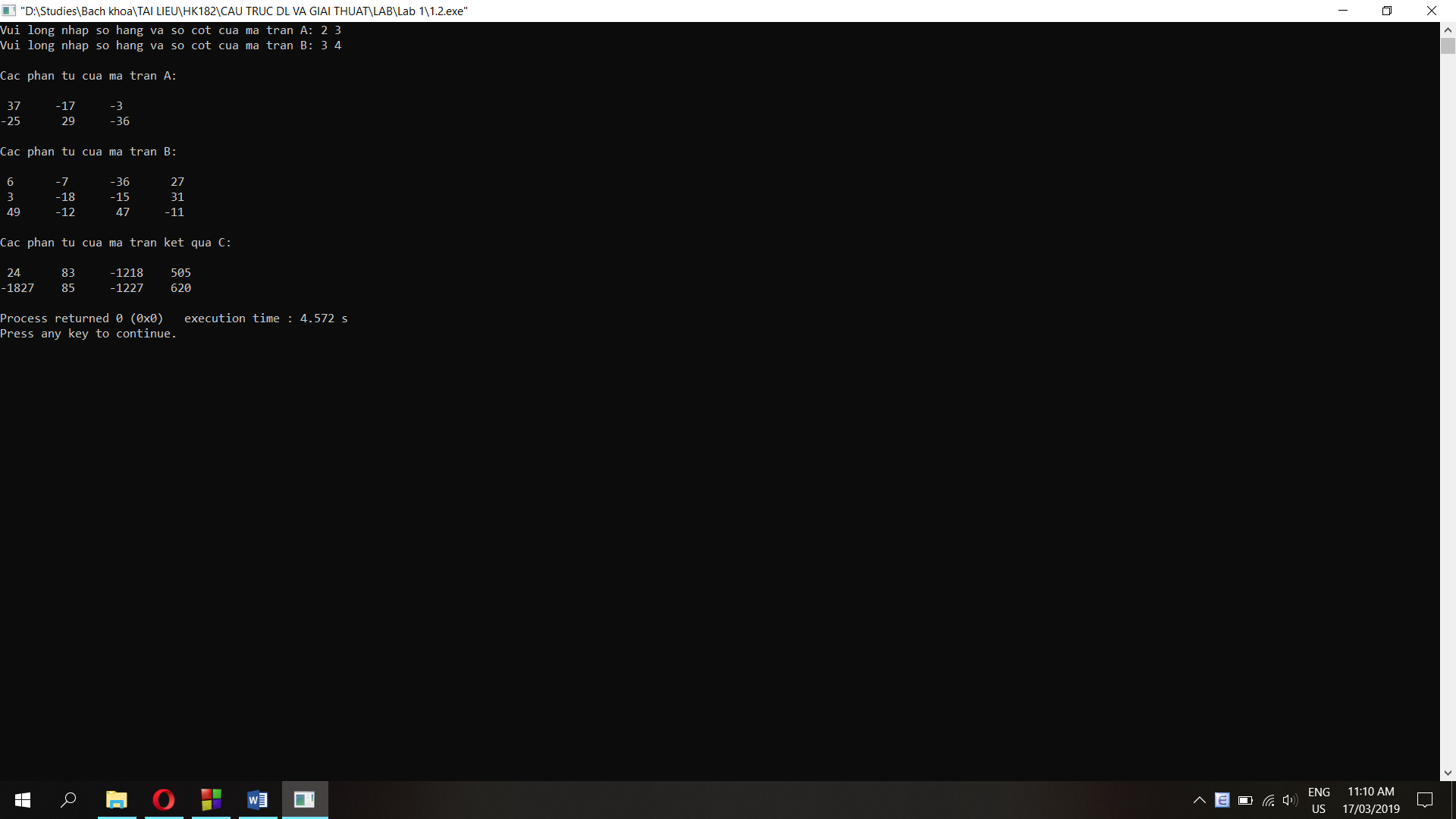
}

//Goi ham nhan ma tran

mul2Matrix(r1, c1, r2, c2, Matrix\_A, Matrix\_B, Matrix\_C);

return 0;

}



**b.**

#include<stdlib.h>

#include<stdio.h>

#include <limits.h>

#include<iostream>

#include<time.h>

#include <cstdlib>

using namespace std;

#define Max 20

void mul2Matrix(int R1, int C1, int R2, int C2, int matrixA[][Max], int matrixB[][Max]){

int i, j, k;

//Cap phat ma tran C trong bo nho Heap

int \*\*Matrix\_C = new int\*[R1];

for (int i=0; i<R1; i++){

Matrix\_C[i] = new int[C2];

}

//in cac phan tu cua ma tran A va B

printf("\nCac phan tu cua ma tran A: \n\n");

for (i=0; i<R1; i++){

for (j=0; j<C1; j++){

if (matrixA[i][j]>=0) printf(" %d\t", matrixA[i][j]);

else printf("%d\t", matrixA[i][j]);

}

printf("\n");

}

printf("\nCac phan tu cua ma tran B: \n\n");

for (i=0; i<R2; i++){

for (j=0; j<C2; j++){

if (matrixB[i][j]>=0) printf(" %d\t", matrixB[i][j]);

else printf("%d\t", matrixB[i][j]);

}

printf("\n");

}

//Nhan 2 ma tran A va B, luu ket qua vao C

for (i=0; i<R1; i++){

for (j=0; j<C2; j++){

int sum = 0;

for (k=0; k<C1; k++)

sum += matrixA[i][k] \* matrixB[k][j];

Matrix\_C[i][j] = sum;

}

}

//In ma tran ket qua C

printf("\nCac phan tu cua ma tran ket qua C: \n\n");

for (i=0; i<R1; i++){

for (j=0; j<C2; j++){

if (Matrix\_C[i][j]>=0) printf(" %d\t", Matrix\_C[i][j]);

else printf("%d\t", Matrix\_C[i][j]);

}

printf("\n");

}

}

int main () {

srand(time(0));

int Matrix\_A[Max][Max], Matrix\_B[Max][Max];

int r1, c1, r2, c2;

printf("Vui long nhap so hang va so cot cua ma tran A: ");

scanf("%d %d", &r1, &c1);

printf("Vui long nhap so hang va so cot cua ma tran B: ");

scanf("%d %d", &r2, &c2);

while (c1!=r2){

printf("Error: Phai thoa man dieu kien C1 = R2\n");

printf("Vui long nhap so hang va so cot cua ma tran A: ");

scanf("%d %d", &r1, &c1);

printf("Vui long nhap so hang va so cot cua ma tran B: ");

scanf("%d %d", &r2, &c2);

}

//Cap phat bo nho ma tran C trong stack

int Matrix\_C[Max][Max];

//Random cac gia tri cua 2 ma tran A va B

for (int i=0; i<r1; i++){

for (int j=0; j<c1; j++){

Matrix\_A[i][j] = -40 + (rand()/(RAND\_MAX/91));

}

}

for (int i=0; i<r2; i++){

for (int j=0; j<c2; j++){

Matrix\_B[i][j] = -40 + (rand()/(RAND\_MAX/91));

}

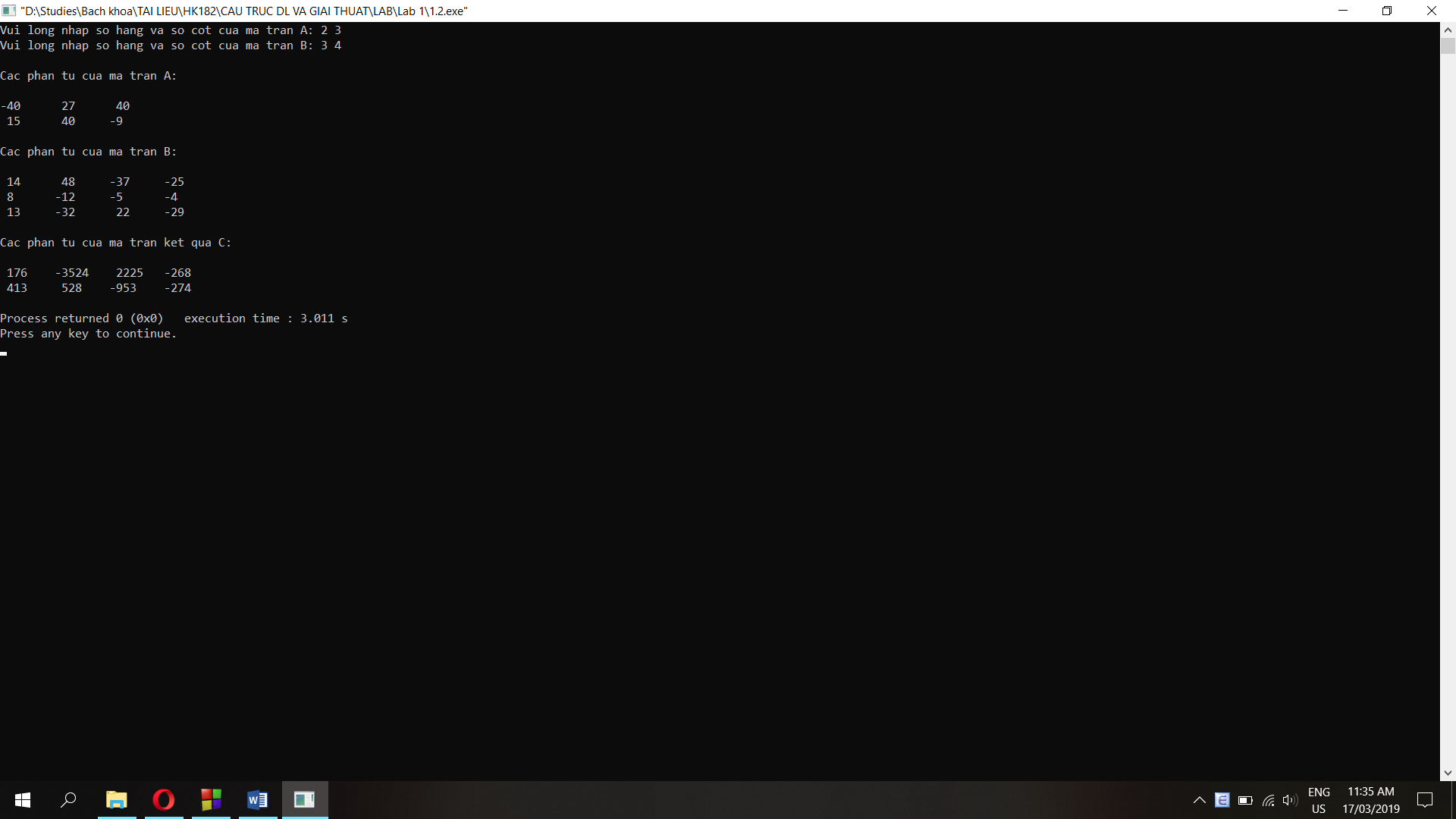
}

//Goi ham nhan ma tran

mul2Matrix(r1, c1, r2, c2, Matrix\_A, Matrix\_B);

return 0;

}



**Câu 3:**

**a.**

int pow(int a, int b){

int t = 1;

while (b>0){

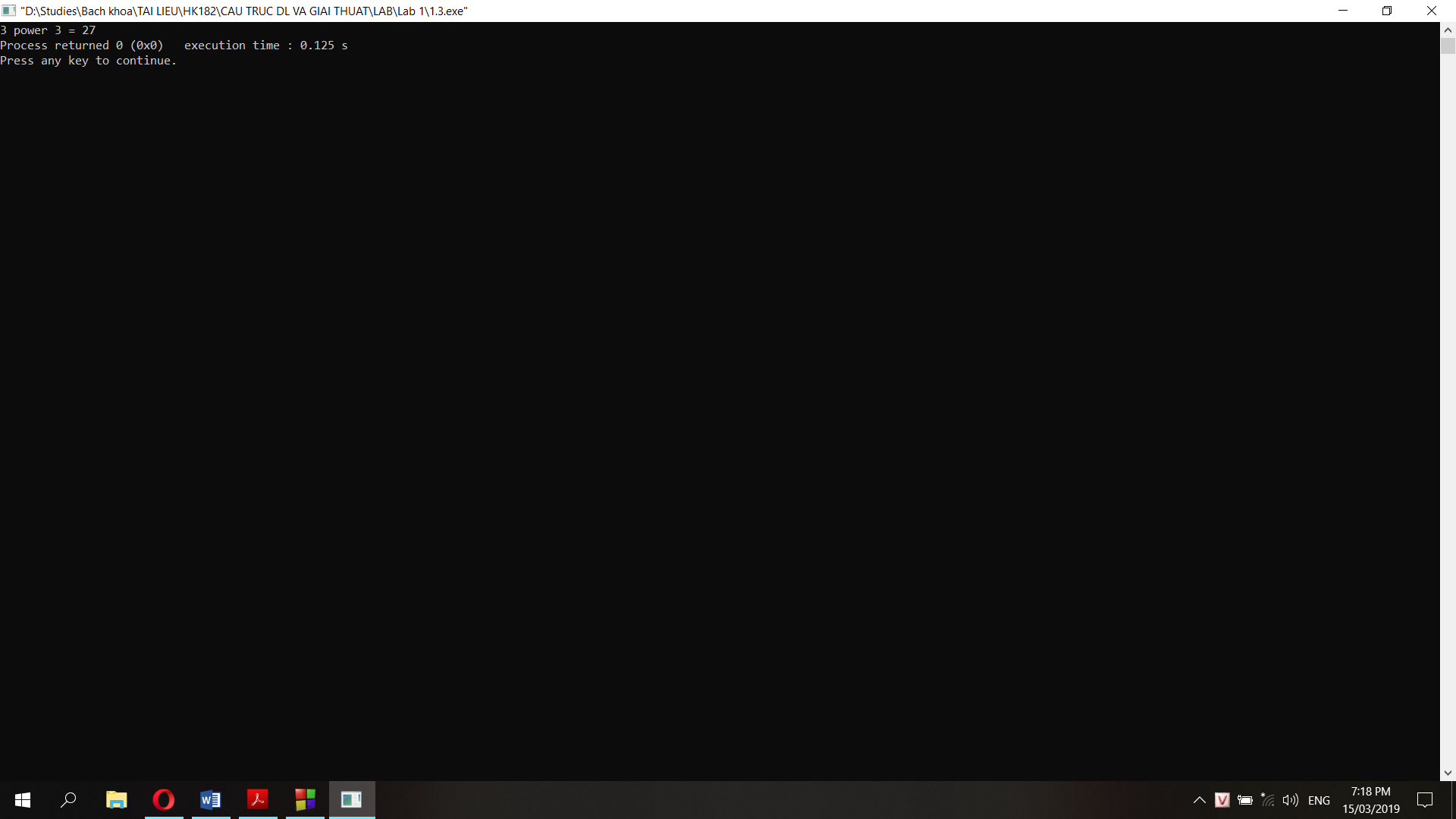
t = t\*a;

b--;

}

return t;

}



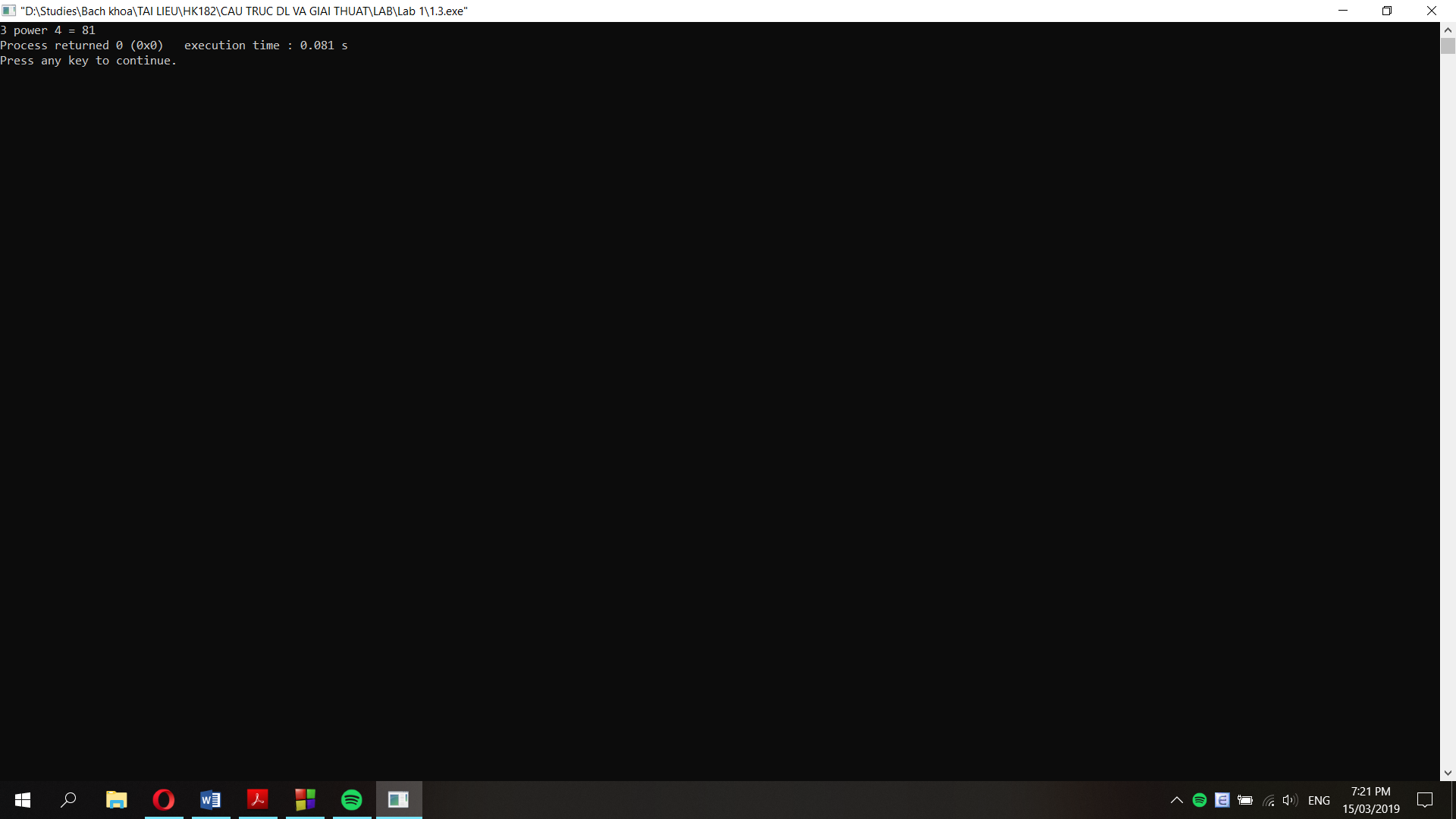
**b.**

int pow(int a, int b){

if (b>0) return a\*pow(a,b-1);

else return 1;

}



**Câu 4:**

**a.**

int compare\_str(char a[], char b[]){

int i;

//Tim kich thuoc that cua chuoi

int size\_A, size\_B;

for (i=0; i<sizeof(a); i++){

if (a[i]=='\0') break;

}

size\_A = i+1;

for (i=0; i<sizeof(b); i++){

if (b[i]=='\0') break;

}

size\_B = i+1;

//Gan flag = F: 2 chuoi giong nhau

bool Flag = false;

//Truong hop 2 chuoi co do dai khac nhau

if (size\_A != size\_B){

Flag = true;

if (size\_A<size\_B){

for (i=0; i<size\_A; i++){

if (a[i]!=b[i]){

return (a[i]-b[i]);

}

}

return b[i];

}

else{

for (i=0; i<size\_B; i++){

if (a[i]!=b[i]){

return (a[i]-b[i]);

}

}

return a[i];

}

}

else{

for (i=0; i<size\_A; i++){

if (a[i] != b[i]){

Flag = true;

return (a[i]-b[i]);

}

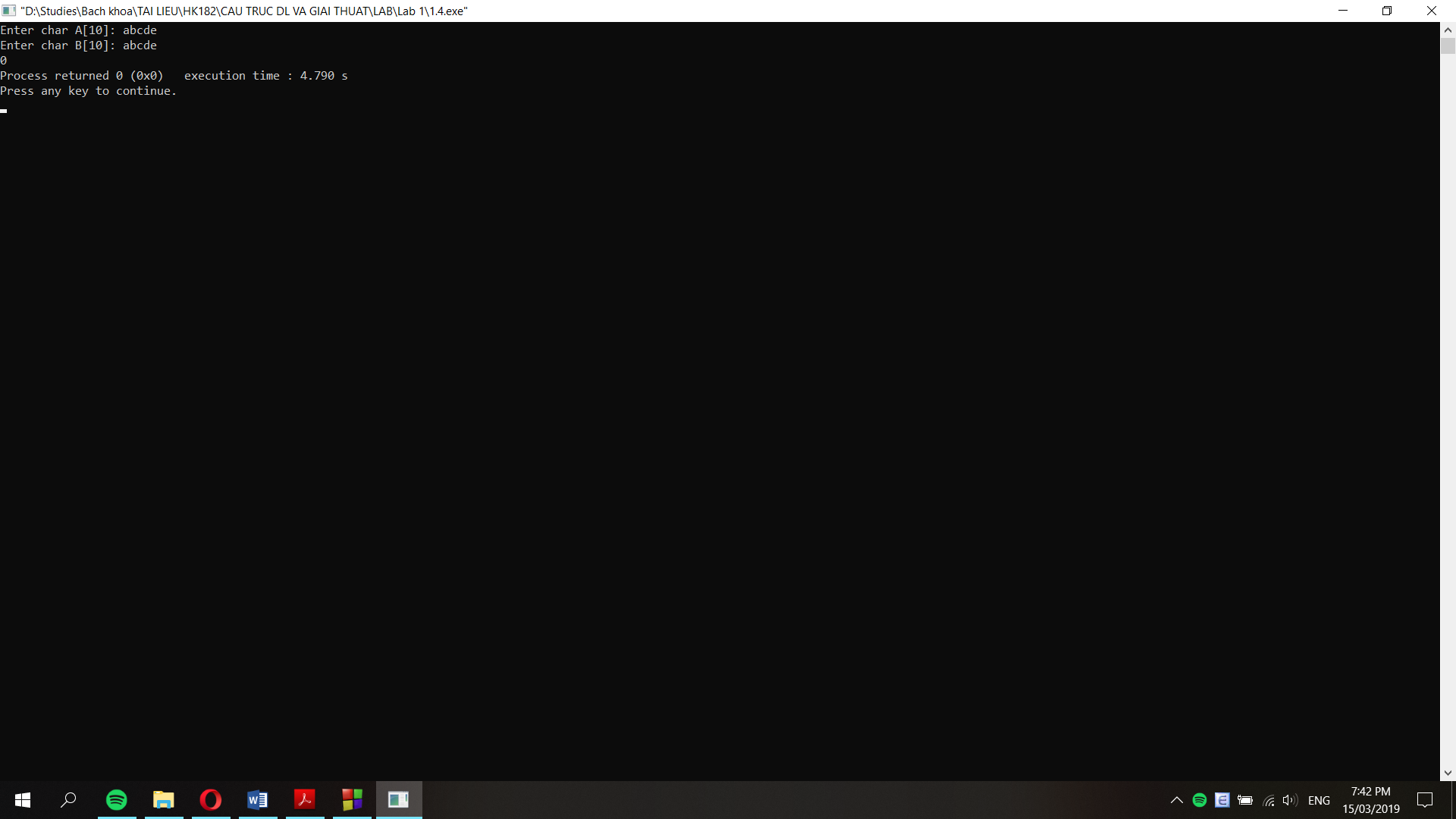
}

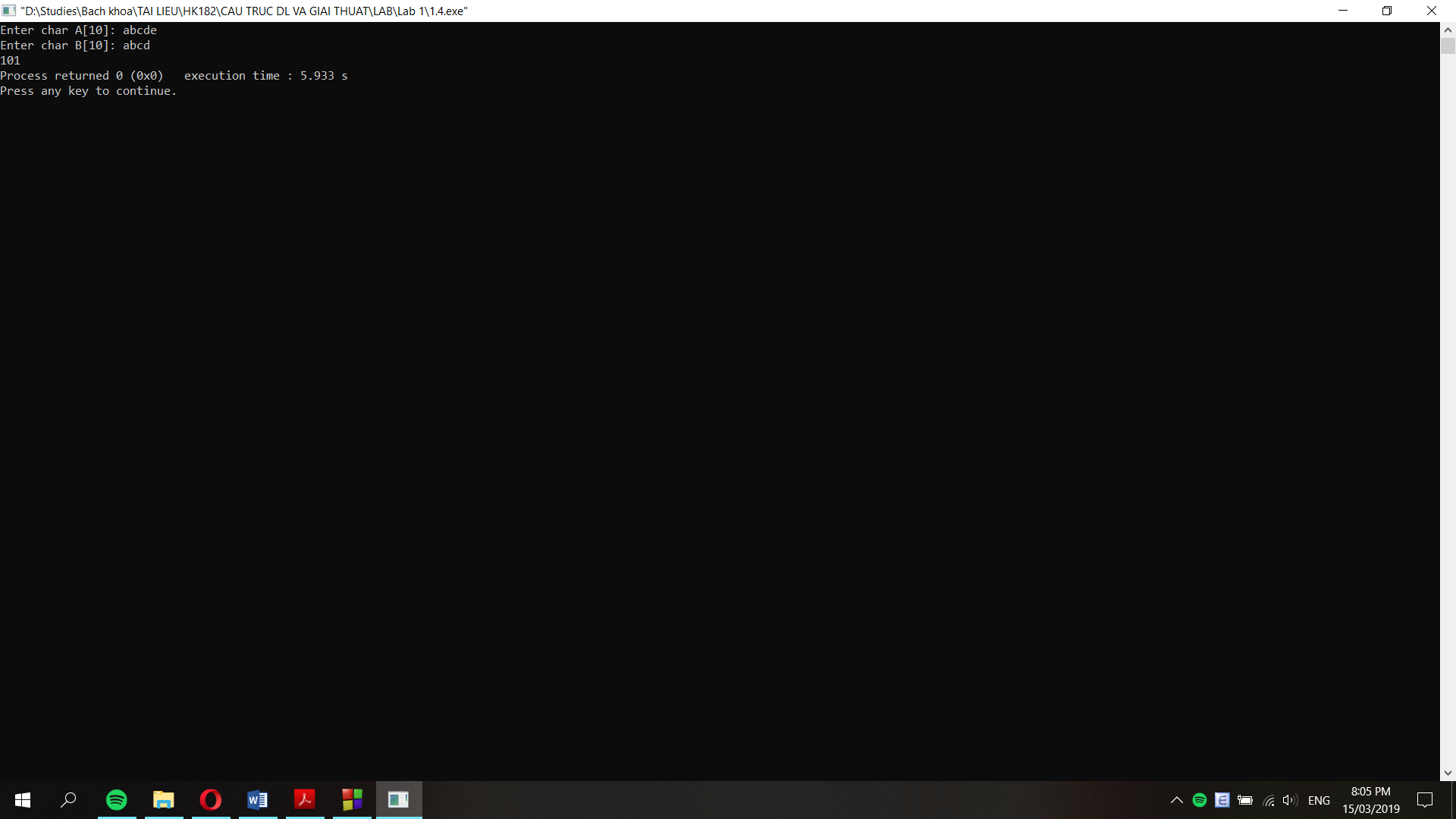
}

if (Flag == false){

return 0;

}

}



**b.**

#include <iostream>

#include <stdbool.h>

#include <string>

#include <strstream>

using namespace std;

#define Max 20

char \*compare\_str(char a[], char b[]){

char\* p;

int i, j;

//Tim kich thuoc that cua chuoi

int size\_A, size\_B;

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

if (a[i]=='\0') break;

}

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

if (b[j]=='\0') break;

}

size\_B = j;

//cout<<size\_A<<" "<<size\_;

//Gan flag = F: 2 chuoi giong nhau

bool Flag = false;

//Truong hop 2 chuoi co do dai khac nhau

if (size\_A != size\_B){

Flag = true;

if (size\_A<size\_B){

for (i=0; i<size\_A; i++){

if (b[i]!=a[i]){

p = &b[i];

return p;

}

}

p = &b[i];

return p;

}

else{

for (i=0; i<size\_B; i++){

if (a[i]!=b[i]){

p = &a[i];

return p;

}

}

p = &a[i];

return p;

}

}

else{

for (i=0; i<size\_A; i++){

if (a[i] != b[i]){

Flag = true;

p = &a[i];

return p;

}

}

}

if (Flag == false){

return 0;

}

}

int main(){

char A[Max], B[Max];

cout<<"Enter char A[Max]: ";

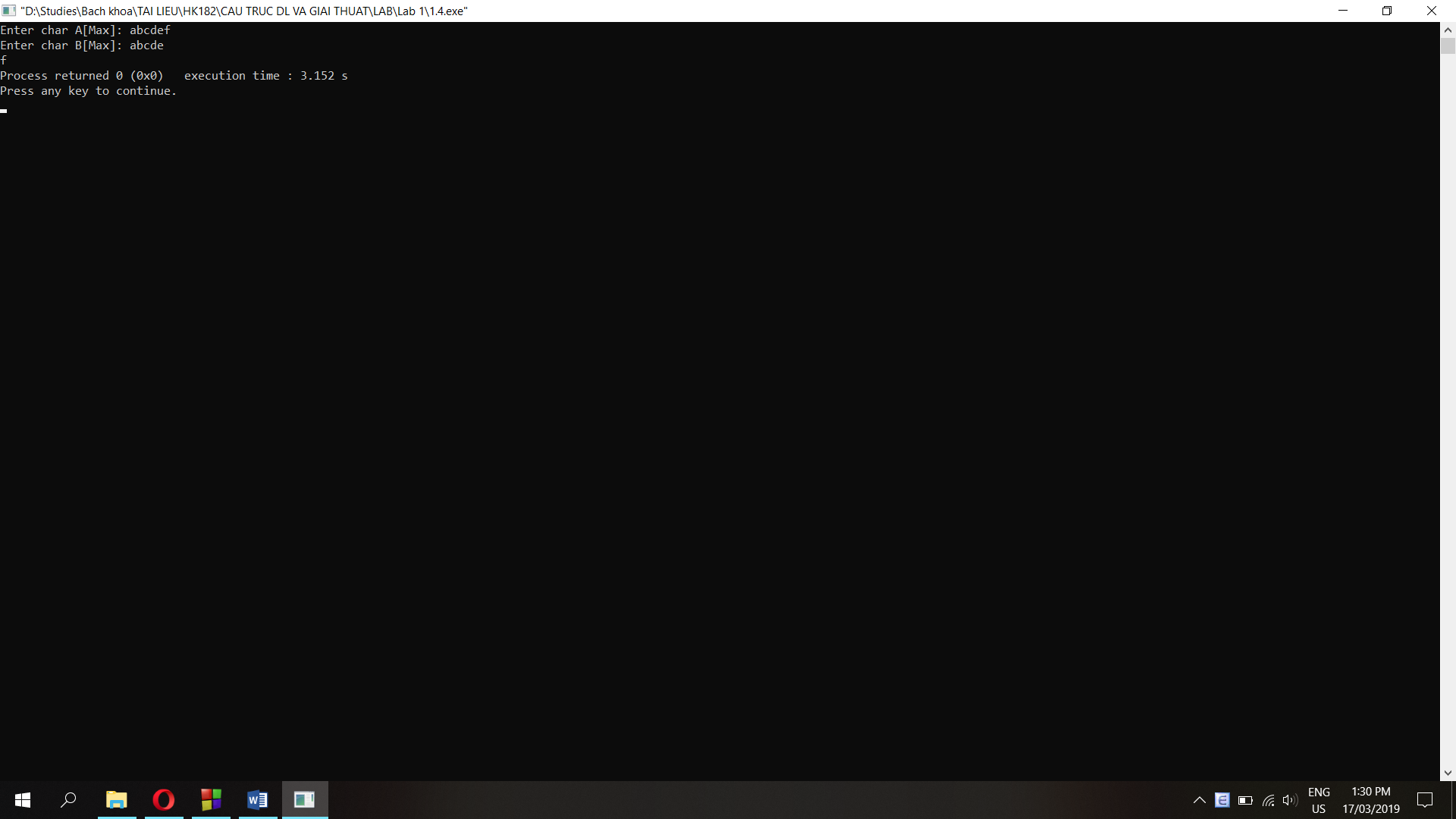
cin.getline(A, Max);

cout<<"Enter char B[Max]: ";

cin.getline(B, Max);

cout<<\*compare\_str(A, B);

}



**c.**

#include <iostream>

#include <stdbool.h>

#include <string>

#include <strstream>

using namespace std;

#define Max 20

char \*compare\_str(char a[], char b[]){

char\* p;

int i, j;

//Tim kich thuoc that cua chuoi

int size\_A, size\_B;

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

if (a[i]=='\0') break;

}

size\_A = i;

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

if (b[j]=='\0') break;

}

size\_B = j;

//Gan flag = F: chuoi nay khong phai la substring cua chuoi kia

bool Flag = false;

//Truong hop 2 chuoi co do dai khac nhau

if (size\_A != size\_B){

if (size\_A<size\_B){

for (i=0; i<size\_A; i++){

if (b[i]==a[i]){

for (j=i+1; j<size\_A; j++){

if (b[j]!=a[j]) break;

}

}

if (j==size\_A){

Flag = true;

p = &a[i];

return p;

}

}

}

else{

for (i=0; i<size\_B; i++){

if (a[i]==b[i]){

for (j=i+1; j<size\_B; j++){

if (a[j]!=b[j]) break;

}

}

if (j==size\_B){

Flag = true;

p = &b[i];

return p;

}

}

}

}

else{

for (i=0; i<size\_A; i++){

if (a[i] != b[i]){

Flag = false;

}

}

}

if (Flag == false){

return 0;

}

}

int main(){

char A[10], B[10];

cout<<"Enter char A[10]: ";

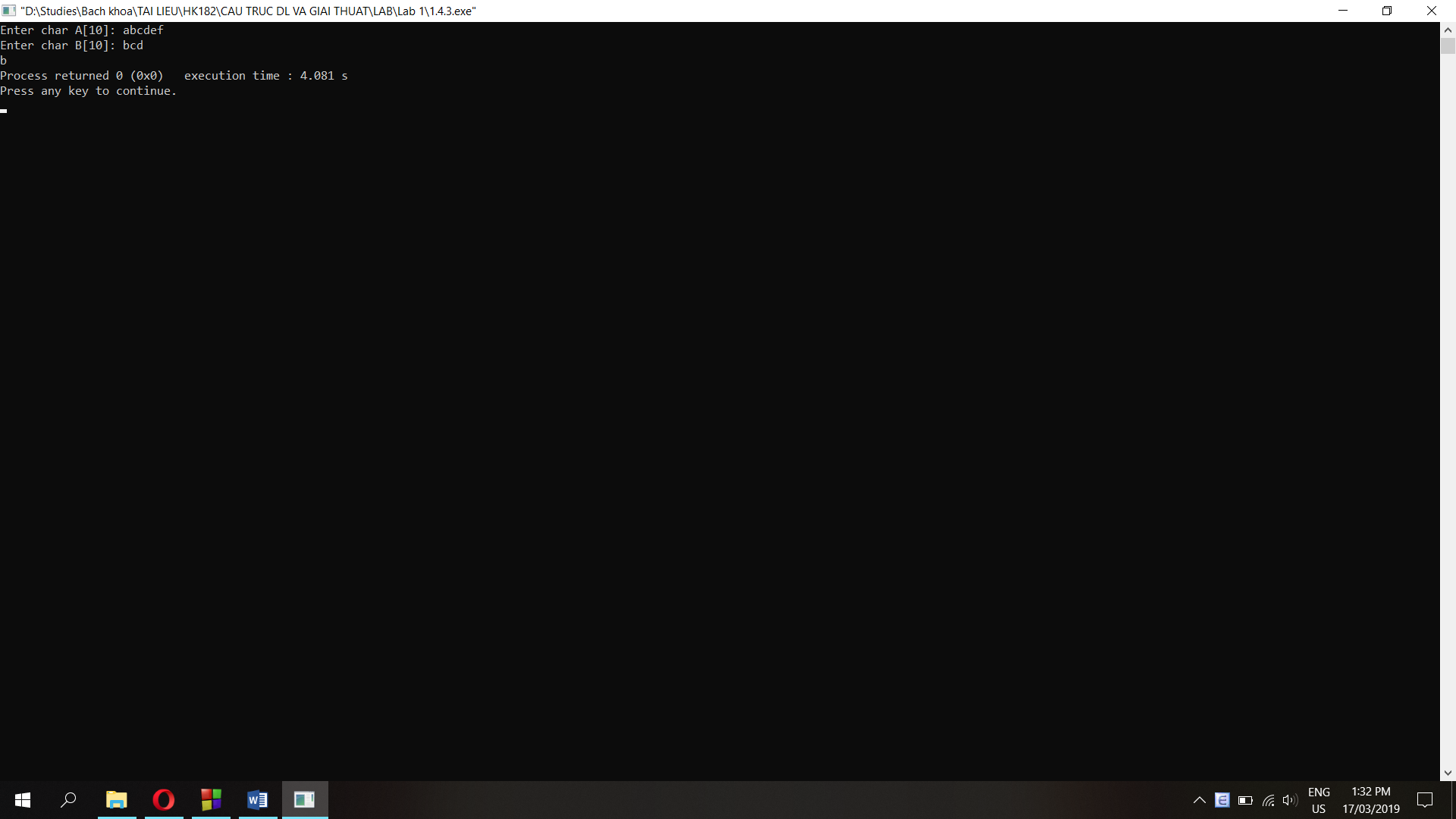
cin.getline(A, 10);

cout<<"Enter char B[10]: ";

cin.getline(B, 10);

cout<<\*compare\_str(A, B);

}



**Câu 5:**

#include <iostream>

#include <stdbool.h>

#include <string.h>

#include <strstream>

using namespace std;

class candidates{

private:

float math, physic, chemistry;

public:

int id;

char \*name;

candidates(int id, char\* name, float math, float physic, float chemistry);

~candidates(){

delete [] name;

}

float total\_grade();

void show\_info(){ cout<<id<<" "<<name<<" "<<math<<" "<<physic<<" "<<chemistry;}

};

candidates::candidates(int ID, char\* NAME, float MATH, float PHYSIC, float CHEMISTRY): id(ID), math(MATH), physic(PHYSIC), chemistry(CHEMISTRY){

name = new char[50];

strcpy(name, NAME);

}

float candidates::total\_grade(){

return math+physic+chemistry;

}

int main() {

candidates A(1412,"Nguyen Van A", 8.8, 9.9, 7.7);

A.show\_info();

cout<<"\nTotal grade: "<<A.total\_grade();

}

