**Họ và Tên: Trần Thị Ngọc Diệp**

**MSSV: 1827005**

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

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

**TUT 3**

**Question 1:**

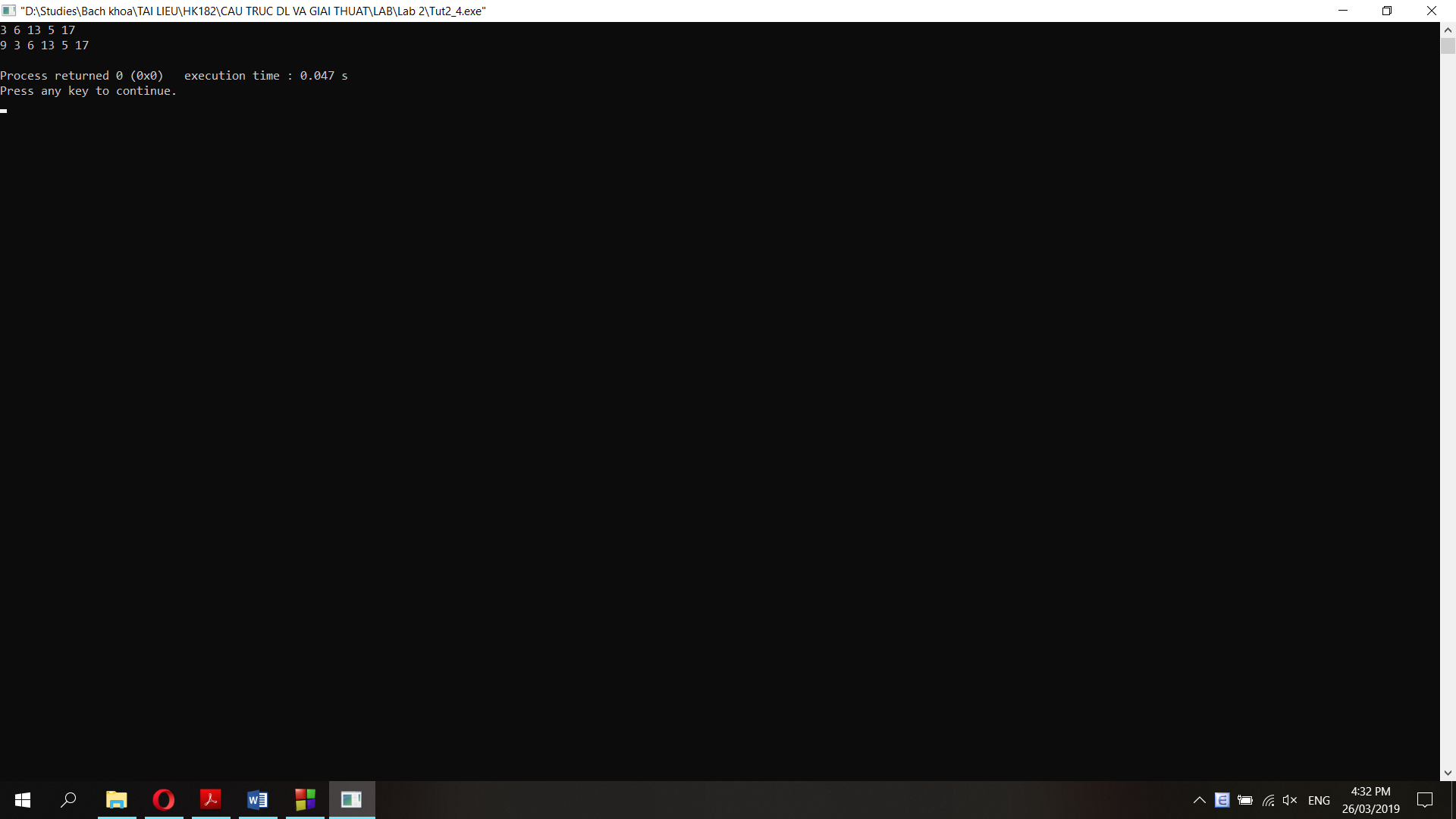
**a. Insert 1 node (data = 9) at the beginning:**

pNew = new Node;

pNew->data = 9;

pNew->next = pHead;

pHead = pNew;



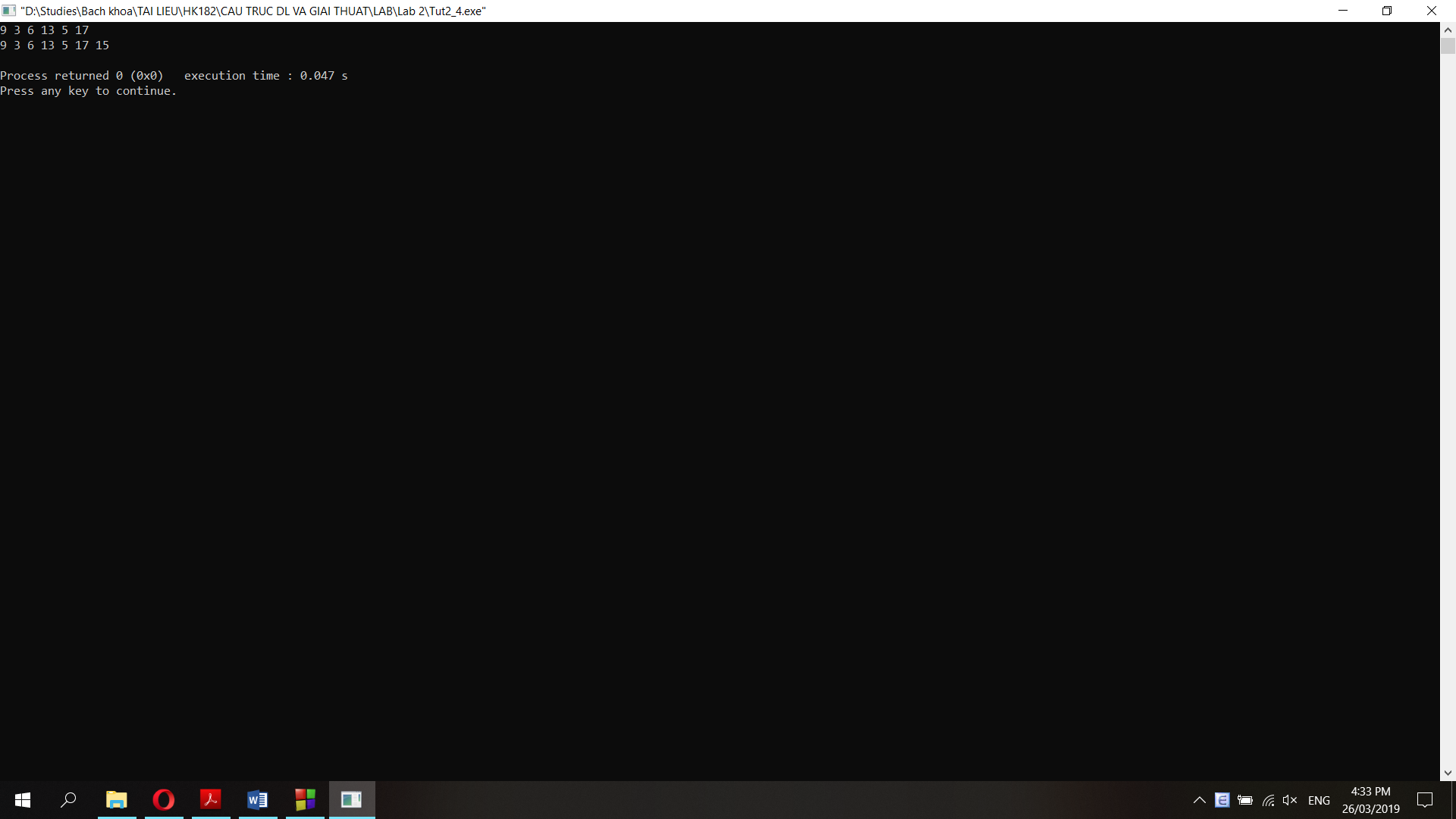
**b. Insert 1 node (data = 15) at the end:**

pNew = new Node;

pNew->data = 15;

pTail->next = pNew;

pTail = pNew;



**c. Delete the node which has the data = 17:**

Node \*pTemp2;

pNew = pHead;

while (pNew->data != 17){

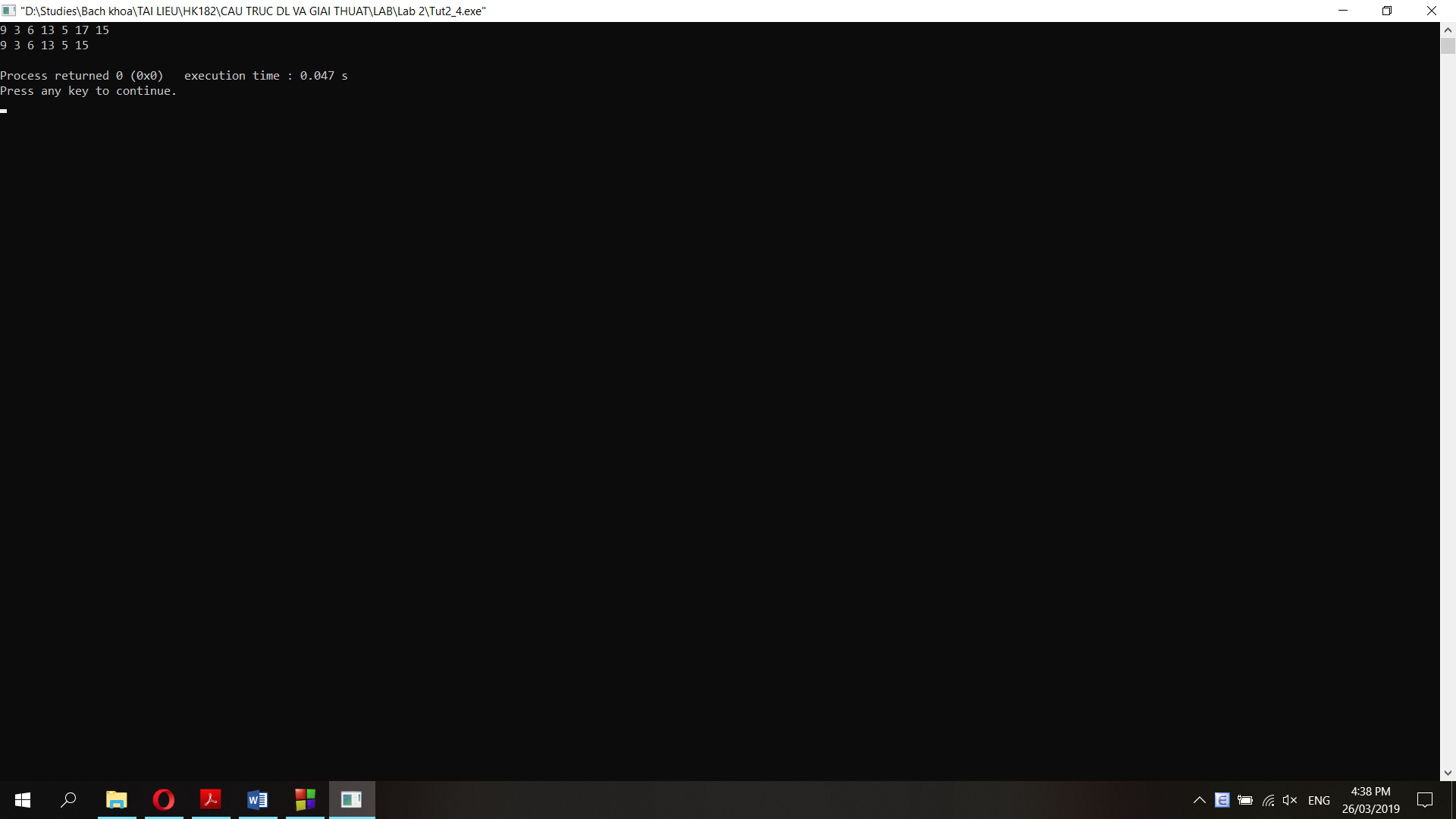
pTemp2 = pNew;

pNew = pNew->next;

}

pTemp2->next = pNew->next;

delete(pNew);



**f. Delete the node which pTemp points to:**

pTemp2 = pHead;

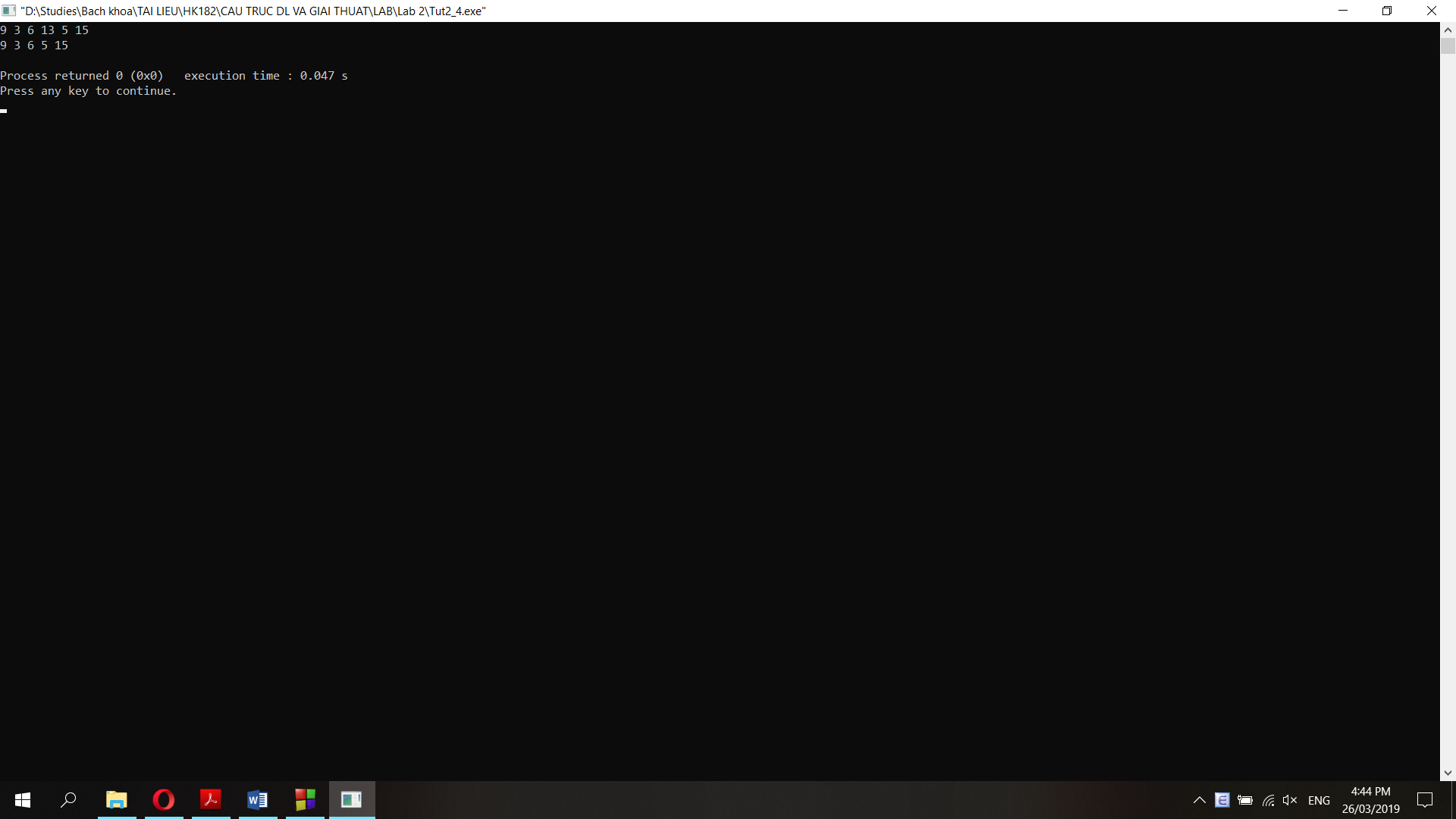
while (pTemp2->next != pTemp){

pTemp2 = pTemp2->next;

}

pTemp2->next = pTemp->next;

delete(pTemp);



**Question 2:**

**a. Function prints a single linked list (pTail->next = NULL):**

void printSingleLinkedList(node\* pHead){

node \*pTemp;

pTemp = pHead;

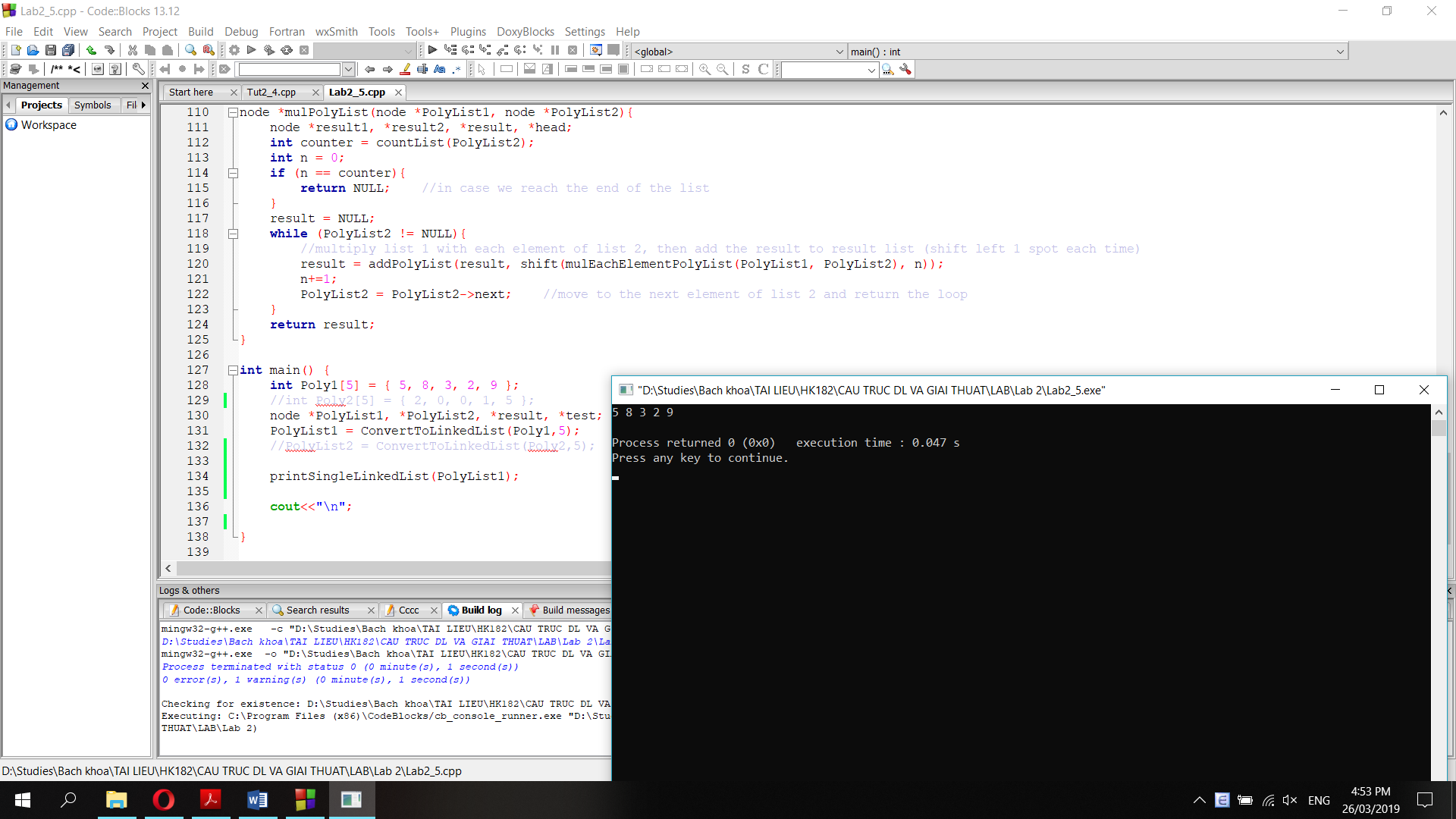
while (pTemp != NULL){

cout<<pTemp->data<<" ";

pTemp = pTemp->next;

}

}



**b. Function prints a circular linked list (pTail->next = pHead->next)**

void printCircularLinkedList(node\* pHead){

node \*pTemp;

pTemp = pHead;

cout<<pTemp->data<<" ";

pTemp = pTemp->next;

cout<<pTemp->data<<" ";

pTemp = pTemp->next;

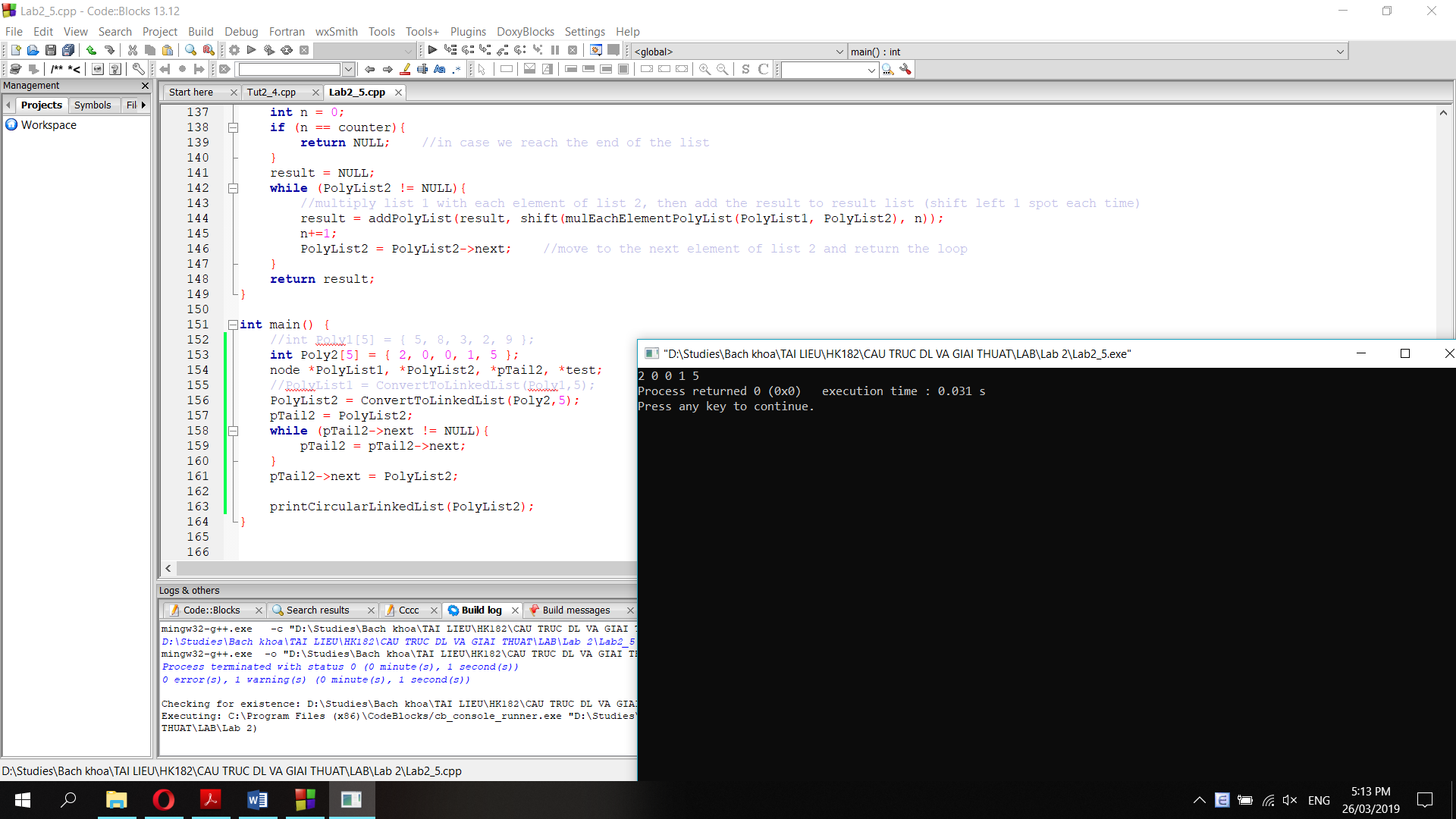
while (pTemp != pHead){

cout<<pTemp->data<<" ";

pTemp = pTemp->next;

}

}



**c. Function prints both single linked list and circular linked list:**

void printList(node\* pHead){

node \*pTemp;

pTemp = pHead;

cout<<pTemp->data<<" ";

pTemp = pTemp->next;

cout<<pTemp->data<<" ";

pTemp = pTemp->next;

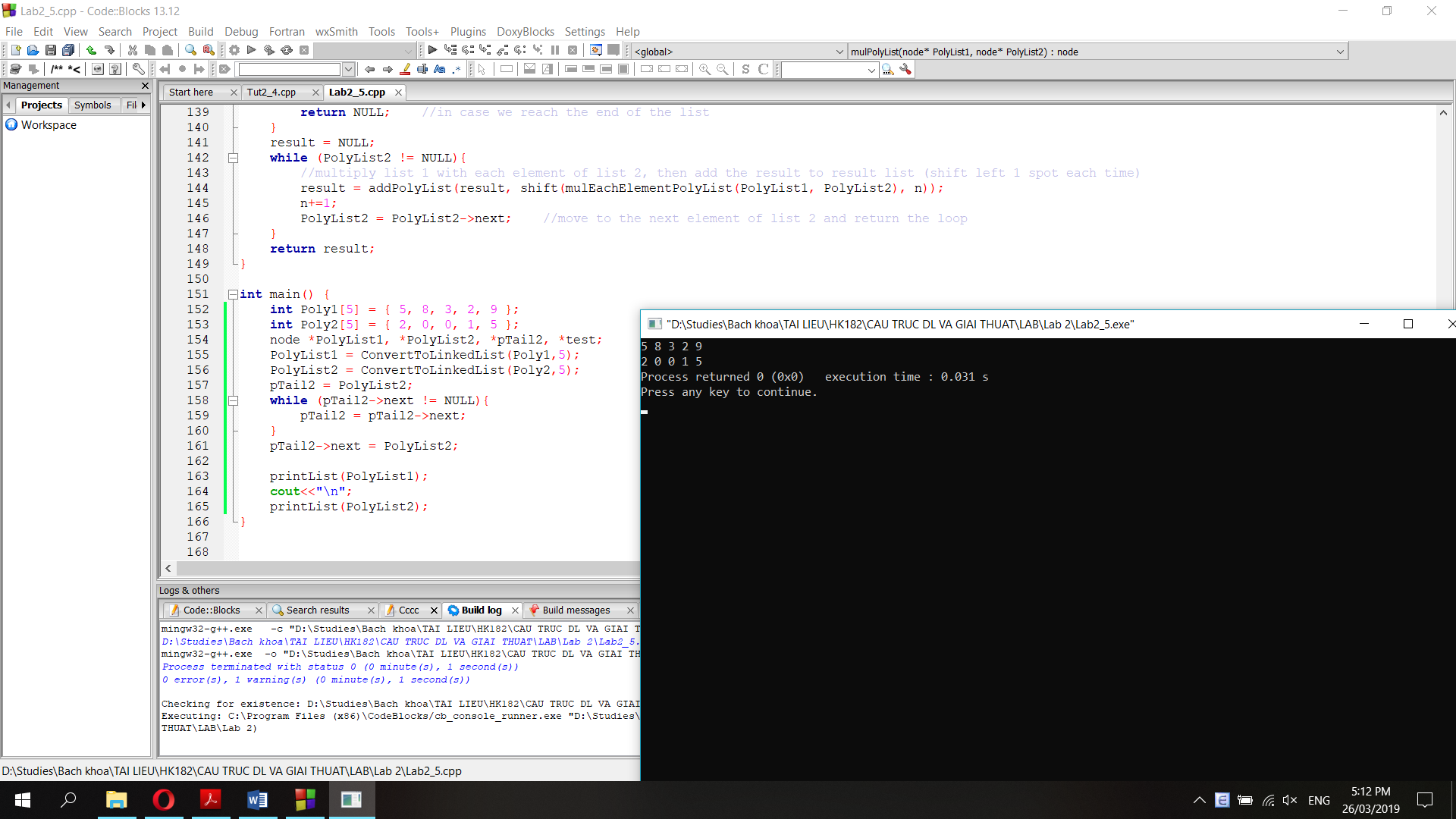
while (pTemp != NULL && pTemp != pHead){

cout<<pTemp->data<<" ";

pTemp = pTemp->next;

}

}



**Question 3: Function searches a node of a single linked list**

node\* searchList(node\* pHead, int data){

node \*result;

result = pHead;

while (result->data != data && result != NULL){

result = result->next;

}

if (result->data = data){

return result;

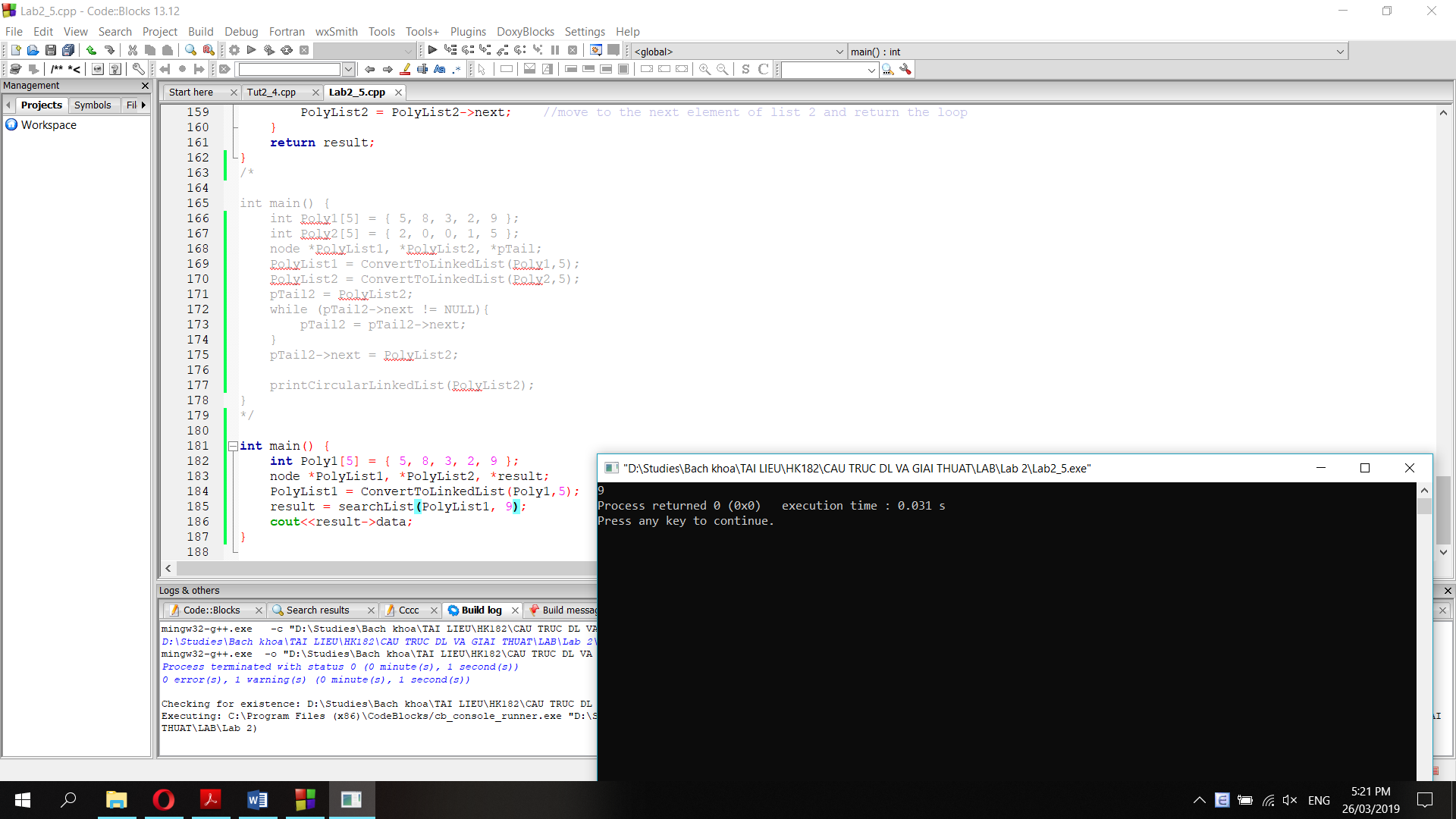
}

else {

return NULL;

}

}



**Question 4: Function delete the Nth node.**

//count the number of nodes in a list

int countList(node \*head){

int counter=1;

while (head->next != NULL){

counter += 1;

head = head->next;

}

return counter;

}

//delete the Nth node

node\* deleteNth(node\* head, int n){

node \*result;

int counter = countList(head);

if (n==0){ //if n=0 delete the head and return a new list

result = head->next;

delete(head);

return result;

}

else if (n<0 || n>counter){ //if n<0 or greater than the length of the list, do nothing

return head;

}

else {

int counter2 = 0;

result = head;

node \*temp;

while (counter2 != n-1){ //point result to the n-1 node

result = result->next;

counter2 += 1;

}

temp = result;

result = result->next;

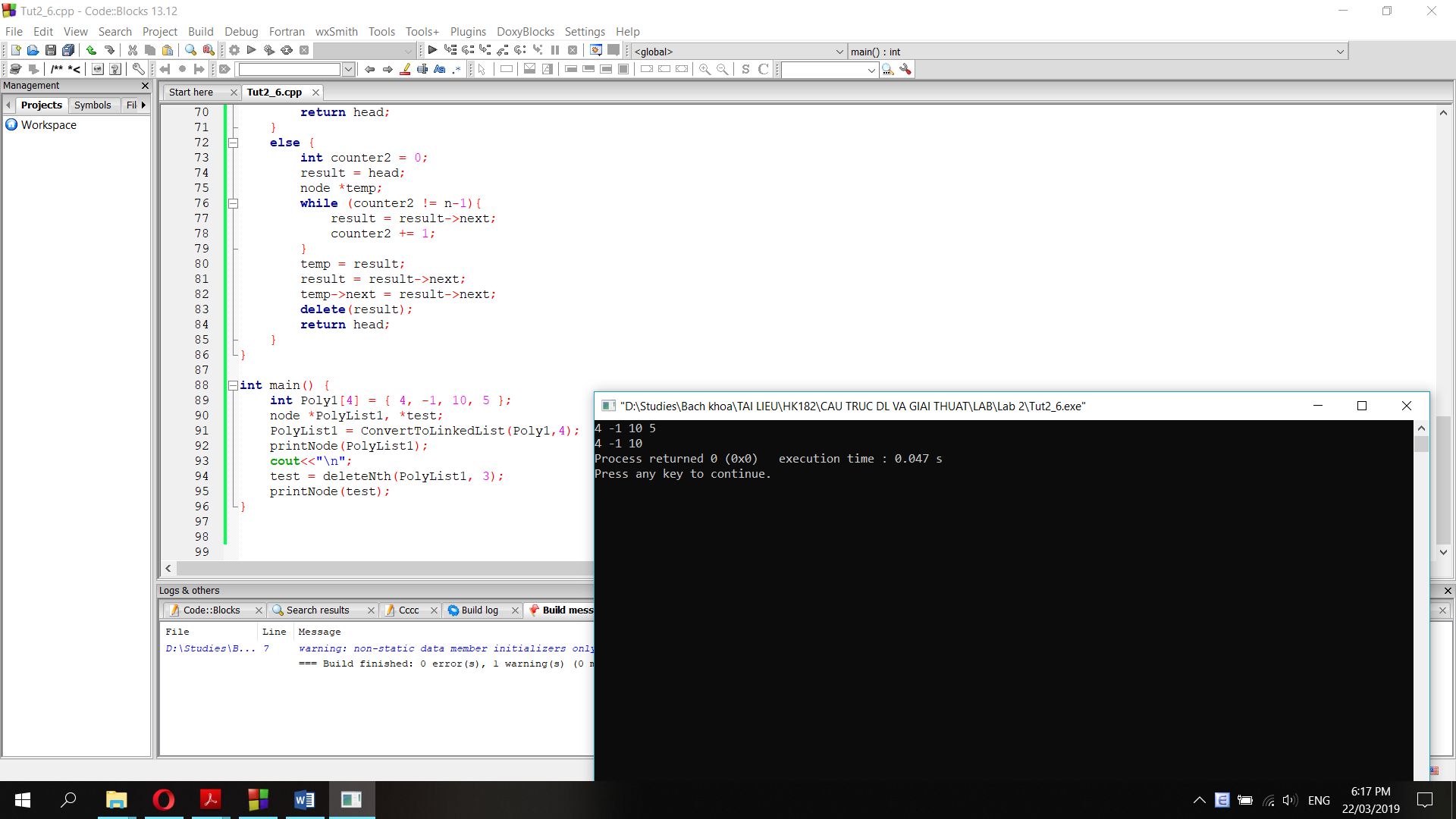
temp->next = result->next;

delete(result);

return head;

}

}



**Question 5:**

void func1(node\* head) {

node\* temp = head;

while (temp != NULL) {

if (temp->next == NULL) {

temp->next = head;

return;

}

temp = temp->next;

}

}

**Nếu truyền con trỏ head của một single linked list vào hàm func1() trên, hàm sẽ chuyển single linked list thành circular linked list.**

**Giải thích: hàm func1() duyệt con trỏ temp từ nút head cho đến nút cuối cùng, sau đấy cho temp->next = head, nghĩa là tail->next = head.**