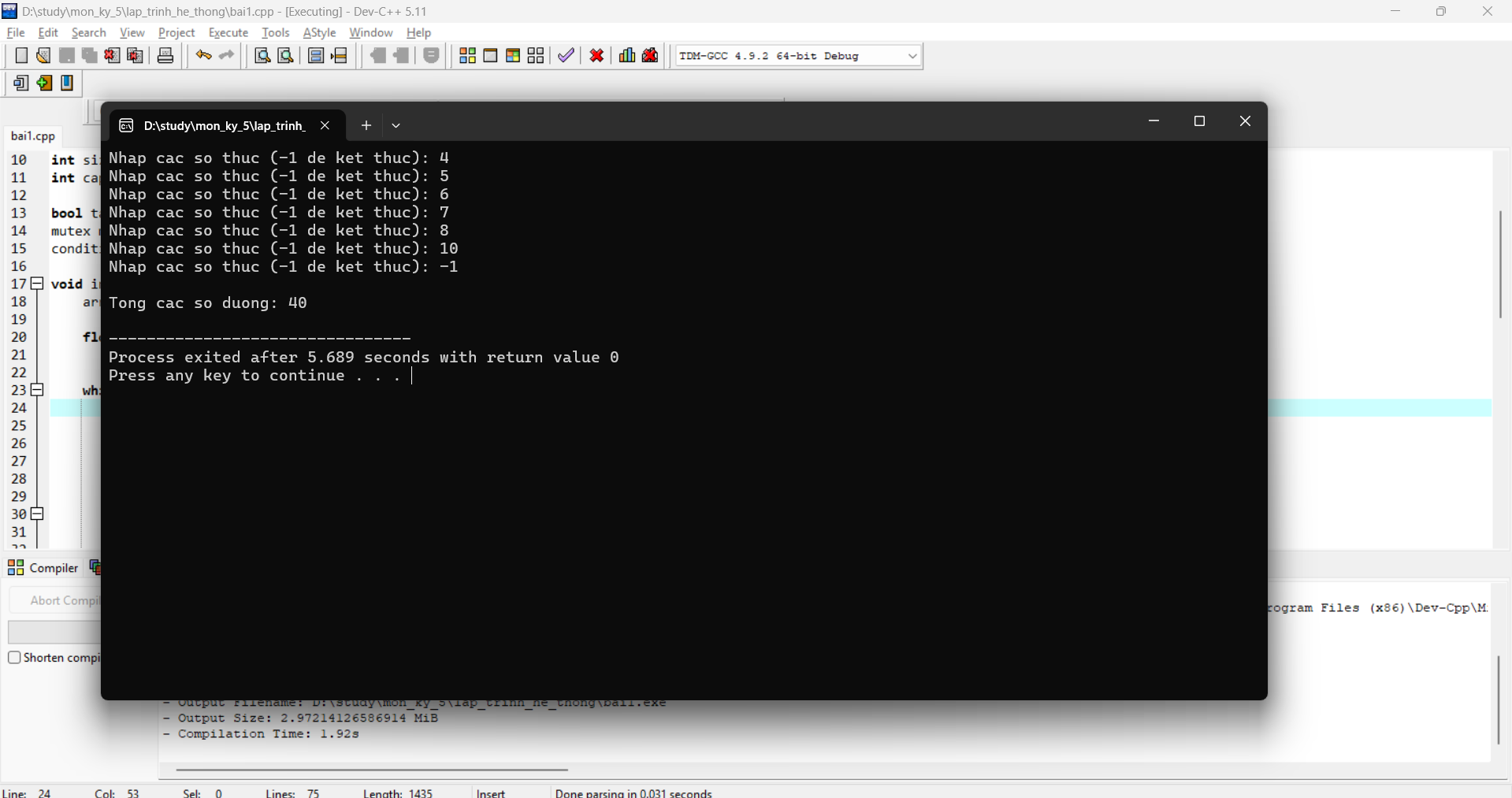
Nguyễn Hải Dương 22810310241

Kiểm tra 08/04/2025

**Bài 01:**  
Lập 1 chương trình trên ngôn ngữ lập trình tùy chọn C++/Java/C# gồm 2 tác vụ (Task1, Task2) hoạt động ở chế độ không đồng bộ như sau:  
  
***Task1****:*  
1. Viết một hàm nhập một mảng số thực từ bàn phím, mảng dừng khi nhập giá trị ‘-1’. Sử dụng bộ nhớ động cho mảng. Viết hàm main()  
gọi hàm, dịch và chạy chương trình.  
2.  Thiết lập Task1 bằng cách đặt hàm trên vào 1 luồng, kích hoạt luồng trong hàm main(). Dịch và chạy lại chương trình.  
  
***Task2****:*  
3.Viết một hàm tính tổng các giá trị dương trong mảng, vòng lặp dừng khi Task1 hoàn tất. Bổ sung lệnh gọi hàm trong hàm main(),  
dịch và chạy lại chương trình.  
4. Thiết lập Task2 bằng cách đặt hàm trên vào 1 luồng, kích hoạt luồng trong hàm main(). Thêm thông báo bằng âm thanh khi tính tổng xong. Dịch và chạy lại chương trình.

Bài làm :



**Code Bai1:**

#include <iostream>

#include <thread>

#include <mutex>

#include <condition\_variable>

#include <windows.h>

using namespace std;

float\* arr = nullptr;

int size = 0;

int capacity = 10;

bool task1\_done = false;

mutex mtx;

condition\_variable cv;

void inputArray() {

arr = new float[capacity];

float num;

while (true) {

cout << "Nhap cac so thuc (-1 de ket thuc): ";

cin >> num;

if (num == -1) break;

unique\_lock<mutex> lock(mtx);

if (size >= capacity) {

capacity \*= 2;

float\* newArr = new float[capacity];

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

newArr[i] = arr[i];

}

delete[] arr;

arr = newArr;

}

arr[size++] = num;

}

{

unique\_lock<mutex> lock(mtx);

task1\_done = true;

}

cv.notify\_one();

}

void sumPositiveNumbers() {

unique\_lock<mutex> lock(mtx);

cv.wait(lock, [] { return task1\_done; });

float sum = 0;

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

if (arr[i] > 0) {

sum += arr[i];

}

}

cout << "\nTong cac so duong: " << sum << endl;

Beep(750, 300); // Âm thanh thông báo

}

int main() {

thread t1(inputArray);

thread t2(sumPositiveNumbers);

t1.join();

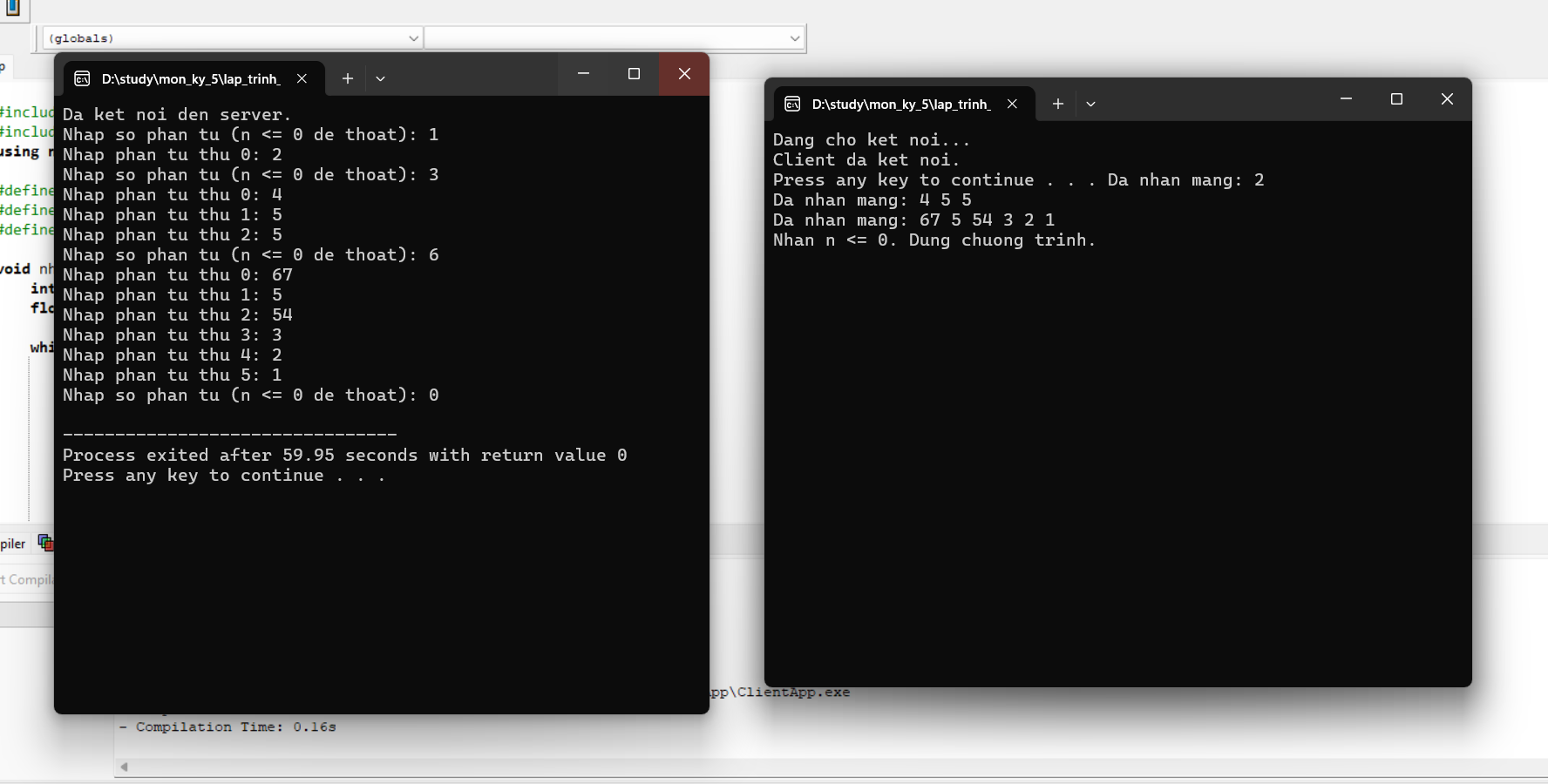
t2.join();

delete[] arr;

return 0;

}

**Bài 02:**  
Lập 2 chương trình trên ngôn ngữ lập trình tùy chọn ***C++/Java/C#*** gồm CT1(client), CT2 (server) sử dụng Socket API, giao thức TCPIP truyền nhận dữ liệu tại IP="127.0.0.1" (localhost), port=100 và hoạt động như sau:  
  
***CT1****:*  
1. Viết 1 hàm gồm 1 vòng lặp nhập vào từ bàn phím mảng số thực (nhập số phần tử **n**, sau đó nhập từng phần tử mảng), nếu nhập n thỏa **n** <=0  thì thoát khỏi vòng lặp. Viết hàm main() gọi hàm, dịch và chạy chương trình.  
2. Bổ sung lệnh truyền giá trị **n**và truyền biến mảng đến cho **CT2** trong vòng lặp. Dịch và chạy lại chương trình **CT1**.  
  
***CT2****:*  
3. Viết 1 hàm có 1 vòng lặp vô hạn nhận giá trị biến nguyên **n**được gửi đến từ **CT1,**Nếu nhận được **n**thì:  
Trường hợp 1: Nếu **n** **< 0**  thì thoát khỏi vòng lặp.  
Trường hợp 2: Trái lại, nhận tiếp mảng **n**số thực được gửi đến từ **CT1** và hiển thị **n** số nhận được. Viết hàm main(), dịch chương trình **CT2**.  
Rồi đóng **CT1** đang chạy, sau đó lần lượt chạy **CT2** và chạy **CT1** để xem hiệu ứng nhập mảng ở **CT1**, hiển thị mảng ở **CT2**.  
4. Đặt hàm ở câu 3 vào 1 timer, bổ sung lệnh kích hoạt timer trong hàm main(). Dịch chương trình.  
  
***Biên dịch và chạy lần sau cùng:***  
+ Đóng tất cả **CT1**, **CT2**đang chạy.  
+ Chạy chương trình **CT2**sau đó chạy **CT1** để xem hiệu ứng kết nối, truyền và nhận dữ liệu.  
+ Chụp ảnh màn hình kết quả chạy



**Code của ClientApp:**

#include <iostream>

#include <winsock2.h>

using namespace std;

#define SERVER\_IP "127.0.0.1"

#define PORT 100

#define MAX\_SIZE 100

void nhapVaGuiMang(SOCKET sock) {

int n;

float arr[MAX\_SIZE];

while (true) {

cout << "Nhap so phan tu (n <= 0 de thoat): ";

cin >> n;

send(sock, (char\*)&n, sizeof(int), 0);

if (n <= 0) break;

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

cout << "Nhap phan tu thu " << i << ": ";

cin >> arr[i];

}

send(sock, (char\*)arr, sizeof(float) \* n, 0);

}

}

int main() {

WSADATA wsa;

WSAStartup(MAKEWORD(2, 2), &wsa);

SOCKET sock = socket(AF\_INET, SOCK\_STREAM, 0);

sockaddr\_in server;

server.sin\_family = AF\_INET;

server.sin\_port = htons(PORT);

server.sin\_addr.s\_addr = inet\_addr(SERVER\_IP);

connect(sock, (sockaddr\*)&server, sizeof(server));

cout << "Da ket noi den server.\n";

nhapVaGuiMang(sock);

closesocket(sock);

WSACleanup();

return 0;

}

**Code của ServerApp:**

#include <iostream>

#include <winsock2.h>

#include <windows.h>

using namespace std;

#define PORT 100

#define MAX\_SIZE 100

SOCKET clientSock;

void nhanMang() {

int n;

float arr[MAX\_SIZE];

while (true) {

int recvBytes = recv(clientSock, (char\*)&n, sizeof(int), 0);

if (recvBytes <= 0) break;

if (n <= 0) {

cout << "Nhan n <= 0. Dung chuong trinh.\n";

break;

}

recv(clientSock, (char\*)arr, sizeof(float) \* n, 0);

cout << "Da nhan mang: ";

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

cout << arr[i] << " ";

}

cout << "\n";

}

}

VOID CALLBACK TimerCallback(PVOID lpParam, BOOLEAN TimerOrWaitFired) {

nhanMang();

}

int main() {

WSADATA wsa;

WSAStartup(MAKEWORD(2, 2), &wsa);

SOCKET serverSock = socket(AF\_INET, SOCK\_STREAM, 0);

sockaddr\_in server;

server.sin\_family = AF\_INET;

server.sin\_port = htons(PORT);

server.sin\_addr.s\_addr = INADDR\_ANY;

bind(serverSock, (sockaddr\*)&server, sizeof(server));

listen(serverSock, 1);

cout << "Dang cho ket noi...\n";

sockaddr\_in client;

int clientLen = sizeof(client);

clientSock = accept(serverSock, (sockaddr\*)&client, &clientLen);

cout << "Client da ket noi.\n";

HANDLE hTimer = NULL;

HANDLE hTimerQueue = CreateTimerQueue();

CreateTimerQueueTimer(&hTimer, hTimerQueue, TimerCallback, NULL, 0, 0, WT\_EXECUTEDEFAULT);

system("pause");

CloseHandle(hTimer);

CloseHandle(hTimerQueue);

closesocket(clientSock);

closesocket(serverSock);

WSACleanup();

return 0;

}