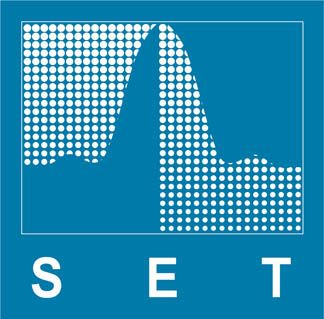
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN ĐIỆN TỬ - VIỄN THÔNG**

\*\*\*



**BÁO CÁO THÍ NGHIỆM**

**MÔN: NGÔN NGỮ LẬP TRÌNH**

**Phòng thí nghiệm: 306A Nhà C9**

Họ tên sinh viên: Lương Quỳnh Thư

Mã lớp TN: 683609

Mã số SV: 20172841

Lớp: ĐTVT-01

Email: quynhthuluong@gmail.com

*Hà Nội, 2019*

# **MODULE 1: CÁC THAO TÁC TRÊN DEV-C++**

## **BÀI 1: XUẤT, NHẬP VÀ GÁN DỮ LIỆU**

### Program 1.1:

Viết một chương trình thực hiện các công việc sau:

1. Nhập 1 kí tự từ bàn phím

2. In kí tự đó và mã ASCII, mã cơ số 8 và mã cơ số 16 của nó ra màn hình

Code:

#include<iostream>

using namespace std;

int main(){

char c;

cout<< "Nhap 1 ki tu: " ;

c=cin.get();

cout<<endl;

cout<<"Ki tu: "<<char(c)<<endl;

cout.setf(ios::showbase);

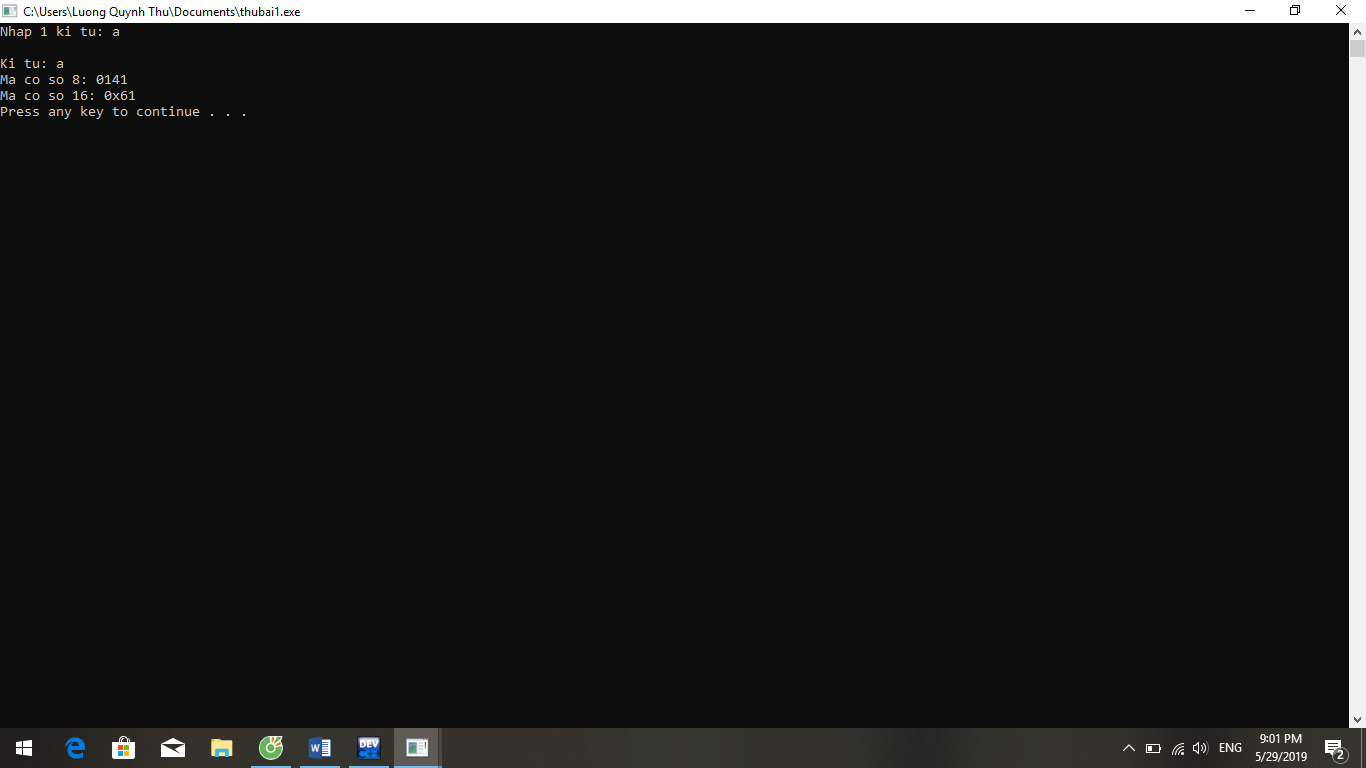
cout<<"Ma co so 8: "<<oct<<int (c)<<endl;

cout<<"Ma co so 16: "<<hex<<int (c)<<endl;

system("PAUSE");

}

Kết quả chương trình:



### Program 1.2:

Viết chương trình tính điện kháng của một tụ điện theo công thức:  với f và C tương ứng là tần số và điện dung của tụ. Chương trình sẽ thực hiện các công việc sau:

1. Tự khai báo hằng số 

2. Nhập các giá trị f, C từ bàn phím

3. Tính điện kháng rồi in ra màn hình

Code:

#include<iostream>

using namespace std;

int main(){

const double PI = 3.1415926;

double f,C;

cout<<"Nhap tan so f = ";

cin>>f;

cout<<"Nhap dien dung C = ";

cin>>C;

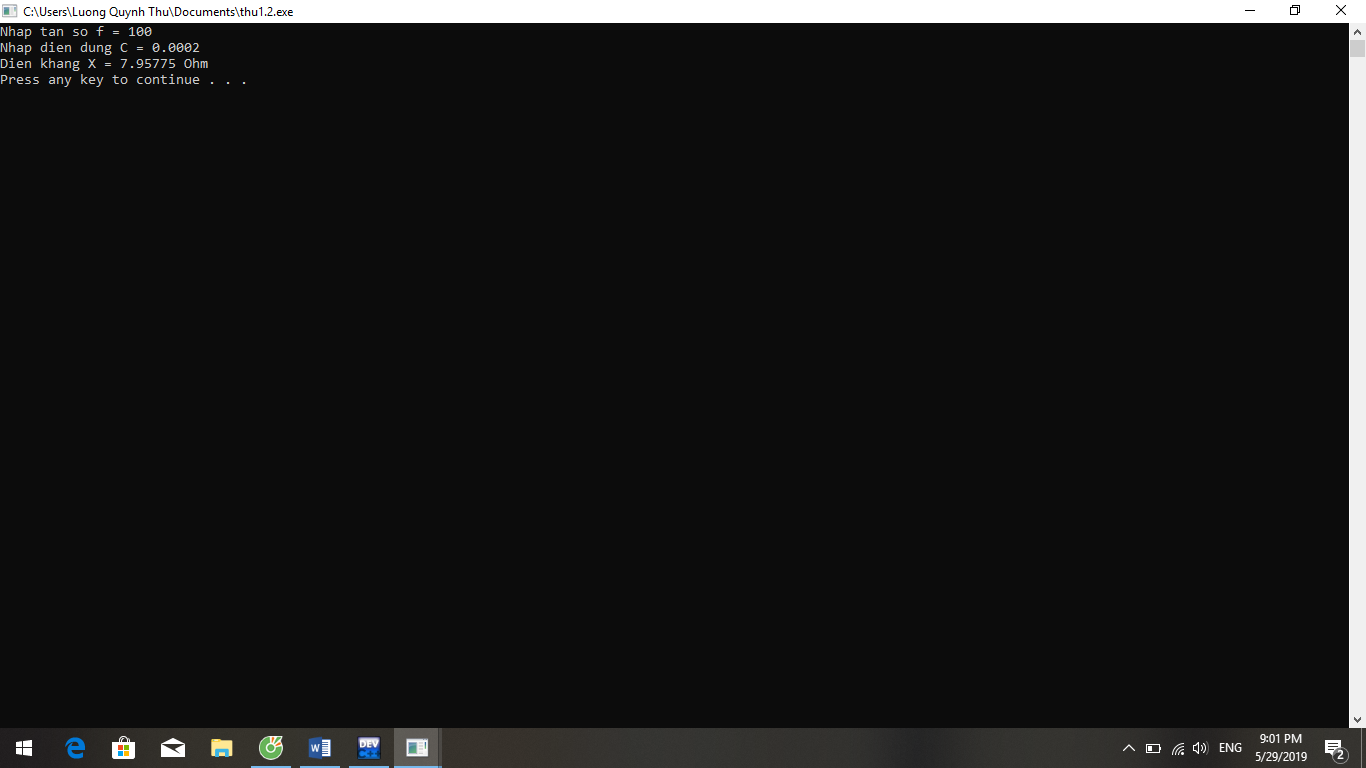
double X=1/(2\*PI\*f\*C);

cout<<"Dien khang X = "<<X<<" Ohm"<<endl;

system("pause");

}

Kết quả chương trình:



### Program 1.3:

Viết chương trình tính điện trở tương đương của 3 điện trở mắc nối tiếp hoặc song song. Chương trình sẽ thực hiện các công việc sau:

1. Nhập giá trị từng điện trở từ bàn phím
2. Nhập cách mắc điện trở (nối tiếp hoặc song song)
3. Tính và in kết quả ra màn hình

Code:

#include<iostream>

using namespace std;

int main(){

double R1,R2,R3,R;

cout<<"Nhap dien tro R1, R2, R3: ";

cin>>R1>>R2>>R3;

int choose;

cout<<"Moi ban chon cach mac dien tro: "<<endl;

cout<<" 1. Noi tiep"<<endl;

cout<<" 2. Song song"<<endl;

do{

cout<<" Lua chon: ";

cin>>choose;

if(choose-1 && choose-2)

cout<<" Lua chon khong hop le! Moi ban chon lai!"<<endl;

} while(choose-1 && choose-2);

switch(choose){

case 1:

R=R1+R2+R3;

break;

case 2:

R=1/R1+1/R2+1/R3;

R=1/R;

break;

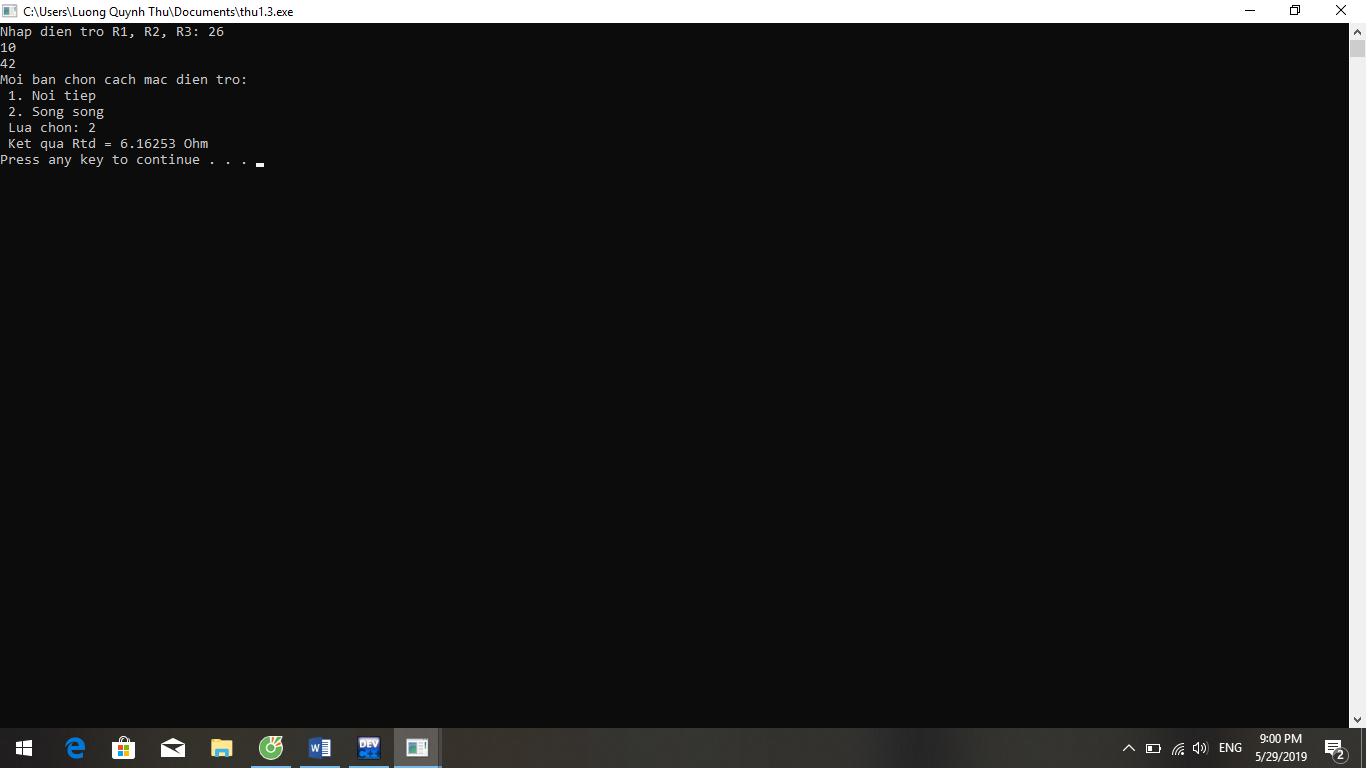
}

cout<<" Ket qua Rtd = "<<R<<" Ohm"<<endl;

system("pause");

}

Kết quả chương trình:



## **BÀI 2: THAO TÁC VỚI MẢNG NHIỀU CHIỀU**

### Program 2.1:

Mở rộng Program 1.3, viết chương trình tính điện trở tương đương của N điện trở mắc nối tiếp hoặc song song. Chương trình sẽ thực hiện các công việc sau:

1. Nhập số điện trở N và giá trị từng điện trở từ bàn phím. Có kiểm tra tính hợp lệ của dữ liệu gán vào (giá trị N>0 và N<20; các điện trở phải có giá trị >0)

2. Nhập cách mắc điện trở (nối tiếp hoặc song song)

3. Tính và in kết quả ra màn hình

Code:

#include<iostream>

using namespace std;

int main(){

int N;

double \*R,res=0;

cout<<"Nhap so dien tro: ";

do{

cin>>N;

if(N<=0 || N>=20)

cout<<"Gia tri nhap vao khong hop le. Moi nhap lai: ";

} while(N<=0 || N>=20);

R = new double[N];

cout<<"Nhap gia tri cho "<<N<<" dien tro:"<<endl;

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

cout<<"Nhap gia tri dien tro R"<<i+1<<": ";

do{

cin>>R[i];

if(R[i]<=0) cout<<"Gia tri nhap vao khong hop le. Moi nhap lai: ";

} while(R[i]<=0);

}

int choose;

cout<<"Moi ban chon cach mac dien tro: "<<endl;

cout<<" 1. Noi tiep"<<endl;

cout<<" 2. Song song"<<endl;

do{

cout<<" Lua chon: ";

cin>>choose;

if(choose-1 && choose-2)

cout<<" Lua chon khong hop le! Moi ban chon lai!"<<endl;

} while(choose-1 && choose-2);

switch(choose){

case 1:

for(int i=0;i<N;i++) res+=R[i];

break;

case 2:

for(int i=0;i<N;i++) res+=1/R[i];

res=1/res;

break;

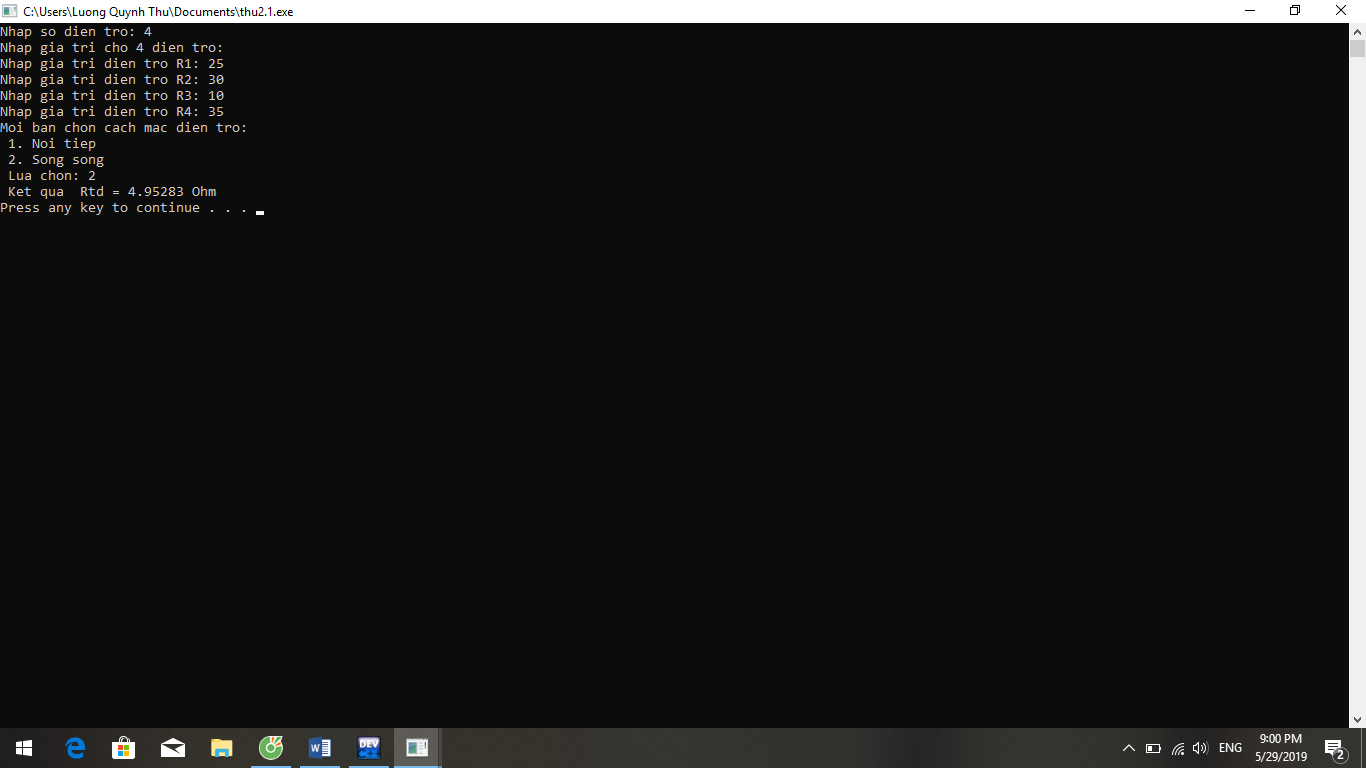
}

cout<<" Ket qua Rtd = "<<res<<" Ohm"<<endl;

system("PAUSE");

}

Kết quả chương trình:



### Program 2.2:

Viết chương trình cho phép đếm số từ có trong một chuỗi. Yêu cầu:

1. Nhập vào một chuỗi có độ dài không lớn quá 200 kí tự. Có kiểm tra tính hợp lệ của dữ liệu nhập vào.
2. Đếm số từ có trong chuỗi và in kết quả ra màn hình

Code:

#include<iostream>

#include<string>

using namespace std;

void Input(string& s){

do{

cout<<"Nhap vao 1 chuoi (khong qua 200 ki tu) : ";

getline(cin,s);

if(s.size()==0 || s.size()>200)

cout<<"Chuoi nhap vao khong hop le. Moi nhap lai!"<<endl;

} while(s.size()==0 || s.size()>200);

}

int numOfWords(string s){

int num=0, id=0;

while(id<s.size()){

if(s[id]!=' '){

num++;

while(s[id]!=' ' && id<s.size())

id++;

}

id++;

}

return num;

}

int main(){

string s;

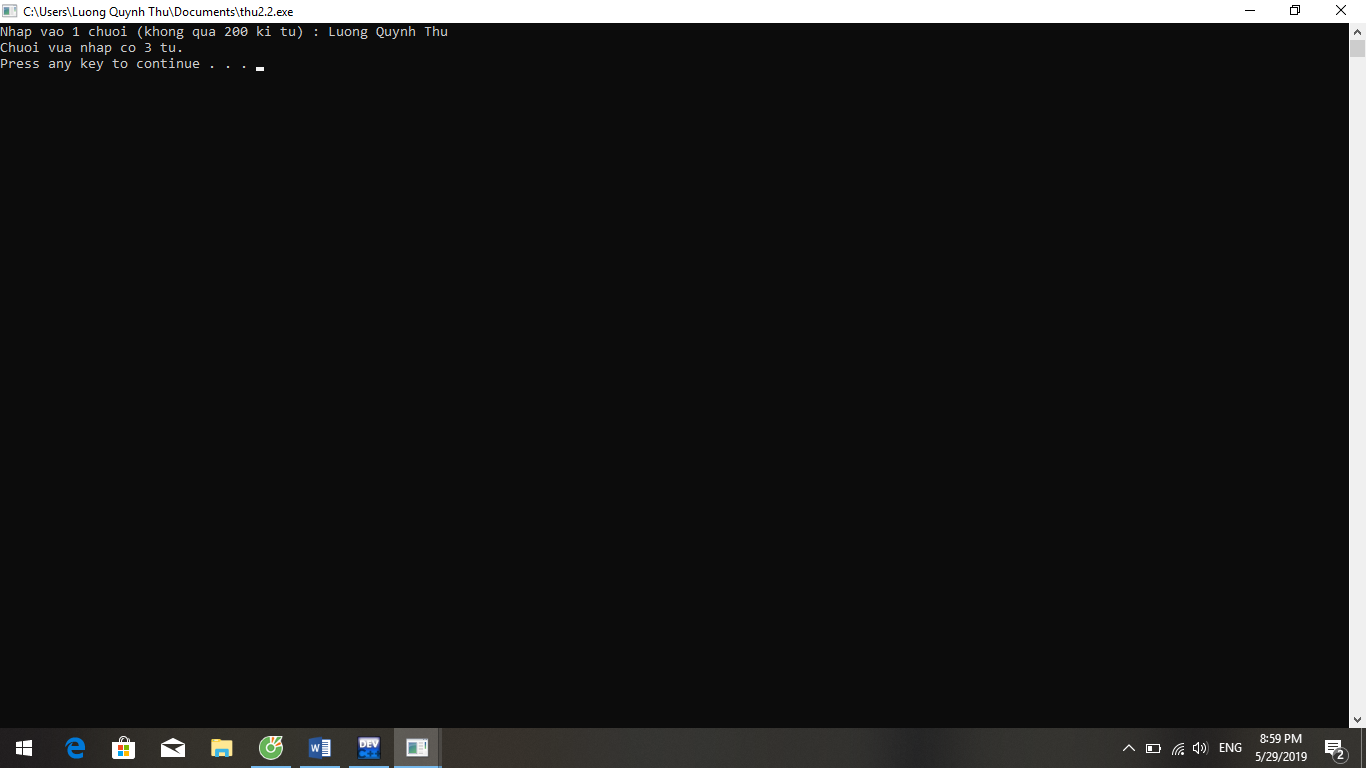
Input(s);

cout<<"Chuoi vua nhap co "<<numOfWords(s)<<" tu."<<endl;

system("pause");

}

Kết quả chương trình:



### Program 2.3:

Viết chương trình thực hiện các yêu cầu sau:

1. Nhập một ma trận kích thước MxN kiểu số thực (0 < M, N < 10) và các giá trị của các phần tử trong đó từ bàn phím
2. Tính tổng các phần tử theo từng hàng, theo từng cột rồi in ra. In tổng của toàn bộ các phần tử trong ma trận

Code:

#include<iostream>

using namespace std;

int M,N;

double\*\* matrix;

void Nhap(){

do{

cout<<"Nhap so hang cua ma tran M = ";

cin>>M;

cout<<"Nhap so cot cua ma tran N = ";

cin>>N;

if(!(M>0 && M<10 && N>0 && N<10)){

cout<<"Gia tri ban vua nhap khong hop le: 0 < M,N < 10"<<endl;

cout<<"Moi ban nhap lai!"<<endl;

}

}

while(!(M>0 && M<10 && N>0 && N<10));

matrix = new double\*[M];

for(int i=0;i<M;i++)

matrix[i] = new double[N];

cout<<"Nhap ma tran:"<<endl;

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

cout<<"\tNhap hang "<<i+1<<": ";

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

cin>>matrix[i][j];

}

}

double Tong\_Hang(int row){

double s=0;

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

s+=matrix[row-1][i];

return s;

}

double Tong\_Cot(int col){

double s=0;

for(int i=0;i<M;i++)

s+=matrix[i][col-1];

return s;

}

int main(){

double s=0;

Nhap();

cout<<endl;

cout<<"Tong cac phan tu theo tung hang:"<<endl;

for(int i=0;i<M;i++)

cout<<"\tTong theo hang "<<i+1<<": "<<Tong\_Hang(i+1)<<endl;

cout<<"Tong cac phan tu theo tung cot: "<<endl;

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

cout<<"\tTong theo cot "<<i+1<<": "<<Tong\_Cot(i+1)<<endl;

s+=Tong\_Cot(i+1);

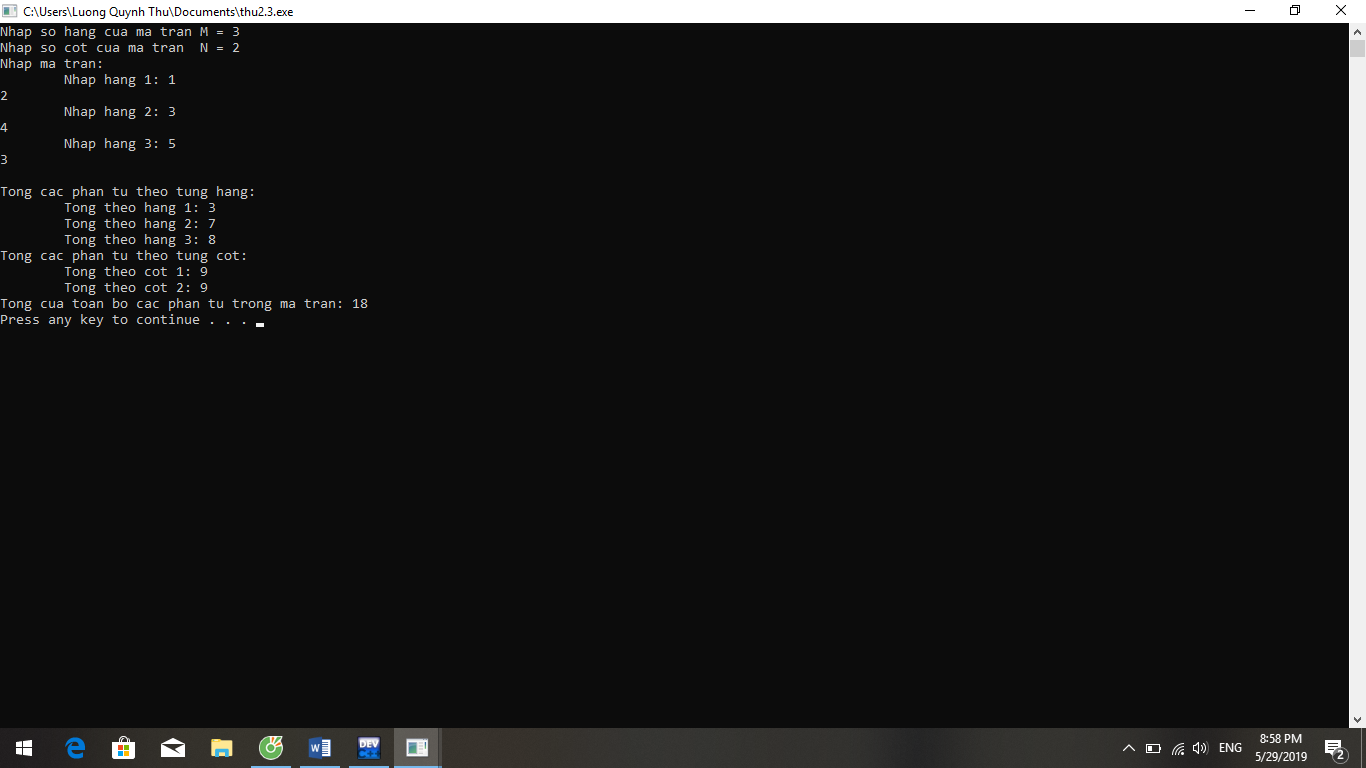
}

cout<<"Tong cua toan bo cac phan tu trong ma tran: "<<s<<endl;

system("pause");

}

Kết quả chương trình:



### Yêu cầu về nhà:

1. Hàm đệ quy int Factorial(int N) tính N!.

Code:

int Factorial(int N) {

if(!N) return 1;

return N\*Factorial(N-1);

}

1. Hàm đệ quy double Pow(double x, int n) tính xn.

Code:

double Pow(double x, int n){

if(!n) return 1;

if(n>0) return x\*Pow(x,n-1);

if(n<0) return 1.0/Pow(x,-n);

}

1. Hàm đệ quy USCLN(int a, int b) tìm ước số chung lớn nhất của a và b.

Code:

int USCLN(int a, int b){

if(!a) return b;

return USCLN(b%a,a);

}

1. Viết hàm double Value(double a[], int n, double x0) để tính giá trị của hàm



Code:

double Value(double a[], int n, double x0){

double s=0;

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

s+=a[i]\*Pow(x0,i);

}

return s;

}

Hàm *Pow* đã được định nghĩa ở mục 2.

# **MODULE 2: CÁC BÀI TẬP NÂNG CAO**

Program 1.1: Số phức

Code:

#include <iostream>

using namespace std;

class Complex{

private:

double re,im;

public:

// Ham tao thiet lap tu 2 tham so

Complex(double r = 0, double i = 0): re(r), im(i) {}

// Ham tao copy tu 1 so phuc

Complex(const Complex& C): re(C.re), im(C.im) {}

public:

Complex operator + (Complex C); // phep cong

Complex operator - (Complex C); // phep tru

Complex operator \* (Complex C); // phep nhan

Complex operator / (Complex C); // phep chia

public:

friend ostream& operator<<(ostream& out,const Complex& C)

{

return out << '(' << C.re << ", " << C.im << "i) ";

}

};

Complex Complex::operator + (Complex C){

return Complex(this->re + C.re, this->im + C.im);

}

Complex Complex::operator - (Complex C){

return Complex(this->re - C.re, this->im - C.im);

}

Complex Complex::operator \* (Complex C){

return Complex(this->re\*C.re-this->im\*C.im,this->re\* C.im+this->im\*C.re);

}

Complex Complex::operator / (Complex C){

double m=C.re\*C.re+C.im\*C.im;

return Complex((this->re\*C.re+this->im\*C.im)/m,(-this->re\*C.im+this->im\*C.re)/m);

}

int main(){

Complex y(3.0,1.5);

Complex z(1.7,-3.0);

double a=2.5;

cout<<y<<" + "<<z<<" + "<<a<<" = "<<y+z+a<<endl;

cout<<y<<" - "<<z<<" = "<<y-z<<endl;

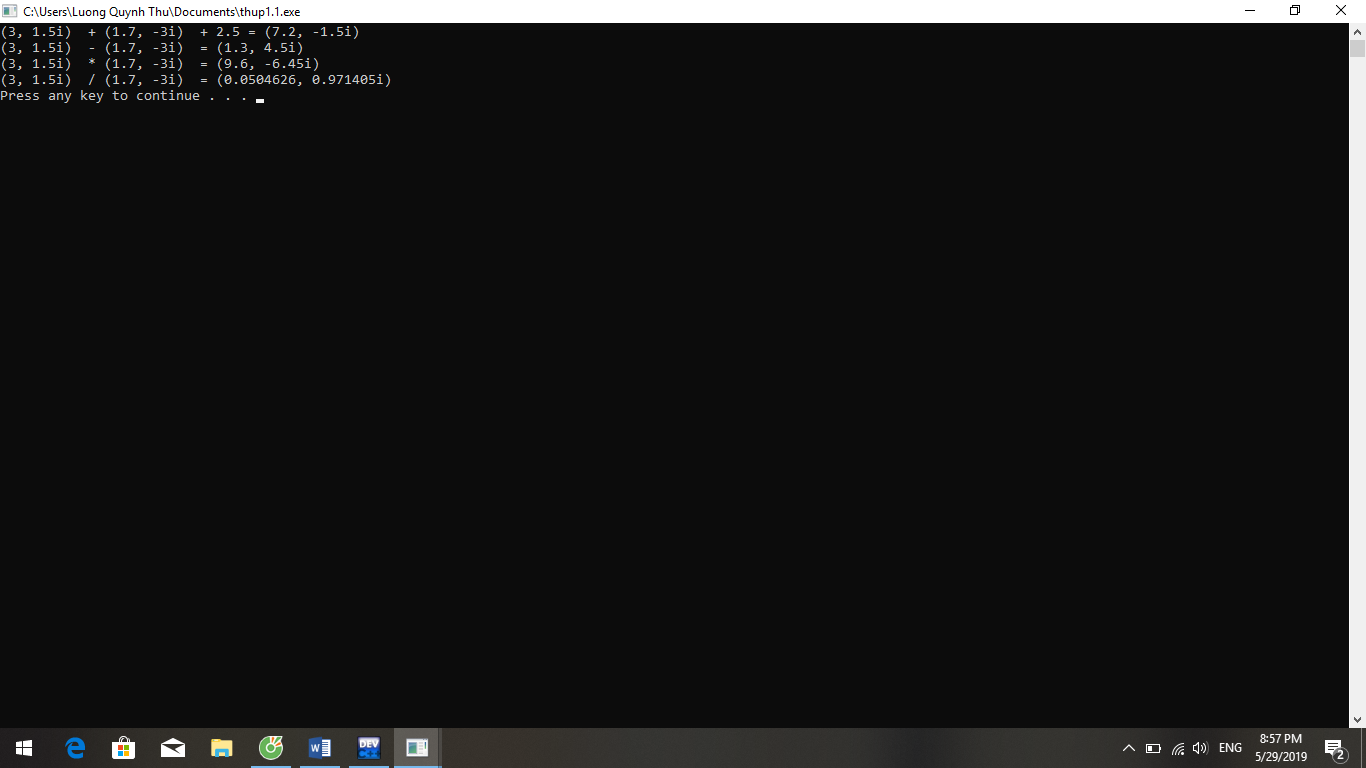
cout<<y<<" \* "<<z<<" = "<<y\*z<<endl;

cout<<y<<" / "<<z<<" = "<<y/z<<endl;

system("PAUSE");

}

Kết quả chương trình:



Program 1.2: Xâu kí tự

Code:

#include<iostream>

#include<cstring>

using namespace std;

class String{

private:

int length;

char \*data;

private:

// Ham tao String tu mang ki tu voi do dai cho truoc

String(int length,char \*data):length(length), data(data) {}

public:

// ham tao mac dinh xau rong

String(): length(0), data(new char[1]) {data[0]='\0';}

// ham tao String tu 1 hang xau ki tu

String(const char\* s){

length=strlen(s);

data = new char[length+1];

strcpy(data,s);

}

// ham tao String copy tu 1 String da co

String (const String& s);

// ham huy

~String() {delete[] data;}

public:

// ham so sanh 2 String

int Compare(String s);

public:

// toan tu truy nhap ki tu tai vi tri index

char & operator[] (int index) {return data[index];}

//toan tu gan s vao String

String& operator = (const String &s);

public:

// toan tu cong thuc hien them ki tu c vao cuoi String

String operator + (char c);

// toan tu cong noi s vao cuoi String

String operator + (String s);

public:

// toan tu so sanh bang

bool operator == (String s);

// toan tu so sanh khac

bool operator != (String s);

// toan tu in String ra man hinh

friend ostream& operator <<(ostream& out,const String &s)

{

return out << s.data;

}

};

// toan tu gan 1 hang xau vao String s

String& String:: operator = (const String &s){

delete[] this->data;

length = s.length;

this->data = new char[length+1];

strcpy(this->data,s.data);

return (\*this);

}

// toan tu cong them 1 hang ki tu vao String s

String String:: operator + (char c){

int length=this->length + 1;

char \*data = new char[length+1];

strcpy(data, this->data);

data[this->length] = c;

data[length]='\0';

return String(length,data);

}

// toan tu cong them 1 hang xau vao String s

String String:: operator + (String s){

int length=this->length+s.length;

char \*data = new char[length+1];

strcpy(data,this->data);

for(int i=0;i<s.length;i++){

data[this->length+i]=s[i];

}

data[length]='\0';

return String(length,data);

}

// ham tao String copy tu 1 String da co

String::String(const String& s){

this->length=s.length;

this->data= new char[this->length+1];

strcpy(this->data,s.data);

}

// ham so sanh 2 String

int String::Compare(String s){

return strcmp(this->data,s.data);

}

// toan tu so sanh bang

bool String::operator == (String s){

return !strcmp(this->data,s.data);

}

// toan tu so sanh khac

bool String::operator != (String s){

return strcmp(this->data,s.data);

}

int main(){

String s="Thu ";

cout<<s+'S'<<endl;

cout<<s+"Thu"<<endl;

system("PAUSE");

}

Kết quả chương trình:

