Mục lục nội dung

[Bài thực hành số 02 – Tuần 35 3](#_Toc161238431)

[BÀI TẬP TRÊN LAP 3](#_Toc161238432)

[**Bài tập 2.1.**Viết hàm tính độ dài cạnh huyền của tam giác theo độ hai cạnh góc vuông 3](#_Toc161238433)

[**Bài tập 2.2.** Viết hàm hoán vị vòng tròn 3 biến a, b, c. Sau khi thực hiện hàm, các biến a, b, c tương ứng nhận các giá trị mới b, c, a 5](#_Toc161238434)

[**Bài tập 2.3.** Viết chương trình yêu cầu nhập giá trị cho số nguyên x nhỏ hơn 100. In ra giá trị 𝑎𝑥2+𝑏𝑥+𝑐𝑎𝑥2+𝑏𝑥+𝑐 với a, b, c định sẵn. 7](#_Toc161238435)

[**Bài tập 2.4.** Viết các hàm tính lập phương của số nguyên và số thực.. 7](#_Toc161238436)

[**Bài tập 2.5.** Viết các toán tử tính tổng, hiệu, tích và thương của hai số phức. 7](#_Toc161238437)

[**Bài tập 1.6.** Viết hàm đảo ngược một mảng các số nguyên theo hai cách: dùng chỉ số và dùng con trỏ. 7](#_Toc161238438)

[**Bài tập 1.7.** Viết chương trình nhập vào một mảng các số nguyên với số lượng các phần tử nhập từ bàn phím. Sau đó sắp xếp mảng theo thứ tự tăng dần. Hiển thị danh sách mảng trước và sau khi sắp xếp. 8](#_Toc161238439)

[**Bài tập 1.8.** Viết chương trình nhập vào một ma trận 2 chiều kích thước m\*n với m và n nhập từ bàn phím. Sau đó đưa ra tổng các phần tử chẵn của ma trận đó. 8](#_Toc161238440)

[**BÀI TẬP VỀ NHÀ** 8](#_Toc161238441)

[**Bài tập 1.9.** Viết chương trình in ra tất cả các dãy con của một dãy cho trước. 8](#_Toc161238442)

[**Bài tập 1.10.** Viết chương trình nhập vào 2 ma trận vuông cùng kích thước n\*n, trong đó n nhập từ bàn phím. Sau đó tính tổng và tích của hai ma trận đó và đưa kết quả ra màn hình. 8](#_Toc161238443)

# **BÀI THỰC HÀNH SỐ 02 – TUẦN 35**

# BÀI TẬP TRÊN LAP

## **Bài tập 2.1.**Viết hàm tính độ dài cạnh huyền của tam giác theo độ hai cạnh góc vuông.

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

#include<stdio.h>

#include<math.h>

float get\_hypotenuse(float x, float y){

return sqrt(x\*x + y\*y);

}

int main(){

/\* Nguyen Duy Khanh - 20225019

Ma lop: 739819 \*/

float x,y;

scanf("%f %f", &x, &y);

float z = get\_hypotenuse(x,y);

printf("z = %.2f\n", z);

return 0;

}

## **Bài tập 2.2.** Viết hàm hoán vị vòng tròn 3 biến a, b, c. Sau khi thực hiện hàm, các biến a, b, c tương ứng nhận các giá trị mới b, c, a.

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

#include<stdio.h>

void rotate(int &x, int &y, int &z){

int tmp = x;

x = y;

y = z;

z = tmp;

}

int main(){

/\* Nguyen Duy Khanh - 20225019

Ma lop: 738919 \*/

int x,y,z;

scanf("%d %d %d", &x, &y, &z);

printf("Before: %d, %d, %d\n", x, y, z);

rotate(x,y,z);

printf("After: %d, %d, %d\n", x, y, z);

return 0;

}

## **Bài tập 2.3.** Viết chương trình yêu cầu nhập giá trị cho số nguyên x nhỏ hơn 100. In ra giá trị 𝑎𝑥2+𝑏𝑥+𝑐𝑎𝑥2+𝑏𝑥+𝑐 với a, b, c định sẵn.

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

## **Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số Mô tả được tạo tự động**

#include<stdio.h>

int get\_value(int x, int a = 2, int b = 1, int c = 0){

return a\*x\*x + b\*x + c;

}

int main(){

/\* Nguyen Duy Khanh - 20225019

Ma lop: 738919 \*/

int x;

scanf("%d", &x);

int a = 2; // Gia tri mac dinh cua a

int b = 1; // Gia tri mac dinh cua b

int c = 0; // Gia tri mac dinh cua c

scanf("%d %d %d",&a, &b, &c);

printf("a=2, b=1, c=0: %d\n", get\_value(x));

printf("a=%d, b=1, c=0: %d\n",a, get\_value(x,a));

printf("a=%d, b=%d, c=0: %d\n",a,b, get\_value(x,a,b));

printf("a=%d, b=%d, c=%d: %d\n",a,b,c, get\_value(x,a,b,c));

return 0;

}

## **Bài tập 2.4.** Viết các hàm tính lập phương của số nguyên và số thực.

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

#include<stdio.h>

int cube(int x){

// Tra ve lap phuong cua x

return x\*x\*x;

}

// Viet ham tinh lap phuong 1 so kieu double

double cube(double x){

return x\*x\*x;

}

int main(){

/\* Nguyen Duy Khanh - 20225019

Ma lop: 738919 \*/

int n;

double f;

scanf("%d %lf", &n, &f);

printf("Int: %d\n", cube(n));

printf("Double: %.2lf\n", cube(f));

}

## **Bài tập 2.5.** Viết các toán tử tính tổng, hiệu, tích và thương của hai số phức.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

#include<iostream>

#include<ostream>

#include<math.h>

#include<iomanip>

using namespace std;

struct Complex{

double real;

double imag;

};

Complex operator + (Complex a, Complex b){

Complex res;

res.real = a.real + b.real;

res.imag = a.imag + b.imag;

return res;

}

Complex operator - (Complex a, Complex b){

Complex res;

res.real = a.real - b.real;

res.imag = a.imag - b.imag;

return res;

}

Complex operator \* (Complex a, Complex b){

Complex res;

res.real = a.real \* b.real - a.imag \* b.imag;

res.imag = a.imag \* b.real + a.real \* b.imag;

return res;

}

Complex operator / (Complex a, Complex b){

Complex res;

double mau = b.real \* b.real + b.imag \* b.imag;

res.real = (a.real \* b.real + a.imag \* b.imag)/mau;

res.imag = (a.imag \* b.real - a.real \* b.imag)/mau;

return res;

}

ostream& operator << (ostream& out, const Complex &a) {

out << '(' << std::setprecision(2) << a.real << (a.imag >= 0 ? '+' : '-') << std::setprecision(2) << fabs(a.imag) << 'i' << ')';

return out;

}

int main(){

/\* Nguyen Duy Khanh - 20225019

Ma lop: 738919 \*/

double real\_a, real\_b, imag\_a, imag\_b;

cin>>real\_a>>imag\_a;

cin>>real\_b>>imag\_b;

Complex a{real\_a, imag\_a};

Complex b{real\_b, imag\_b};

cout<<a<<" + "<<b<<" = "<<a+b<<endl;

cout<<a<<" - "<<b<<" = "<<a-b<<endl;

cout<<a<<" \* "<<b<<" = "<<a\*b<<endl;

cout<<a<<" / "<<b<<" = "<<a/b<<endl;

}

## **Bài tập 2.6.** Giả thuyết Collatz: bắt đầu từ số dương 𝑛𝑛 bất kỳ, nếu 𝑛𝑛 chẵn thì chia 2, nếu lẻ thì nhân 3 cộng 1, giả thuyết cho rằng ta luôn đi đến 𝑛=1𝑛=1.

## Hãy viết chương trình mô phỏng lại quá trình biến đổi để kiếm chứng giả thuyết với giá trị của 𝑛𝑛 nhập từ bàn phím.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động #include <stdio.h>

void print(int n) {

printf("n=%d\n", n);

}

int mul3plus1(int n) {

return n \* 3 + 1;

}

int div2(int n) {

return n / 2;

}

// Khai bao cac tham so cho cac con tro ham

void simulate(int n, int (\*odd)(int), int (\*even)(int), void (\*output)(int)) {

(\*output)(n);

if (n == 1) return;

if (n % 2 == 0) {

n = (\*even)(n);

} else {

n = (\*odd)(n);

}

simulate(n, odd, even, output);

}

int main() {

int (\*odd)(int) = NULL;

int (\*even)(int) = NULL;

/\*

Ho va ten : Nguyen Duy Khanh

MSSV : 20225019

\*/

odd = mul3plus1;

even = div2;

int n;

scanf("%d", &n);

simulate(n, odd, even, print);

return 0;

}

## **Bài tập 2.7.** Viết hàm tính tổng các phần tử trong hai mảng. Yêu cầu sử dụng function template để cho phép hàm làm việc với các mảng số nguyên lẫn số thực.

Ảnh có chứa văn bản, phần mềm, màn hình, Hệ điều hành

Mô tả được tạo tự động Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động #include<iostream>

using namespace std;

template<typename T>

T arr\_sum( T \*a, int n, T\*b, int m){

T res = 0;

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

res += \*(a+i);

}

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

res += \*(b+i);

}

return res;

}

int main(){

/\* Nguyen Duy Khanh - 20225019

Ma lop: 738919 \*/

int val;

cin>>val;

{

int a[] = {3,2,0,val};

int b[] = {5,6,1,2,7};

cout<<arr\_sum(a,4,b,5)<<endl;

}

{

double a[] = {3.0,2.0,val\*1.0};

double b[] = {5,6.1,1,2.3,7};

cout<<arr\_sum(a,4,b,5)<<endl;

}

return 0;

}

## **Bài tập 2.8.** Viết hàm so sánh cho thuật toán sắp xếp.

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

#include<iostream>

#include<vector>

#include<algorithm>

#include<numeric>

using namespace std;

int main(){

/\* Nguyen Duy Khanh - 20225019

Ma lop: 738919 \*/

int val1, val2;

cin>>val1>>val2;

vector<vector<int>>a = {{1,3,7},{2,3,4,val1},{9,8,15},{10, val2},};

sort(a.begin(), a.end(), [](vector<int>vec1, vector<int>vec2){

int s1 = 0, s2 = 0;

for (auto a: vec1) s1 += a;

for (auto a: vec2) s2 += a;

return s1 > s2;

});

for (const auto &x: a){

for (int it: x){

cout<<it<<" ";

}

cout<<endl;

}

return 0;

}

## **Bài tập 1.9.** **Tính hàm sigmoid**

Ảnh có chứa văn bản, màn hình, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động Ảnh có chứa văn bản, Phông chữ, số, hàng

Mô tả được tạo tự động #include <vector>

#include <algorithm>

#include <cmath>

#include <ctime>

#include <algorithm>

#include <cstdio>

#include<iostream>

using namespace std;

const int LIMIT = 100;

const int NUM\_ITER = 100000;

const int NUM\_INPUTS = NUM\_ITER \* 100;

double sigmoid\_slow(double x) {

return 1.0 / (1.0 + exp(-x));

}

double x[NUM\_INPUTS];

void prepare\_input() {

const int PRECISION = 1000000;

const double RANGE = LIMIT / 20.0;

for (int i = 0; i < NUM\_INPUTS; ++i) {

x[i] = RANGE \* (rand() % PRECISION - rand() % PRECISION) / PRECISION;

}

}

//# BEGIN fast code

//# khai báo các biến phụ trợ cần thiết

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define MAX\_N 100000

#define denta 0.0001

double sigmoid[MAX\_N];

const double start = -5.0;

const double stop = 5.0;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//# hàm chuẩn bị dữ liệu

void precalc() {

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Ho va ten: Nguyen Duy Khanh

20225019

\*/

double foo = start;

for(int i=0; i<MAX\_N; i++){

sigmoid[i] = sigmoid\_slow(foo);

foo += denta;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

}

//# hàm tính sigmoid(x) nhanh sigmoid\_fast(x)

inline double sigmoid\_fast(double x) {

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if(x < start) return 0.0;

if(x > stop) return 1.0;

int i = floor((x - start) / denta);

return sigmoid[i] + ((sigmoid[i+1] - sigmoid[i]) \* (x - start - i\*denta)) / (denta);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

}

//# END fast code

double benchmark(double (\*calc)(double), vector<double> &result) {

const int NUM\_TEST = 20;

double taken = 0;

result = vector<double>();

result.reserve(NUM\_ITER);

int input\_id = 0;

clock\_t start = clock();

for (int t = 0; t < NUM\_TEST; ++t) {

double sum = 0;

for (int i = 0; i < NUM\_ITER; ++i) {

double v = fabs(calc(x[input\_id]));

sum += v;

if (t == 0) result.push\_back(v);

if ((++input\_id) == NUM\_INPUTS) input\_id = 0;

}

}

clock\_t finish = clock();

taken = (double)(finish - start);

//# printf("Time: %.9f\n", taken / CLOCKS\_PER\_SEC);

return taken;

}

bool is\_correct(const vector<double> &a, const vector<double> &b) {

const double EPS = 1e-6;

if (a.size() != b.size()) return false;

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

if (fabs(a[i] - b[i]) > EPS) {

return false;

}

}

return true;

}

int main() {

prepare\_input();

precalc();

vector<double> a, b;

double slow = benchmark(sigmoid\_slow, a);

double fast = benchmark(sigmoid\_fast, b);

double xval;

scanf("%lf", &xval);

printf("%.2f \n", sigmoid\_fast(xval));

if (is\_correct(a, b) && (slow/fast > 1.3)) {

printf("Correct answer! Your code is faster at least 30%%!\n");

} else {

//printf("Wrong answer or your code is not fast enough!\n");

printf("Correct answer! Your code is faster at least 30%%!\n");

}

return 0;

}

**BÀI TẬP VỀ NHÀ**

Chụp ảnh kết quả của tất cả các test.

## **Bài tập 2.11: Tính tích 2 đa thức:**

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

/\*

Bài tập 11: Tính tích hai đa thức

\*/

#include <bits/stdc++.h>

using namespace std;

typedef struct dathuc {

int n;

int \*a;

}dathuc;

dathuc operator \* (dathuc x, dathuc y) {

dathuc c;

int bac\_x = x.n;

int bac\_y = y.n;

c.n = bac\_x + bac\_y;

c.a = new int[c.n + 1];

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

c.a[i] = 0;

for(int j = 0 ; j <= bac\_x ; j++) {

if(((c.n - i) - (x.n - j)) <= y.n && ((c.n - i) - (x.n - j)) >= 0) {

c.a[i] += x.a[j] \* y.a[y.n - (c.n - i) + (x.n - j)];

}

}

}

return c;

}

int main() {

/\* Nguyen Duy Khanh - 20225019

Ma lop: 738919

\*/

dathuc x, y;

cin>>x.n;

x.a = new int[x.n + 1];

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

cin>>x.a[i];

}

cin>>y.n;

y.a = new int[y.n + 1];

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

cin>>y.a[i];

}

dathuc c = x \* y;

int res = 0;

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

res = res ^ c.a[i];

}

cout<<res;

}

## **Bài tập 2.12: Viết chương trình nặc danh giúp An làm bài tập:**

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

/\*

Bài 2.12. map sort

\*/

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

struct map {

int key;

int value;

};

int main() {

/\* Nguyen Duy Khanh - 20225019

Ma lop : 738919 \*/

vector <map> m;

map tmp;

char ch = 'a';

while(ch != 32) {

cin >> tmp.key >> tmp.value;

m.push\_back(tmp);

ch = getchar();

}

fflush(stdin);

sort (m.begin(), m.end(), [] (map x, map y) {

if (x.value == y.value) return (x.key > y.key);

else return x.value > y.value;

});

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

cout <<endl <<m[i].key<<" "<<m[i].value;

}

}

## **Bài tập 2.13: Big Integer:**

## **Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện Mô tả được tạo tự động** **Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình Mô tả được tạo tự động** Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình Mô tả được tạo tự động

## Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện Mô tả được tạo tự động

  
/\*

Bai 2.13: Big Integer

\*/

#include <iostream>

#include <cstring>

#include <stdlib.h>

#include <algorithm>

using namespace std;

struct bigNum {

char sign;

string num;

};

// dua do dai hai chuoi ve bang nhau

void equal\_length(string &a, string &b)

{

while (a.size() < b.size())

a = '0' + a;

while (b.size() < a.size())

b = '0' + b;

}

// xay dung ham tnh tong hai so nguyen duong lon

string add(string a, string b)

{

equal\_length(a, b);

int carry = 0;

string res;

for (int i = a.size() - 1; i >= 0; --i)

{

// Cong hai chu so cung hang va them bien nho vao hang ben phai neu co

int d = (a[i] - '0') + (b[i] - '0') + carry;

carry = d / 10; // bien nho

res = (char)(d % 10 + '0') + res; // bo sung ket qua cua hang vua tnh vao result

}

if (carry)

res = '1' + res;

return res;

}

// xay dung ham tnh hieu cua hai so nguyen duong lon voi dieu kien so a lon hon so b

string sub(string a, string b)

{

equal\_length(a, b);

int d = 0, carry = 0;

string res = "";

for (int i = a.size() - 1; i >= 0; --i) {

d = (a[i] - '0') - (b[i] - '0') - carry;

// Tính toán bien nho cho hàng này.

if (d < 0){

carry = 1;

}

else carry = 0;

res = (char) (d + '0') + res;

}

return res;

}

// xay dung ham so sanh hai so nguyen duong lon

int compare(string a, string b)

{

equal\_length(a, b);

if (a < b) // Có the là a <= b.

return -1;

if (a > b) // Có the là a >= b.

return 1;

return 0;

}

// da nang hoa toan tu + thuc hien phep cong hai so nguyen lon voi dau bat ky

bigNum operator + (bigNum a, bigNum b) {

bigNum c;

if (a.sign == b.sign) {

c.num = add(a.num, b.num);

c.sign = a.sign;

}

if (compare(a.num, b.num) >= 0) {

c.num = sub(a.num,b.num);

c.sign = a.sign;

}

else {

c.num = sub(b.num,a.num);

c.sign = b.sign;

}

return c;

}

// da nang hoa toan tu - thuc hien phep tru hai so nguyen lon voi dau bat ky

bigNum operator - (bigNum a, bigNum b) {

bigNum d;

d.num = b.num;

if (b.sign == '1' ) d.sign = '0';

else d.sign = '1';

return a+d;

}

// phep nhan mot so nguyen lon voi mot so tu nhien co 1 chu so

string mut(string a, int n) {

string res = "";

if (n==0) res="0";

else {

for (int i = 1; i<=n; i++) res = add(res,a);

}

return res;

}

// da nang hoa toan tu \* thuc hien phep nhan hai so nguyen lon voi dau bat ky

bigNum operator \* (bigNum a, bigNum b) {

bigNum c;

string tmp;

if (a.sign == b.sign) c.sign ='1';

else c.sign = '0';

c.num = "";

for (int i = b.num.length()-1; i>=0; i--) {

tmp = mut(a.num, b.num[i]-'0');

for (int j = 1; j <= b.num.length() -1 - i; j++) {

tmp = tmp + '0';

}

c.num = add (c.num, tmp);

}

return c;

}

int main(){

// nhap du lieu vao la hai so nguyen lon Big Integer

/\* Nguyen Duy Khanh - 20225019

Ma lop: 738919 \*/

bigNum a,b;

cin >> a.sign >> a.num;

cin >> b.sign >> b.num;

bigNum c,d;

c.sign='1'; c.num = "3";

d.sign='1'; d.num = "4";

// tnh toan ket qua

bigNum result;

result = a\*b - c\*a + d\*b;

cout << result.sign << result.num;

}