

Contents

Bài thực hành số 2 – Tuần 34.....	2
Bài tập 1: Truyền tham trị	2
Bài tập 2: Truyền tham chiếu	4
Bài tập 3: Tham số ngầm định.....	5
Bài tập 4: Đa năng hóa hàm	7
Bài tập 5: Đa năng hóa toán tử	9
Bài tập 6: Con trỏ hàm	13
Bài tập 7: Khái quát hóa hàm	16
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.....	16
Bài tập 8: Sắp xếp.....	18
Bài tập 9: Tính hàm sigmoid	21
Bài tập 10 (bonus): Tính tích hai ma trận vuông.....	26
Bài tập 11: Tính tích hai đa thức	31
Bài tập 12: Map Sort	36
Bài tập 13: Big Integer.....	38

Nguyễn Hữu Hậu 20215573
Bài thực hành số 2 – Tuần 34

Bài tập 1: Truyền tham trị.

Viết hàm tính độ dài cạnh huyền của tam giác theo độ hai cạnh góc vuông.

Online Learning My Courses English (en) Nguyễn Hữu Hậu

Time left 23:06:08

Quiz navigation

1 2 3 4 5
6 7 8 9

Finish attempt ...

Question 1
Correct
Marked out of 10.00
Flag question

Bài 2.1. Viết hàm tính độ dài cạnh huyền của tam giác theo độ hai cạnh góc vuông.
For example:

Input	Result
3 4	z = 5.00

Answer: (penalty regime: 10, 20, ... %)

```
1 // Nguyen Huu Hau 20215573
2
3 #include <stdio.h>
4 #include <math.h>
5
6 float get_hypotenuse(float x, float y) {
7     float kq, x_1, y_1;
8     x_1 = pow(x,2);
9     y_1 = pow(y,2);
10    kq = sqrt(x_1+y_1);
11    return kq;
12 }
13
14 int main(){
15     float x, y;
16     scanf("%f%f", &x, &y);
17
18     float z = get_hypotenuse(x, y);
19     printf("z = %.2f\n", z);
20
21     return 0;
22 }
```

Time left 23:05:48

Precheck Check

	Input	Expected	Got	
✓	3 4	z = 5.00	z = 5.00	✓
✓	5 6	z = 7.81	z = 7.81	✓

Passed all tests! ✓

Next page

// Nguyen Huu Hau 20215573

#include <stdio.h>

#include <math.h>

```
float get_hypotenuse(float x, float y) {  
    float kq,x_1,y_1;  
    x_1 = pow(x,2);  
    y_1 = pow(y,2);  
    kq = sqrt(x_1+y_1);  
    return kq;  
  
}
```

```
int main(){  
    float x, y;  
    scanf("%f%f", &x, &y);  
  
    float z = get_hypotenuse(x, y);  
    printf("z = %.2f\n", z);  
  
    return 0;  
}  
  
// Nguyen Huu Hau 20215573
```

Bài tập 2: Truyền tham chiếu

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.

Online Learning My Courses English (en) Nguyễn Hữu Hậu

Time left 23:04:22

Question 2
Correct
Marked out of 10.00
Flag question

Bài 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.

For example:

Input	Result
3 4 5	Before: 3, 4, 5 After: 4, 5, 3

Answer: (penalty regime: 10, 20, ... %)

```

1 // Nguyen Huu Hau 20215573
2 #include <stdio.h>
3
4 void rotate(int &x, int &y, int &z) {
5     int tmp;
6     tmp = x;
7     x = y;
8     y = z;
9     z = tmp;
10 }
11 // Nguyen Huu Hau 20215573
12 int main() {
13     int x, y, z;
14     scanf("%d%d%d", &x, &y, &z);
15     printf("Before: %d, %d, %d\n", x, y, z);
16     rotate(x, y, z);
17     printf("After: %d, %d, %d\n", x, y, z);
18
19     return 0;
20 }
21 // Nguyen Huu Hau 20215573

```

Quiz navigation

1	2	3	4	5
6	7	8	9	

Finish attempt ...

Time left 23:04:06

```

7     x = y;
8     y = z;
9     z = tmp;
10 }
11 // Nguyen Huu Hau 20215573
12 int main() {
13     int x, y, z;
14     scanf("%d%d%d", &x, &y, &z);
15     printf("Before: %d, %d, %d\n", x, y, z);
16     rotate(x, y, z);
17     printf("After: %d, %d, %d\n", x, y, z);
18
19     return 0;
20 }
21 // Nguyen Huu Hau 20215573

```

Precheck Check

	Input	Expected	Got	
✓	3 4 5	Before: 3, 4, 5 After: 4, 5, 3	Before: 3, 4, 5 After: 4, 5, 3	✓
✓	5 7 9	Before: 5, 7, 9 After: 7, 9, 5	Before: 5, 7, 9 After: 7, 9, 5	✓

Passed all tests! ✓

Previous page Next page

// Nguyen Huu Hau 20215573

#include <stdio.h>

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


int tmp;

```
tmp = x;
x = y;
y = z;
z = tmp;
}
// Nguyen Huu Hau 20215573
int main() {
    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;
}
// Nguyen Huu Hau 20215573
```

Bài tập 3: Tham số ngầm định

Viết chương trình yêu cầu nhập giá trị cho số nguyên x nhỏ hơn 100. In ra giá trị ax^2+bx+c với a, b, c định sẵn.

Online Learning  My Courses English (en) Nguyễn Hữu Hậu

Question 3
Correct
Marked out of 10.00
Flag question

Time left 23:02:50

Quiz navigation
1 2 3 4 5
6 7 8 9
Finish attempt ...

Bài 2.3. Viết chương trình yêu cầu nhập giá trị cho số nguyên x nhỏ hơn 100. In ra giá trị ax^2+bx+c với a, b, c định sẵn.

For example:


Input	Result
5 3 7 8	a=2, b=1, c=0: 55 a=3, b=1, c=0: 80 a=3, b=7, c=0: 110 a=3, b=7, c=8: 118

Answer: (penalty regime: 10, 20, ... %)

```

1 // Nguyen Huu Hau 20215573
2 #include <stdio.h>
3
4 // Viết hàm get_value
5 int get_value(int x, int a = 2, int b = 1, int c = 0){ // tham so ngam dinh a = 2, b = 1, c = 0
6     return a*x*x + b*x+c;
7 }
8
9 int main(){
10     int x;
11     scanf("%d", &x);
12
13     int a = 2; // gia tri mac dinh cua a
14     int b = 1; // gia tri mac dinh cua b
15     int c = 0; // gia tri mac dinh cua c
16
17     // Nhập 3 số nguyên a, b, c từ bàn phím
18     scanf("%d %d %d",&a, &b, &c);
19

```

Online Learning  My Courses English (en) Nguyễn Hữu Hậu

Time left 23:02:36

```

10     int x;
11     scanf("%d", &x);
12
13     int a = 2; // gia tri mac dinh cua a
14     int b = 1; // gia tri mac dinh cua b
15     int c = 0; // gia tri mac dinh cua c
16
17     // Nhập 3 số nguyên a, b, c từ bàn phím
18     scanf("%d %d %d",&a, &b, &c);
19
20     printf("a=2, b=1, c=0: %d\n", get_value(x));
21     printf("a=%d, b=1, c=0: %d\n", a, get_value(x, a));
22     printf("a=%d, b=%d, c=0: %d\n", a, b, get_value(x, a, b));

```

Precheck Check

	Input	Expected	Got	
✓	5 3 7 8	a=2, b=1, c=0: 55 a=3, b=1, c=0: 80 a=3, b=7, c=0: 110 a=3, b=7, c=8: 118	a=2, b=1, c=0: 55 a=3, b=1, c=0: 80 a=3, b=7, c=0: 110 a=3, b=7, c=8: 118	✓
✓	9 -1 5 -3	a=2, b=1, c=0: 171 a=-1, b=1, c=0: -72 a=-1, b=5, c=0: -36 a=-1, b=5, c=-3: -39	a=2, b=1, c=0: 171 a=-1, b=1, c=0: -72 a=-1, b=5, c=0: -36 a=-1, b=5, c=-3: -39	✓

Passed all tests! ✓

Previous page Next page

// Nguyen Huu Hau 20215573

#include <stdio.h>

//# Viết hàm get_value

int get_value(int x, int a = 2, int b = 1, int c = 0){ // tham so ngam dinh a = 2, b = 1, c = 0

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

}

```
int main(){  
    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  
  
    //# Nhập 3 số nguyên a, b, c từ bàn phím  
    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;  
}  
// Nguyen Huu Hau 20215573
```

Bài tập 4: Đa năng hóa hàm

Viết các hàm tính lập phương của số nguyên và số thực.

Online Learning My Courses English (en) Nguyễn Hữu Hậu

IT3040 – Kỹ thuật lập trình – 2022.2

Dashboard My courses IT3040_2022.2 Thực hành kỹ thuật lập trình Chấm điểm Bài thực hành số 2

Time left 23:01:02

Question 4
Correct
Marked out of 10.00
Flag question

Bài 2.4. Viết các hàm tính lập phương của số nguyên và số thực.
For example:

Input	Result
3 5.2	Int: 27 Double: 140.61

Answer: (penalty regime: 10, 20, ... %)

```

1 // Nguyen Huu Hau 20215573
2 #include <stdio.h>
3
4 // ham tinh lap phuong kieu int
5 int cube(int x) {
6     return x*x*x;
7 }
8 // ham tinh lap phuong cua so kieu double
9 double cube( double x){
10    return x*x*x;
11 }
12 int main() {
13     int n;
14     double f;
15     scanf("%d %lf", &n, &f);
16
17     printf("Int: %d\n", cube(n));
18     printf("Double: %.2lf\n", cube(f));
19
20     return 0;
21 }
22 // Nguyen Huu Hau 20215573
    
```

Quiz navigation

1	2	3	4	5
6	7	8	9	

Finish attempt ...

Online Learning My Courses English (en) Nguyễn Hữu Hậu

Time left 23:00:49

```

7 }
8 // ham tinh lap phuong cua so kieu double
9 double cube( double x){
10    return x*x*x;
11 }
12 int main() {
13     int n;
14     double f;
15     scanf("%d %lf", &n, &f);
16
17     printf("Int: %d\n", cube(n));
18     printf("Double: %.2lf\n", cube(f));
19
20     return 0;
21 }
22 // Nguyen Huu Hau 20215573
    
```

Precheck

	Input	Expected	Got	
✓	3 5.2	Int: 27 Double: 140.61	Int: 27 Double: 140.61	✓
✓	10 7.12	Int: 1000 Double: 360.94	Int: 1000 Double: 360.94	✓

Passed all tests! ✓

Previous page

// Nguyen Huu Hau 20215573

#include <stdio.h>

// ham tinh lap phuong kieu int

int cube(int x) {

return x*x*x;

}


```
// ham tinh lap phuong cua so kieu double
```

```
double cube( double x){
```

```
    return x*x*x;
```

```
}
```

```
int main() {
```

```
    int n;
```

```
    double f;
```

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

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

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

```
    return 0;
```

```
}
```

```
// Nguyen Huu Hau 20215573
```

Bài tập 5: Đa năng hóa toán tử

Viết các toán tử tính tổng, hiệu, tích và thương của hai số phức

Online Learning | My Courses | English (en) | Time left 22:59:27 | Quiz navigation | Finish attempt ...

Question 5
Correct
Marked out of 10.00
Flag question

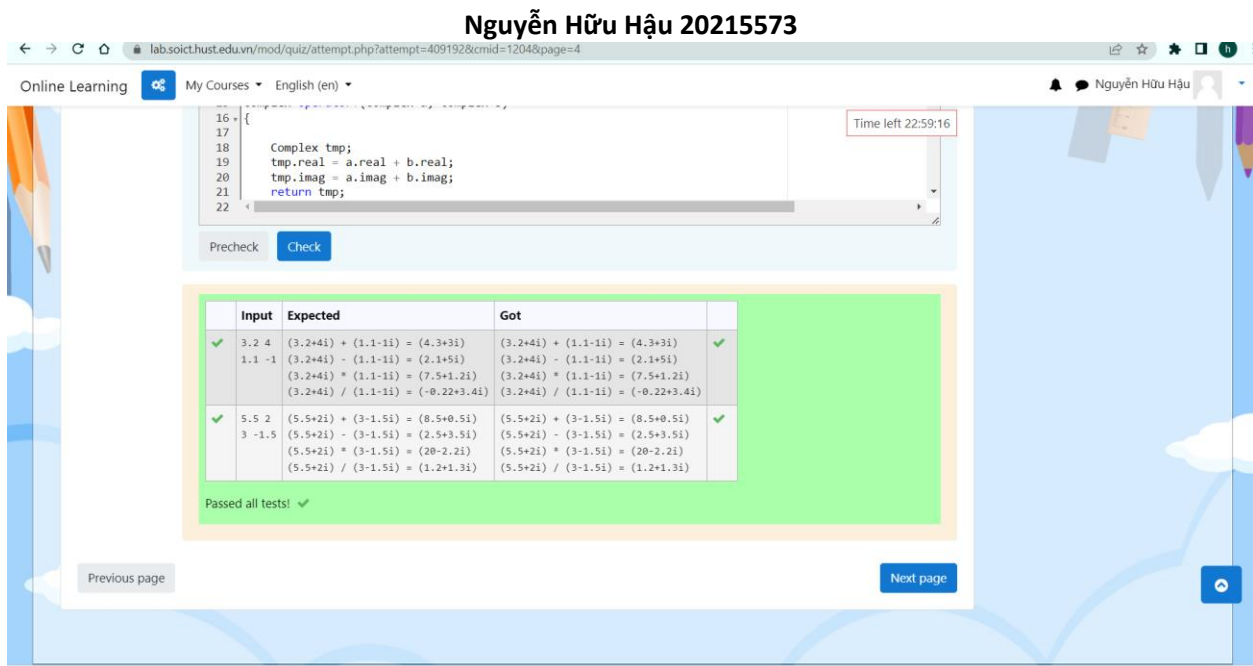
Bài 2.5. Viết các toán tử tính tổng, hiệu, tích và thương của hai số phức

For example:

Input	Result
3.2 4	$(3.2+4i) + (1.1-1i) = (4.3+3i)$
1.1 -1	$(3.2+4i) - (1.1-1i) = (2.1+5i)$
	$(3.2+4i) * (1.1-1i) = (7.5+1.2i)$
	$(3.2+4i) / (1.1-1i) = (-0.22+3.4i)$

Answer: (penalty regime: 10, 20, ... %)

```
1 // Nguyen Huu Hau 20215573
2 #include <iostream>
3 #include <ostream>
4 #include <math.h>
5 #include <iomanip>
6
7 using namespace std;
8
9 struct Complex
10 {
11     double real;
12     double imag;
13 };
14 // đa năng hóa toán tử "+"
15 Complex operator+(Complex a, Complex b)
16 {
17     Complex tmp;
18     tmp.real = a.real + b.real;
```



// Nguyen Huu Hau 20215573

```
#include <iostream>
```

```
#include <ostream>
```

```
#include <math.h>
```

```
#include <iomanip>
```

```
using namespace std;
```

```
struct Complex
```

```
{
```

```
    double real;
```

```
    double imag;
```

```
};
```

```
// da nang hoa toan tu "+"
```

```
Complex operator+(Complex a, Complex b)
```

```
{
```

```
Complex tmp;
tmp.real = a.real + b.real;
tmp.imag = a.imag + b.imag;
return tmp;
}
// Nguyen Huu Hau 20215573
// da nang hoa toan tu "-"
Complex operator - (Complex a, Complex b)
{
    Complex tmp;
    tmp.real = a.real - b.real;
    tmp.imag = a.imag - b.imag;
    return tmp;
}
// Nguyen Huu Hau 20215573
// da nang hoa toan tu "*"
Complex operator * (Complex a, Complex b)
{

    Complex tmp;
    tmp.real = a.real * b.real - a.imag * b.imag;
    tmp.imag = a.real * b.imag + a.imag * b.real;
    return tmp;
}
// da nang hoa toan tu "/"
Complex operator / (Complex a, Complex b)
{
```

```
Complex tmp;
double mauSo = pow(b.real, 2) + pow(b.imag, 2);
Complex b_1;
b_1.real = b.real;
b_1.imag = -b.imag;
tmp.real = (a.real * b_1.real - a.imag * b_1.imag) / mauSo;
tmp.imag = (a.imag * b_1.real + a.real * b_1.imag) / mauSo;
return tmp;
}
// Nguyen Huu Hau 20215573

// da nang hoa toan tu <<
ostream &operator << (ostream &out, const Complex &a)
{
    out << '(' << setprecision(2) << a.real << (a.imag >= 0 ? '+' : '-') << setprecision(2) <<
    fabs(a.imag) << 'i' << ')';
    return out;
}

int main()
{
    double real_a, real_b, img_a, img_b;
    cin >> real_a >> img_a;
    cin >> real_b >> img_b;

    Complex a{real_a, img_a};
    Complex b{real_b, img_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;

return 0;

}

// Nguyen Huu Hau 20215573
```

Bài tập 6: Con trỏ hàm

Giả thuyết Collatz: bắt đầu từ số dương n bất kỳ, nếu n chẵn thì chia 2, nếu lẻ thì nhân 3 cộng 1, giả thuyết cho rằng ta luôn đi đến $n = 1$. Hãy viết chương trình mô phỏng lại quá trình biến đổi để kiểm chứng giả thuyết với giá trị của n nhập từ bàn phím.

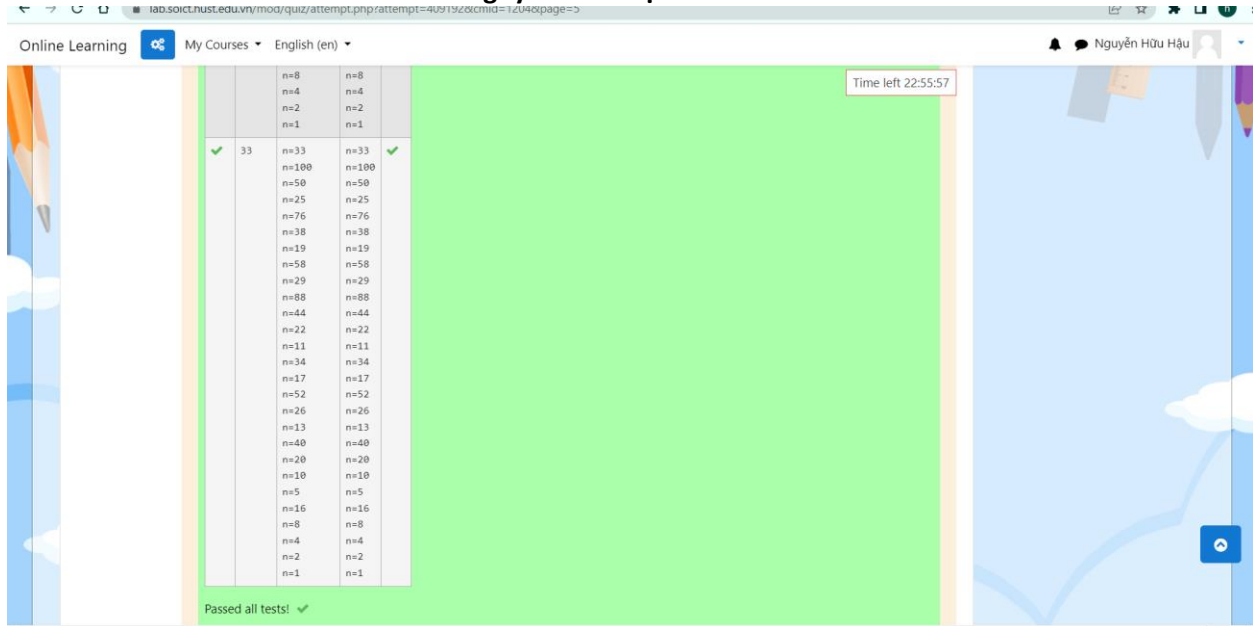
The screenshot shows a quiz interface with the following elements:

- Header:** "Online Learning" logo, "My Courses", "English (en)", and user profile "Nguyễn Hữu Hậu".
- Question 6:** Marked correct, 10.00 points. Includes a "Flag question" button.
- Question Text:**

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

Hãy viết chương trình mô phỏng lại quá trình biến đổi để kiểm chứng giả thuyết với giá trị của n nhập từ bàn phím.
- For example:** A table showing the sequence of values for $n=19$ until it reaches 1.
- Table:**

Input	Result
19	$n=19$
	$n=58$
	$n=29$
	$n=88$
	$n=44$
	$n=22$
	$n=11$
	$n=34$
	$n=17$
	$n=52$
	$n=26$
	$n=13$
	$n=40$
	$n=20$
	$n=10$
	$n=5$
	$n=16$
	$n=8$
	$n=4$
	$n=2$
	$n=1$
- Time left:** 22:56:19
- Quiz navigation:** A grid of buttons numbered 1 to 9, with button 1 highlighted.
- Finish attempt ...** button.



// Nguyen Huu Hau 20215573

```
#include <stdio.h>
```

```
//in ket qua tren cac dong khac nhau
```

```
void print(int n)
```

```
{
    printf("n=%d\n", n);
}
```

```
// n = n * 3 + 1 khi n le
```

```
int mul3plus1(int n)
```

```
{
    return n * 3 + 1;
}
```

```
// n = n / 2 khi n chan
```

```
int div2(int n)
```

```
{  
    return n / 2;  
}  
  
// khai bao cac tham so cho cac con tro ham odd, even va output  
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;  
    odd = mul3plus1;  
    even = div2;
```

```

int n;

scanf("%d", &n);

simulate(n, odd, even, print);


return 0;

}

// Nguyen Huu Hau 20215573

```

Bài tập 7: Khái quát hóa hàm

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.

Online Learning

My Courses

English (en)

Nguyễn Hữu Hậu

Time left 22:54:45

Question 7

Correct

Marked out of 10.00

Flag question

Bài 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.

For example:

Input	Result
5	31
	31.4

Answer: (penalty regime: 10, 20, ... %)

```

1 // Nguyen Huu Hau 20215573
2 #include <iostream>
3 using namespace std;
4
5 template < typename type>
6
7 // viết hàm arr_sum
8 type arr_sum(type a[],int n, type b[], int m ){
9     type sum = 0;
10    for( int i = 0; i < n; i++)
11        sum += a[i];
12    for( int j = 0 ; j < m ; j++)
13        sum += b[j];
14    return sum;
15 }
16 // Nguyen Huu Hau 20215573
17 int main() {
18     int val;
19     cin >> val;
20

```

Quiz navigation

1 2 3 4 5

6 7 8 9

Finish attempt ...

Online Learning My Courses English (en) Time left 22:54:31

```

5 template < typename type>
6
7 // viết hàm arr_sum
8 type arr_sum(type a[],int n, type b[], int m ){
9     type sum = 0;
10    for( int i = 0; i< n; i++)
11        sum += a[i];
12    for( int j = 0 ; j < m ; j++)
13        sum += b[j];
14    return sum;
15 }
16 // Nguyen Huu Hau 20215573
17 int main() {
18     int val;
19     cin >> val;
20
21     {
22         int a[] = {3, 2, 0, val};

```

Precheck Check

	Input	Expected	Got	
✓	5	31 31.4	31 31.4	✓
✓	17	43 43.4	43 43.4	✓

Passed all tests! ✓

```
// Nguyen Huu Hau 20215573
```

```
#include <iostream>
```

```
using namespace std;
```

```
template < typename type>
```

```
// viết hàm arr_sum
```

```
type arr_sum(type a[],int n, type b[], int m ){
```

```
    type sum = 0;
```

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

```
        sum += a[i];
```

```
    for( int j = 0 ; j < m ; j++)
```

```
        sum += b[j];
```

```
    return sum;
```

```
}
```

```
// Nguyen Huu Hau 20215573
```

```
int main() {
```

```
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;
}

// Nguyen Huu Hau 20215573
```

Bài tập 8: Sắp xếp

Viết hàm so sánh cho thuật toán sắp xếp.

Online Learning My Courses English (en) Time left 22:53:48

Question 8
Correct
Marked out of 10.00
Flag question

Bài 2.8. Viết hàm so sánh cho thuật toán sắp xếp.
For example:

Input	Result
-10 -5	9 8 15 1 3 7 10 -5 2 3 4 -10

Answer: (penalty regime: 10, 20, ... %)

```

1 // Nguyen Huu Hau 20215573
2 #include <iostream>
3 #include <vector>
4 #include <algorithm>
5 #include <numeric>
6
7 using namespace std;
8 // ham so sanh tong cac phan tu trong 2 vector và tra ve gia tri true/false
9 bool comparison( const vector<int> &p, const vector<int> &q){
10     int sum1 = accumulate(p.begin(), p.end(), 0);
11     int sum2 = accumulate(q.begin(), q.end(), 0);
12     return sum1>sum2 ;
13 }
14
15
16 int main() {
17     int val1, val2;
18     cin >> val1 >> val2;
19     vector< vector<int> > a = {
20

```

Precheck Check

	Input	Expected	Got	
✓	-10 -5	9 8 15 1 3 7 10 -5 2 3 4 -10	9 8 15 1 3 7 10 -5 2 3 4 -10	✓
✓	100 -100	2 3 4 100 9 8 15 1 3 7 10 -100	2 3 4 100 9 8 15 1 3 7 10 -100	✓

Passed all tests! ✓

Previous page Next page

Quiz navigation
1 2 3 4 5
6 7 8 9
Finish attempt ...

Time left 22:53:30

// Nguyen Huu Hau 20215573

#include <iostream>

#include <vector>

#include <algorithm>

#include <numeric>

using namespace std;

// ham so sanh tong cac phan tu trong 2 vector và tra ve gia tri true/false

```
bool comparison( const vector<int> &p,const vector<int>&q){  
    int sum1 = accumulate(p.begin(), p.end(), 0);  
    int sum2 = accumulate(q.begin(), q.end(), 0);  
    return sum1>sum2 ;  
}
```

```
int main() {  
    int val1, val2;  
    cin >> val1 >> val2;  
    vector< vector<int> > a = {  
        {1, 3, 7},  
        {2, 3, 4, val1},  
        {9, 8, 15},  
        {10, val2},  
    };  
  
    // sap xep cac vector trong a theo tong cac phan tu giam dan  
  
    sort (a.begin(),a.end(),comparison);  
  
    for (const auto &v : a) {  
        for (int it : v) {  
            cout << it << ' ';  
        }  
        cout << endl;  
    }
```

return 0;

}

// Nguyen Huu Hau 20215573

Bài tập 9: Tính hàm sigmoid

Dưới đây cung cấp đoạn code đơn giản để tính hàm sigmoid theo công thức trực tiếp. Hãy viết hàm tính xấp xỉ sigmoid(x) đến độ chính xác 10^{-6} và có tốc độ nhanh hơn ít nhất 30% so với code đơn giản.

Online Learning My Courses English (en) Nguyễn Hữu Hậu

Time left 22:52:06

Question 9
Correct
Marked out of 20.00
Flag question

Bài 2.9. Tính hàm sigmoid

Dưới đây cung cấp đoạn code đơn giản để tính hàm sigmoid theo công thức trực tiếp.
Hãy viết hàm tính xấp xỉ sigmoid(x) đến độ chính xác 10^{-6} và có tốc độ nhanh hơn ít nhất 30% so với code đơn giản.
Gợi ý: sử dụng kỹ thuật "chuẩn bị trước" như trong slide.

For example:

Input	Result
1.5	0.82 Correct answer! Your code is faster at least 30%!

Answer: (penalty regime: 10, 20, ... %)

```

1 // Nguyen Huu Hau 20215573
2 #include <vector>
3 #include <algorithm>
4 #include <cmath>
5 #include <ctime>
6 #include <algorithm>
7 #include <cstdio>
8 #include <iostream>
9
10 using namespace std;
11
12 // khai báo các giá trị cơ định
13 const int LIMIT = 100;
14 const int NUM_ITER = 100000;
15 const int NUM_INPUTS = NUM_ITER * 100;
16
17
18
19
20
21
22

```

Time left 22:51:50

Check

	Input	Expected	Got	
✓	1.5	0.82 Correct answer! Your code is faster at least 30%!	0.82 Correct answer! Your code is faster at least 30%!	✓
✓	2.15	0.90 Correct answer! Your code is faster at least 30%!	0.90 Correct answer! Your code is faster at least 30%!	✓

Passed all tests! ✓

Previous page Finish attempt ...

// Nguyen Huu Hau 20215573

```
#include <vector>
#include <algorithm>
#include <cmath>
#include <ctime>
#include <algorithm>
#include <cstdio>
#include <iostream>

using namespace std;

// khai bao cac gia tri co dinh
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;
    }
}
```

```
}  
  
// Nguyen Huu Hau 20215573  
  
///  
  
///  
  
///  
  
///  
  
// khai báo các biến phụ trợ cần thiết  
/*****  
  
# YOUR CODE HERE #  
*****/  
  
#define MAX_N 100000  
#define denta 0.0001  
double sigmoid[MAX_N];  
const double start = -5.0;  
const double stop = 5.0;  
// ham chuan bi du lieu  
void precalc() {  
    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  
///  
Nguyen Huu Hau 20215573

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

return 0;
}
```

// Nguyen Huu Hau 20215573

Bài tập 10 (bonus): Tính tích hai ma trận vuông

Dưới đây cung cấp đoạn code đơn giản để tính tích của hai ma trận cỡ NxN theo công thức trực tiếp. Hãy viết hàm tính tích hai ma trận nhưng có tốc độ nhanh hơn ít nhất 10% so với code đơn giản.

```
main.cpp
103     taken /= NUM_TEST;
104
105     printf("Time: %.9f\n", taken / CLOCKS_PER_SEC);
106     return taken;
107 }
108 // Nguyen Huu Hau 20215573
109 int main()
110 {
111     base = gen_random_matrix();
112
113     Matrix a, b;
114     printf("Slow version\n");
115     double slow = benchmark(multiply_naive, a);
116     printf("Fast version\n");
117     double fast = benchmark(multiply_fast, b);
118
119     if (a == b)
120     {
121         printf("Correct answer! Your code is %.2f%% faster\n", slow / fast * 100.0);
122     }
123     else
124     {
125         printf("Wrong answer!\n");
126     }
127     return 0;
128 }
129 // Nguyen Huu Hau 20215573
130
```

```
Input
Slow version
4
Time: 0.964140600
Fast version
4
Time: 0.697212200
Correct answer! Your code is 138.29% faster
```

// Nguyen Huu Hau 20215573

```
#include <iostream>
```

```
#include <cstring>

using namespace std;

const int N = 128;
// khoi tao kieu du lieu matrix
struct Matrix
{
    unsigned int mat[N][N];

    Matrix()
    {
        memset(mat, 0, sizeof mat);
    }
};

// da nang hoa toan tu ==, so sanh giua 2 ma tran
bool operator==(const Matrix &a, const Matrix &b)
{
    for (int i = 0; i < N; ++i)
    {
        for (int j = 0; j < N; ++j)
        {
            if (a.mat[i][j] != b.mat[i][j])
                return false;
        }
    }
    return true;
}
```

```
// Nguyen Huu Hau 20215573
// ham tinh tich 2 ma tran thong thuong
Matrix multiply_naive(const Matrix &a, const Matrix &b)
{
    Matrix c;
    for (int i = 0; i < N; ++i)
    {
        for (int j = 0; j < N; ++j)
        {
            for (int k = 0; k < N; ++k)
            {
                c.mat[i][j] += a.mat[i][k] * b.mat[k][j];
            }
        }
    }
    return c;
}
```

```
// Nguyen Huu Hau 20215573
// ham tinh tich 2 ma tran nhanh
Matrix multiply_fast(const Matrix &a, const Matrix &b)
{
    Matrix c;
    for (int i = 0; i < N; ++i)
    {
        for (int j = 0; j < N; ++j)
        {
            int sum = 0;
            for (int k = 0; k < N; ++k)
```

```
{
    sum += a.mat[i][k] * b.mat[k][j];
}
c.mat[i][j] = sum;
}
}
return c;
}
```

Matrix gen_random_matrix()

```
{
    Matrix a;
    for (int i = 0; i < N; ++i)
    {
        for (int j = 0; j < N; ++j)
        {
            a.mat[i][j] = rand();
        }
    }
    return a;
}
```

Matrix base;

double benchmark(Matrix (*multiply)(const Matrix &, const Matrix &), Matrix &result)

```
{
    const int NUM_TEST = 10;
    const int NUM_ITER = 64;
```

```
Matrix a = base;
result = a;

double taken = 0;
for (int t = 0; t < NUM_TEST; ++t)
{
    clock_t start = clock();
    for (int i = 0; i < NUM_ITER; ++i)
    {
        a = multiply(a, result);
        result = multiply(result, a);
    }
    clock_t finish = clock();
    taken += (double)(finish - start);
}
taken /= NUM_TEST;

printf("Time: %.9f\n", taken / CLOCKS_PER_SEC);
return taken;
}

// Nguyen Huu Hau 20215573
int main()
{
    base = gen_random_matrix();

    Matrix a, b;

    printf("Slow version\n");
```

```
double slow = benchmark(multiply_naive, a);  
printf("Fast version\n");  
double fast = benchmark(multiply_fast, b);  
  
if (a == b)  
{  
    printf("Correct answer! Your code is %.2f%% faster\n", slow / fast * 100.0);  
}  
else  
{  
    printf("Wrong answer!\n");  
}  
return 0;  
}  
  
// Nguyen Huu Hau 20215573
```

Bài tập 11: Tính tích hai đa thức

Cho 2 đa thức $A(x)$ và $B(x)$ tương ứng có bậc N và M . Hãy tính ma trận tích $C(x) = A(x) * B(x)$ có bậc $N+M-1$

```

bai11.cpp > main()
76     int res = (int)(real(n[0])+0.5);
77     for (int i=1; i<=p; i++){
78         res = res ^ (int)(real(h[i])+0.5);
79     }
80
81     return res;
82 }
83
84 int main(){
85     // freopen("input11.txt", "r", stdin);
86     // freopen("output11.txt", "w", stdout);
87     int n, m;
88     vector<base> x,y;
89     input(n,m,x,y);
90     cout << multiPly(n,m,x,y);
91 }
92 // Nguyen Huu Hau 20215573

```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL

```

what(): std::bad_alloc
PS D:\tailieu\20215573_NguyenHuuHau_TH02_Tuan34> cd "d:\tailieu\20215573_NguyenHuuHau_TH02_Tuan34\" ; if ($?) { g++ bai11.cpp -o bai11 } ;
if ($?) { .\bai11 }
PS D:\tailieu\20215573_NguyenHuuHau_TH02_Tuan34> cd "d:\tailieu\20215573_NguyenHuuHau_TH02_Tuan34\" ; if ($?) { g++ bai11.cpp -o bai11 } ;
if ($?) { .\bai11 }
9 21 62 27 90 59 63 26 40 26 72
8 36 11 68 67 29 82 30 62 23
5196
PS D:\tailieu\20215573_NguyenHuuHau_TH02_Tuan34>

```

```

bai7.cpp bai8.cpp bai9.cpp bai10.cpp 4 bai17.cpp bai11.cpp input11.txt Bai12.cpp
bai11.cpp > main()
76     int res = (int)(real(n[0])+0.5);
77     for (int i=1; i<=p; i++){
78         res = res ^ (int)(real(h[i])+0.5);
79     }
80
81     return res;
82 }
83
84 int main(){
85     // freopen("input11.txt", "r", stdin);
86     // freopen("output11.txt", "w", stdout);
87     int n, m;
88     vector<base> x,y;
89     input(n,m,x,y);
90     cout << multiPly(n,m,x,y);
91 }
92 // Nguyen Huu Hau 20215573

```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL

```

if ($?) { .\bai11 }
99 67 35 29 2 22 58 69 67 93 56 11 42 29 73 21 19 84 37 98 24 15 70 13 26 91 80 56 73 62 70 96 81 5 25 84 27 36 5 46 29 13 57 24 95 82 45 1
4 67 34 64 43 50 87 8 76 78 88 84 3 51 54 99 32 60 76 68 39 12 26 86 94 39 95 70 34 78 67 1 97 2 17 92 52 56 1 80 86 41 65 89 44 19 40 29 3
1 17 97 71 81 75
98 9 27 67 56 97 53 86 65 6 83 19 24 28 71 32 29 3 19 70 68 8 15 40 49 96 23 18 45 46 51 21 55 79 88 64 28 41 50 93 0 34 64 24 14 87 56 43
91 27 65 59 36 32 51 37 28 75 7 74 21 58 95 29 37 35 93 18 28 43 11 28 29 76 4 43 63 13 38 6 40 4 18 28 88 69 17 17 96 24 43 70 83 90 99 72
25 44 90 5
125190
PS D:\tailieu\20215573_NguyenHuuHau_TH02_Tuan34>

```

// Nguyen Huu Hau 20215573

#include<bits/stdc++.h>

using namespace std;


```
typedef complex<double> base;
typedef vector<base> vb;

void input(int &n, int &m, vb &x, vb &y){
    cin >> n;
    for(int i=0; i<=n; i++){
        int tmp;
        cin >> tmp;
        base mycomplex(tmp,0);
        x.push_back(mycomplex);
    }

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

        base mycomplex(tmp,0);
        y.push_back(mycomplex);
    }
}

void fft(vb & a, bool revert){
    int n = (int)a.size();
```

```

for(int i=1, j=0; i<n; ++i){
    int bit = n >> 1;
    while(j>=bit){
        j =j- bit;
    }
    bit=bit>>1;

    j =j+ bit;
    if (i < j)
        swap (a[i], a[j]);
}

for(int len=2; len<=n; len<=1){
    double ang = 2*M_PI/len;
    if(revert==1) ang=-ang;
    complex<double> wlen (cos(ang), sin(ang));
    for (int i=0; i<n; i+=len) {
        complex<double> w (1);
        for (int j=0; j<len/2; ++j) {
            complex<double> u = a[i+j], v = a[i+j+len/2] * w;
            a[i+j] = u + v;
            a[i+j+len/2] = u - v;
            w =w * wlen;
        }
    }
}

if(revert==1)

```

```

        for(int i=0; i<n; ++i)
            a[i] /=n;
    }

int multiPly(int n, int m, vb x, vb y){
    int p = 1;
    while(p < max (n, m)) p=p<<1;
    p=p<<1;
    x.resize(p);
    y.resize(p);

    fft(x, false);
    fft(y, false);

    vector<base> h(p);
    for (int i=0; i<p; i++)
        h[i] =x[i]*y[i];

    fft (h, true);

    int res = (int)(real(h[0])+0.5);
    for (int i=1; i<=p; i++){
        res = res ^ (int)(real(h[i])+0.5);
    }

    return res;
}

```

```

}

int main(){
    // freopen("input11.txt", "r", stdin);
    // freopen("output11.txt", "w", stdout);

    int n, m;

    vector<base> x,y;

    input(n,m,x,y);

    cout << multiPly(n,m,x,y);
}

```

// Nguyen Huu Hau 20215573

Bài tập 12: Map Sort

Hôm nay, cô giáo giao cho An một câu hỏi học búa. Cô cho một danh sách với mỗi phần tử có dạng <key, value> và yêu cầu An sắp xếp danh sách đó giảm dần theo giá trị value. Nếu 2 phần tử có value giống nhau thì sắp xếp giảm dần theo key. Hãy viết một chương trình sử dụng hàm nặc danh để giúp An làm bài tập.

// Nguyen Huu Hau 20215573

```

#include<bits/stdc++.h>

using namespace std;

struct element {
    int key;
    int value;
};

```

```

vector<element> lst;

void input(){
    int tmp1, tmp2;
}

```

```

while(cin >> tmp1 && cin >> tmp2){
    element tmp;
    tmp.key = tmp1;
    tmp.value = tmp2;

    lst.push_back(tmp);
}
}

void print(){
    for(int i=0; i<lst.size(); i++){
        cout << lst[i].key << " " << lst[i].value << endl;
    }
}

int main(){
    // freopen("input12.txt", "r", stdin);
    // freopen("output12.txt", "w", stdout);
    input();
    sort(lst.begin(),lst.end(),[] (element a, element b){
        if(a.value > b.value) return true;
        else if (a.value < b.value) return false;
        else {
            return a.key >= b.key;
        }
    });
}

```

```
print();

}

// Nguyen Huu Hau 20215573
```

Bài tập 13: Big Integer

```
bai13.cpp > 2. operator-(bigNum, bigNum)
171 {
172     res.sign = '0';
173     sub(res.num, num1.num, num2.num);
174     return res;
175 }
176 else
177 {
178     res.sign = '1';
179     sub(res.num, num2.num, num1.num);
180     return res;
181 }
182 }
183 else
184 {
```

```
0121807015
1347227347
042284724910108772
PS D:\tailieu\20215573_NguyenHuuHau_TH02_Tuan34> cd "d:\tailieu\20215573_NguyenHuuHau_TH02_Tuan34\" ; if ($?) { g++ bai13.cpp -o bai13 } ; if ($?) { .\bai13 }
0800547253714
0389013676936
1311423830729145609193702
PS D:\tailieu\20215573_NguyenHuuHau_TH02_Tuan34> cd "d:\tailieu\20215573_NguyenHuuHau_TH02_Tuan34\" ; if ($?) { g++ bai13.cpp -o bai13 } ; if ($?) { .\bai13 }
1562862701008461237669505996967049208066942997054417216894858422269
1995808870599753186889299576925652087713749800604075758946132
12421824521203240690959994610073539746266906532596685834222631697937399112080780010140762062140731229
PS D:\tailieu\20215573_NguyenHuuHau_TH02_Tuan34> cd "d:\tailieu\20215573_NguyenHuuHau_TH02_Tuan34\" ; if ($?) { g++ bai13.cpp -o bai13 } ; if ($?) { .\bai13 }
12335657506361583062448207352982257795957130202802668419171402333
06461365948045416069564099230023646173885469440616221232595
063232700453674335664686358003691913808521484082405989034863646555726796680527239783370961938317781514
PS D:\tailieu\20215573_NguyenHuuHau_TH02_Tuan34>
```

```
// Nguyen Huu Hau 20215573
```

```
#include <iostream>
```

```
using namespace std;
```

```
struct bigNum
```

```
{
    char sign;
    char num[101];
};
```

```
// Hàm nhập và tiền xử lý dữ liệu
```

```
void input(bigNum &num1, bigNum &num2)
```

```
{
    string tmp;
    cin >> tmp;
    num1.sign = tmp[0];
    int lens1 = tmp.length() - 1;
    for (int i = 0; i < lens1; i++)
    {
        num1.num[100 - lens1 + i + 1] = tmp[i + 1];
    }
    for (int i = 0; i < 100 - lens1 + 1; i++)
        num1.num[i] = '0';

    cin >> tmp;
    num2.sign = tmp[0];
    int lens2 = tmp.length() - 1;
    for (int i = 0; i < lens2; i++)
    {
        num2.num[100 - lens2 + i + 1] = tmp[i + 1];
    }
    for (int i = 0; i < 100 - lens2 + 1; i++)
        num2.num[i] = '0';
}

// Hàm cộng hai số dương lớn
void add(char res[], char *num1, char *num2)
{
```

```
int c = 0;

for (int i = 100; i >= 0; i--)
{
    int tmp = (int)num1[i] - 48 + (int)num2[i] - 48 + c;
    c = tmp / 10;
    res[i] = tmp % 10 + 48;
}
}
```

// Hàm trừ 2 số dương lớn, num1 > num2

```
void sub(char res[], char *num1, char *num2)
```

```
{
    int c = 0;

    for (int i = 100; i >= 0; i--)
    {
        int tmp1 = (int)num1[i] - 48;
        int tmp2 = (int)num2[i] - 48;

        if (tmp1 >= tmp2 + c)
        {
            res[i] = tmp1 - tmp2 - c + 48;
            c = 0;
        }
        else
```



```
{
    tmp1 = tmp1 + 10;
    res[i] = tmp1 - tmp2 - c + 48;
    c = 1;
}
}
}

// Nguyen Huu Hau 20215573
// Hàm nhân 2 số dương lớn
void multi(char res[], char *num1, char *num2)
{
    // clear array res
    for (int i = 0; i < 101; i++)
        res[i] = '0';

    for (int i = 100; i >= 0; i--)
    {
        // init 1 array temp
        char tmp[101];

        // add i number 0 to last array
        int k;
        for (k = 0; k < i; k++)
            tmp[100 - k] = '0';
```

```
int c = 0, sum = 0;
for (int j = 100; j >= 0; j--)
{
    sum = ((int)num1[i] - 48) * ((int)num2[j] - 48) + c;
    tmp[k] = (sum % 10) + 48;
    c = sum / 10;
    k--;
    if (k < 0)
        break;
}

add(res, tmp, res);
}
}
```

```
// check number1 >= number2
bool check(char *num1, char *num2)
{
    int foo1, foo2;
    for (foo1 = 0; foo1 < 101; foo1++)
    {
        if (num1[foo1] != '0')
            break;
    }

    for (foo2 = 0; foo2 < 101; foo2++)
```

```
{
    if (num2[foo2] != '0')
        break;
}

if (foo1 > foo2)
    return false;
else if (foo1 < foo2)
    return true;
else
{ // foo1 == foo2
    int foo = foo1;
    while (foo < 101)
    {
        if (num1[foo] < num2[foo])
            return false;
        else if (num1[foo] > num2[foo])
            return true;
        else
        {
            foo++;
        }
    }
}

return true;
```

```
}

// overloading operator "+"
bigNum operator+(bigNum num1, bigNum num2)
{
    bigNum res;

    if (num1.sign == '1' && num2.sign == '1')
    {
        res.sign = '1';
        add(res.num, num1.num, num2.num);
        return res;
    }
    else if (num1.sign == '1' && num2.sign == '0')
    {
        if (check(num1.num, num2.num))
        {
            res.sign = '1';
            sub(res.num, num1.num, num2.num);
            return res;
        }
        else
        {
            res.sign = '0';
            sub(res.num, num2.num, num1.num);
            return res;
        }
    }
}
```

```
    }  
}  
else if (num1.sign == '0' && num2.sign == '1')  
{  
    if (check(num1.num, num2.num))  
    {  
        res.sign = '0';  
        sub(res.num, num1.num, num2.num);  
        return res;  
    }  
    else  
    {  
        res.sign = '1';  
        sub(res.num, num2.num, num1.num);  
        return res;  
    }  
}  
else  
{  
    res.sign = '0';  
    add(res.num, num1.num, num2.num);  
    return res;  
}  
}
```

bigNum operator-(bigNum num1, bigNum num2)

```
{  
    bigNum res;  
  
    if (num1.sign == '1' && num2.sign == '0')  
    {  
        num2.sign = '1';  
        res = num1 + num2;  
        return res;  
    }  
    else if (num1.sign == '1' && num2.sign == '1')  
    {  
        num2.sign = '0';  
        res = num1 + num2;  
        return res;  
    }  
    else if (num1.sign == '0' && num2.sign == '1')  
    {  
        num2.sign = '0';  
        res = num1 + num2;  
        return res;  
    }  
    else  
    {  
        num2.sign = '1';  
        res = num1 + num2;  
        return res;  
    }  
}
```

```
    }  
}  
  
bigNum operator*(bigNum num1, bigNum num2)  
{  
    bigNum res;  
  
    if (num1.sign == '1' && num2.sign == '1')  
    {  
        res.sign = '1';  
        multi(res.num, num1.num, num2.num);  
        return res;  
    }  
    else if (num1.sign == '1' && num2.sign == '0')  
    {  
        res.sign = '0';  
        multi(res.num, num1.num, num2.num);  
        return res;  
    }  
    else if (num1.sign == '0' && num2.sign == '1')  
    {  
        res.sign = '0';  
        multi(res.num, num1.num, num2.num);  
        return res;  
    }  
    else
```

```
{  
    res.sign = '1';  
    multi(res.num, num1.num, num2.num);  
    return res;  
}  
}
```

// Hàm in ra số lớn

void printBigNumber(bigNum number)

```
{  
    cout << number.sign;  
    int start;  
    for (start = 0; start < 101; start++)  
        if (number.num[start] != '0')  
            break;  
  
    for (int i = start; i < 101; i++)  
        cout << number.num[i];  
}
```

int main()

```
{  
    bigNum num1, num2;  
    input(num1, num2);  
  
    bigNum so3, so4;
```



```
so3.sign = '1', so4.sign = '1';  
for (int i = 0; i < 100; i++)  
{  
    so3.num[i] = '0';  
    so4.num[i] = '0';  
}  
so3.num[100] = 3 + 48;  
so4.num[100] = 4 + 48;  
  
bigNum res = num1 * num2 - so3 * num1 + so4 * num2;  
  
printBigNumber(res);  
}  
// Nguyen Huu Hau 20215573
```