Mục lục nội dung

[Bài thực hành số 02 – Tuần 3](#_Toc165672774)

[BÀI TẬP TRÊN LAP 3](#_Toc165672775)

[**Bài tập 2.1.**Truyền tham trị 3](#_Toc165672776)

[**Bài tập 2.2.** Truyền tham chiếu 6](#_Toc165672777)

[**Bài tập 2.3.** Tham số ngầm định 7](#_Toc165672778)

[**Bài tập 2.4.** Đa năng hóa hàm 9](#_Toc165672779)

[**Bài tập 2.5.** Đa năng hóa toán tử 11](#_Toc165672780)

[**Bài tập 2.6.** Con trỏ hàm 14](#_Toc165672781)

[**Bài tập 2.7.** Khái quát hóa hàm 18](#_Toc165672784)

[**Bài tập 2.8.** Viết hàm so sánh cho thuật toán sắp xếp. 20](#_Toc165672787)

[**Bài tập 2.9.** Tính hàm sigmoid 21](#_Toc165672788)

[**Bài tập 2.10.** Tính tích hai ma trận vuông 25](#_Toc165672789)

[**BÀI TẬP VỀ NHÀ** 29](#_Toc165672790)

[**Bài tập 2.11.** Tính tích hai đa thức 29](#_Toc165672791)

[**Bài tập 2.12.** Map Sort 34](#_Toc165672807)

[**Bài tập 2.13.** Big Integer 41](#_Toc165672823)

Mục lục hình ảnh

[Hình 1 Bài 2.1 Truyền tham trị](#_Toc161238452)

[Hình 2 Bài 2.2 Truyền tham chiếu](#_Toc161238453)

[Hình 3 Bài 2.3 Tham số ngầm định](#_Toc161238454)

[Hình 4 Bài 2.4 Đa năng hóa hàm](#_Toc161238455)

[Hình 5 Bài 2.5 Đa năng hóa toán tử](#_Toc161238456)

[Hình 6 Bài 2.6 Con trỏ hàm](#_Toc161238457)

[Hình 7 Bài 2.7 Khái quát hóa hàm](#_Toc161238458)

[Hình 8 Bài 2.8 Sắp xếp](#_Toc161238458)

[Hình 9 Bài 2.9 Tính hàm sigmoid](#_Toc161238459)

[Hình 10 Bài 2.10 Tính tích hai ma trận vuông](#_Toc161238459)

# Bài thực hành số 02 – Tuần

# BÀI TẬP TRÊN LAP

## **Bài tập 2.1.**Truyền tham trị

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

A screenshot of a computer

Description automatically generated

Hình 1 Bài 2.1

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  #include <stdio.h>  #include <math.h>  float get\_hypotenuse(float x, float y) {  /// Tra ve do dai canh huyen  return sqrt(x\*x+y\*y);  }  int main(){  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.** 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.

A screenshot of a computer

Description automatically generated

Hình 2 Bài 2.2

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  #include <stdio.h>  void rotate(int &x, int &y, int &z) {  /// Tao bien tam luu cac gia tri x,y,z  int tmpx = x, tmpy = y, tmpz= z;  /// Thay doi gia tri theo yeu cau  x = tmpy;  y = tmpz;  z = tmpx;  }  int main() {  int x, y, z;  ///Nhap 3 so nguyen  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.** Tham số ngầm định

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

A screenshot of a computer

Description automatically generated

Hình 3 Bài 2.3

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  #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(){  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  //# Nhap 3 so nguyen tu ban phim  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.** Đa năng hóa hàm

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

Screens screenshot of a computer program

Description automatically generated

Hình 4 Bài 2.4

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  #include <stdio.h>  int cube(int x) {  /// tra ve lap phuong cua x  return x\*x\*x;  }  /// viet ham tinh lap phuong cua mot so kieu double  double cube(double x){  return (double)x\*x\*x;  }  int main() {  #define taskname "hello123"  if ( fopen(taskname".inp","r")){  freopen(taskname".inp","r",stdin);  freopen(taskname".out","w",stdout);  }  int n;  double f;  scanf("%d %lf", &n, &f);  printf("Int: %d\n", cube(n));  printf("Double: %.2lf\n", cube(f));  return 0;  } |

## **Bài tập 2.5.** Đa năng hóa toán tử

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

A screenshot of a computer

Description automatically generated

Hình 5 Bài 2.5

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  #include <iostream>  #include <ostream>  #include <math.h>  #include <iomanip>  using namespace std;  /// Tao cau truc complex  struct Complex {  double real;  double imag;  };  /// Dinh nghia toan tu +  Complex operator + (Complex a, Complex b) {  Complex c;  c.real = a.real + b.real;  c.imag = a.imag + b.imag;  return c;  }  /// Dinh nghia toan tu -  Complex operator - (Complex a, Complex b) {  Complex c;  c.real = a.real - b.real;  c.imag = a.imag - b.imag;  return c;  }  /// Dinh nghia toan tu \*  Complex operator \* (Complex a, Complex b) {  Complex c;  c.real = a.real\*b.real - a.imag\*b.imag;  c.imag = a.real\*b.imag + a.imag\*b.real;  return c;  }  /// Dinh nghia toan tu /  Complex operator / (Complex a, Complex b) {  Complex c;  Complex tmpb;  tmpb.real = b.real;  tmpb.imag = -b.imag;  c = a\*tmpb;  c.real = c.real/(b.real\*b.real+b.imag\*b.imag);  c.imag = c.imag/(b.real\*b.real+b.imag\*b.imag);  return c;  }  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() {  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;  } |

## **Bài tập 2.6.** 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 kiếm chứng giả thuyết với giá trị của n nhập từ bàn phím.Ví dụ mảng đầu vào là [9, -1, 4, 5, 7] thì kết quả là [7, 5, 4, -1, 9].

A screenshot of a computer program

Description automatically generated

A screenshot of a math test

Description automatically generated

Hình 6 Bài 2.6

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  #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 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;  } |

## **Bài tập 2.7.** 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.Yêu cầu chỉ sử dụng con trỏ để truy cập mảng, không truy cập theo index mảng.

A screenshot of a computer program

Description automatically generated

Hình 7 Bài 2.7

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  #include <iostream>  using namespace std;  template <class T>  T arr\_sum(T a[], int m, T b[], int n){  T ans=0;  for ( int i=0 ; i<m ; i++ )  ans = ans + a[i];  for ( int i=0 ; i<n ; i++ )  ans = ans + b[i];  return ans;  }  int main()  {  #define taskname "hello123"  if ( fopen(taskname".inp","r")){  freopen(taskname".inp","r",stdin);  freopen(taskname".out","w",stdout);  }  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.

A screenshot of a computer program

Description automatically generated

Hình 8 Bài 2.8

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  #include <iostream>  #include <vector>  #include <algorithm>  #include <numeric>  using namespace std;  int main()  {  #define taskname "hello123"  if ( fopen(taskname".inp","r") )  freopen(taskname".inp","r",stdin),  freopen(taskname".out","w",stdout);  int val1, val2;  cin >> val1 >> val2;  vector< vector<int> > a =  {  {1, 3, 7},  {2, 3, 4, val1},  {9, 8, 15},  {10, val2},  };  //# sắp xếp các vector trong a theo tổng các phần tử giảm dần  auto compareFuntion = [] (vector<int> &a, vector<int> &b)->bool{  int sum1=0, sum2=0;  for ( int x : a )  sum1 += x;  for ( int x : b )  sum2 += x;  return sum1>sum2;  };  sort(a.begin(),a.end(),compareFuntion);  for (const auto &v : a)  {  for (int it : v)  {  cout << it << ' ';  }  cout << endl;  }  return 0;  } |

## **Bài tập 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.

A screenshot of a computer program

Description automatically generated A screenshot of a computer

Description automatically generated

Hình 9 Bài 2.9

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  #include <vector>  #include <algorithm>  #include <cmath>  #include <ctime>  #include <algorithm>  #include <cstdio>  using namespace std;  const int LIMIT = 100; /// gioi han tinh cua x  const int NUM\_ITER = 100000; /// so diem chuan bi  const int NUM\_INPUTS = NUM\_ITER \* 100;  /// ham tinh sigmoid \_slow: dung ham mac dinh  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 bao cac bien phu tro can thiet  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  double prepare[100001];  #define lowLIMIT -LIMIT  double const delta = 0.002; /// delta = |x(i+1)-x(i)| = 200/100000 = 0.002  //# ham chuan bi du lieu  void precalc()  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  for ( int i=0 ; i<=NUM\_ITER ; i++ )  prepare[i] = sigmoid\_slow( lowLIMIT + i\*delta );  /// tinh sigmoid cua delta x\_i  }  //# hàm tính sigmoid(x) nhanh sigmoid\_fast(x)  inline double sigmoid\_fast(double x)  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  if ( x > LIMIT ) return (double)1; /// neu x > 100 -> tra ve 1  if ( x < lowLIMIT ) return (double)0; /// neu x < -100 -> tra ve 0  int i = ( x-lowLIMIT )/delta; /// tinh i trong x[i]  double x\_i = i\*delta + lowLIMIT; /// Tim x\_i tuong ung  double tmp = ( (x-x\_i)/delta )\*(prepare[i+1]-prepare[i]);/// Khoang cach giua y(i+1) va yi la hieu cua prepare(i+1) va prepare(i)=>dua vao ti le de tim y tuong ung  return prepare[i] + tmp;  }  //# 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 (int i = 0; i < (int)a.size(); ++i)  {  if (fabs(a[i] - b[i]) > EPS)  {  return false;  }  }  return true;  }  int main()  {  #define taskname "hello123"  if ( fopen(taskname".inp","r") )  freopen(taskname".inp","r",stdin),  freopen(taskname".out","w",stdout);  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;  } |

## **Bài tập 2.10.** Tính tích hai ma trận vuông

Dưới đây cung cấp đoạn code đơn giản để tính tích của hai ma trận cỡ NxN 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.

Gợi ý: hãy để ý đến thứ tự truy cập các phần tử trong ma trận, tối ưu cache hoặc sử dụng thuật toán tốt hơn O(N3)

A screenshot of a computer

Description automatically generated

Hình 10 Bài 2.10

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  #include <iostream>  #include <cstring>  #include <ctime>  using namespace std;  const int N = 128;  struct Matrix  {  unsigned int mat[N][N];  Matrix()  {  memset(mat, 0, sizeof mat);  }  };  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;  }  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;  }  Matrix multiply\_fast(const Matrix &a, const Matrix &b)  {  Matrix ans;  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];  ans.mat[i][j] = sum;  }  }  return ans;  }  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;  }  int main()  {  #define taskname "hello123"  if ( fopen(taskname".inp","r") )  freopen(taskname".inp","r",stdin),  freopen(taskname".out","w",stdout);  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;  } |

# **BÀI TẬP VỀ NHÀ**

Chụp ảnh kết quả của tất cả các test.

## **Bài tập 2.11.** Tính tích hai đa thức

Cho 2 đa thức A(x) và B(x) tương ứng có bậc NN và MM. Hãy tính ma trận tích C(x) = A(x) \* B(x) có bậc N+M−.

Input: Gồm 2 dòng biểu diễn các đa thức A(x) và B(x), mỗi dòng

Số đầu tiên N là bậc của đa thức;

N+1 số nguyên tiếp theo, số thứ i là hệ số của xi−1.

Output: Một số nguyên duy nhất là XOR của các hệ số của đa thức C(x).

Ví dụ:

Input:

3 83 86 77 15

4 93 35 86 92 49

Output:

20731

Giải thích: các hệ số của đa thức kết quả lần lượt là 7719, 10903, 17309, 19122, 19126, 12588,5153, 735.

Giới hạn:

Các hệ số của các đa thức đầu vào có trị tuyệt đối nhỏ hơn 100.

Có 5 tests, test thứ i có bậc của các đa thức đầu vào không quá 10^i.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Hình 11 Bài 2.11

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  #include <iostream>  #include <vector>  using namespace std;  void nhapDaThuc(vector<int>& daThuc, int n)  {  daThuc.resize(n+1);  for (int i = 0; i <= n; i++)  cin >> daThuc[i];  }  vector<int> nhanDaThuc(const vector<int>& daThucA, const vector<int>& daThucB)  {  int m = daThucA.size();  int n = daThucB.size();  vector<int> ketQua(m + n - 1, 0);  for (int i = 0; i < m; i++)  for (int j = 0; j < n; ++j)  ketQua[i + j] += daThucA[i] \* daThucB[j];  return ketQua;  }  void inDaThuc(const vector<int>& daThuc)  {  int n = daThuc.size();  for (int i = 0; i < n; i++)  {  if (i == 0)  cout << daThuc[i];  else if ( daThuc[i] != 0 )  cout << " + " << daThuc[i] << "x^" << i;  }  cout << endl;  }  int xorCacHeSo(const vector<int>& heSo) {  int result = 0;  for (int i = 0; i < heSo.size(); ++i) {  result ^= heSo[i];  }  return result;  }  int main()  {  #define taskname "hello123"  if ( fopen(taskname".inp","r") )  freopen(taskname".inp","r",stdin),  freopen(taskname".out","w",stdout);  int nA, nB;  ///cout << "Nhap so he so cua da thuc A: \n";  cin >> nA;  vector<int> daThucA;  nhapDaThuc(daThucA, nA);  ///cout << "Nhap so he so cua da thuc B: \n";  cin >> nB;  vector<int> daThucB;  nhapDaThuc(daThucB, nB);  ///cout << nA << ' ' << nB << endl;  // Tính toán đa thức kết quả khi nhân hai đa thức  vector<int> ketQua = nhanDaThuc(daThucA, daThucB);  // In ra đa thức kết quả  ///cout << "Da thuc A: ";  ///inDaThuc(daThucA);  ///cout << "\nDa thuc B: ";  ///inDaThuc(daThucB);  ///cout << "\nKet qua sau khi nhan hai da thuc: ";  ///inDaThuc(ketQua);  cout << xorCacHeSo(ketQua);  return 0;  } |

## **Bài tập 2.12.** Map Sort

Hôm nay, cô giáo giao cho An một câu hỏi hóc 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.

Input: Danh sách đầu vào. Mỗi dòng ghi một cặp giá trị key, value cách nhau bởi dấu cách (|key| ≤109≤109, |value| ≤109≤109).

Output: In danh sách đã được sắp xếp theo yêu cầu. Mỗi dòng ghi một cặp giá trị key, value cách nhau bởi dấu cách.

Ví dụ:

Input:

2 3

4 8

9 1

1 8

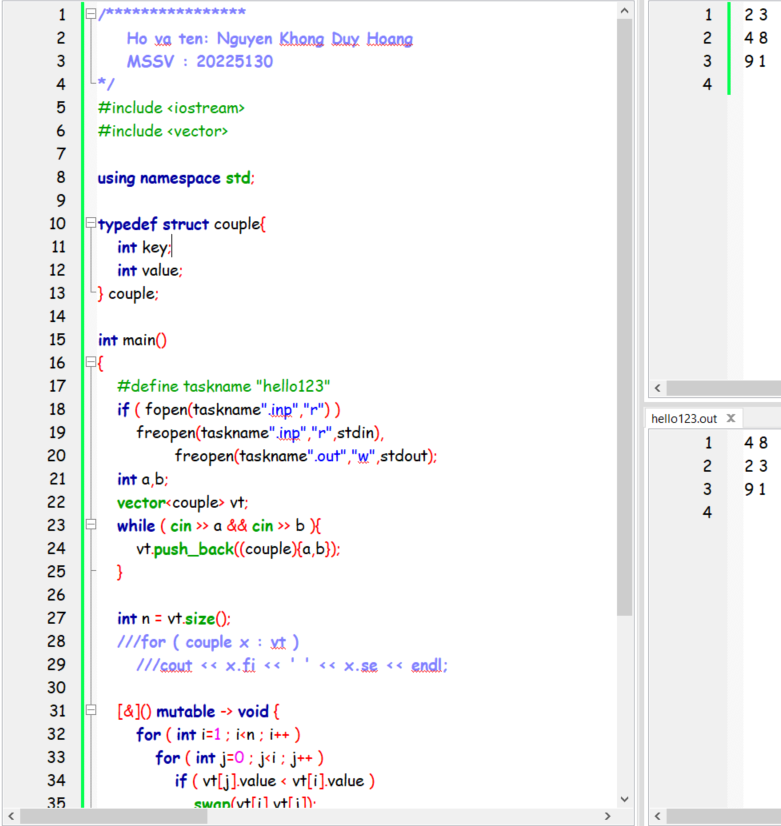
Output:

4 8

1 8

2 3

9 1



A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Hình 12 Bài 2.12

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  #include <iostream>  #include <vector>  using namespace std;  typedef struct couple{  int key;  int value;  } couple;  int main()  {  #define taskname "hello123"  if ( fopen(taskname".inp","r") )  freopen(taskname".inp","r",stdin),  freopen(taskname".out","w",stdout);  int a,b;  vector<couple> vt;  while ( cin >> a && cin >> b ){  vt.push\_back((couple){a,b});  }  int n = vt.size();  ///for ( couple x : vt )  ///cout << x.fi << ' ' << x.se << endl;  [&]() mutable -> void {  for ( int i=1 ; i<n ; i++ )  for ( int j=0 ; j<i ; j++ )  if ( vt[j].value < vt[i].value )  swap(vt[i],vt[j]);  else if ( vt[i].value == vt[j].value && vt[j].key < vt[i].key )  swap(vt[i],vt[j]);  }();  for ( auto x : vt )  cout << x.key << ' ' << x.value << endl;  return 0;  } |

## **Bài tập 2.13.** Big Integer

Số nguyên lớn là các số nguyên có giá trị rất lớn và không thể biểu diễn bằng các kiểu dữ liệu nguyên cơ bản. Để biểu diễn số nguyên lớn, ta có thể dùng kiểu struct như sau:

struct bigNum{

char sign;

char num[101];

};

Nhiệm vụ các bạn là đa năng hóa các toán tử để thực hiện các phép toán số học với kiểu dữ liệu số nguyên lớn vừa định nghĩa ở trên.

Input: Dữ liệu vào gồm hai dòng mô tả hai số nguyên lớn aa và bb, mỗi dòng chứa 1 chuỗi ký tự mô tả 1 số nguyên lớn không vượt quá 10^100. Chữ số đầu của mỗi chuỗi ký tự sẽ thể hiện dấu của số đó: 0 là âm, 1 là dương. Các chữ số sau thể hiện giá trị của số đó.

Output: In ra giá trị của biểu thức ab−3a+4b. Kết quả in ra một số nguyên lớn dưới dạng chuỗi ký tự có định dạng như mô tả trong dữ liệu vào.

Ví dụ:

Input:

0121807015

1347227347

Output:

042294724910108772

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

Hình 13 Bài 2.13

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  #include <iostream>  #include <string>  #include <algorithm>  #include <ostream>  using namespace std;  string operator + ( string a, string b )  {  string result = "";  int carry = 0;  int i = a.size() - 1, j = b.size() - 1;  while (i >= 0 || j >= 0 || carry > 0)  {  int digitA = (i >= 0) ? (a[i--] - '0') : 0;  int digitB = (j >= 0) ? (b[j--] - '0') : 0;  int sum = digitA + digitB + carry;  carry = sum / 10;  result.push\_back((sum % 10) + '0');  }  reverse(result.begin(), result.end());  return result;  }  string operator - ( string a, string b )  {  string result = "";  int borrow = 0;  int i = a.size() - 1, j = b.size() - 1;  while (i >= 0 || j >= 0)  {  int digitA = (i >= 0) ? (a[i--] - '0') : 0;  int digitB = (j >= 0) ? (b[j--] - '0') : 0;  int diff = digitA - digitB - borrow;  if (diff < 0)  {  diff += 10;  borrow = 1;  }  else  {  borrow = 0;  }  result.push\_back(diff + '0');  }  reverse(result.begin(), result.end());  return result;  }  string operator \* ( string a, string b ){  if (a == "0" || b == "0") return "0";  string result = "0";  for (int i = b.size() - 1; i >= 0; i--)  {  string temp = "";  int digitB = b[i] - '0';  int carry = 0;  for (int j = a.size() - 1; j >= 0; j--)  {  int digitA = a[j] - '0';  int product = digitA \* digitB + carry;  carry = product / 10;  temp.push\_back((product % 10) + '0');  }  if (carry > 0) temp.push\_back(carry + '0');  reverse(temp.begin(), temp.end());  for (int k = i; k < b.size() - 1; k++)  temp.push\_back('0');  result = (string)(result + temp);  }  return result;  }  int compare(string a, string b) {  while (a.size() > 1 && a[0] == '0') a.erase(0, 1);  while (b.size() > 1 && b[0] == '0') b.erase(0, 1);  int lenA = a.size();  int lenB = b.size();  if (lenA < lenB) return -1;  if (lenA > lenB) return 1;  for (int i = lenA-1; i >=0; i--) {  if (a[i] < b[i]) return -1;  if (a[i] > b[i]) return 1;  }  return 0;  }  int main()  {  #define taskname "hello123"  if ( fopen(taskname".inp","r") )  freopen(taskname".inp","r",stdin),  freopen(taskname".out","w",stdout);  string a, b;  cin >> a;  cin >> b;  int aneg = 0, bneg = 0;  if ( a[0] == '0' ) aneg = 1;  if ( b[0] == '0' ) bneg = 1;  a = a.substr(1);  b = b.substr(1);  string num3 = "3", num4 = "4", num12 = "12";  if ( compare(a, num4) == -1 && aneg == 1 ){  a = num4 - a;  aneg = 0;  }  else {  if ( aneg == 1 )  a = a - num4;  else  a = a + num4;  }  ///cout << a << " " << aneg << endl;  if ( compare(b,num3) == -1 && bneg == 0 ){  b = num3 - b;  bneg = 1;  }  else {  if ( bneg == 1 )  b = b+num3;  else  b = b-num3;  }  ///cout << b << " " << bneg << endl;  string res = a \* b;  int resneg = 0;  if ( aneg != bneg )  resneg = 1;  ///cout << "here " << resneg << ' ';  if ( compare(res,num12) == -1 && resneg == 1 ){  res = num12 - res;  resneg = 0;  }  else {  if ( resneg == 1 )  res = res - num12;  else  res = res + num12;  }  cout << 1 - resneg ;  cout << res;  return 0;  } |