**THỰC HÀNH CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT 2025**

**LAB01B**

Tên : Trịnh Tâm Như

Mssv: 3123411215

**Code**

**Bài 1**

#include <iostream>

#include <cmath>

#include <iomanip>

using namespace std;

// hàm giải phương trình bậc 2

int giaiPTBac2(float a, float b, float c, float\* x1, float\* x2) {

int sn = -1;

if (a == 0) {

if (b == 0) {

if (c == 0)

sn = -1; // Phương trình có vô số nghiệm

else

sn = 0; // Phương trình vô nghiệm

}

else {

sn = 1; // Phương trình bậc nhất bx + c = 0

\*x1 = -c / b;

}

}

else {

float delta = b \* b - 4 \* a \* c; // Tính delta

if (delta < 0) {

sn = 0; // Phương trình vô nghiệm

}

else if (delta == 0) {

sn = 1; // Phương trình có nghiệm kép

\*x1 = -b / (2 \* a);

}

else {

sn = 2; // Phương trình có 2 nghiệm phân biệt

\*x1 = (-b - sqrt(delta)) / (2 \* a);

\*x2 = (-b + sqrt(delta)) / (2 \* a);

// Sắp xếp nghiệm theo thứ tự tăng dần

if (\*x1 > \*x2) {

float tmp = \*x1;

\*x1 = \*x2;

\*x2 = tmp;

}

}

}

return sn;

}

int main() {

float a, b, c, x1, x2;

int sn;

// Nhập dữ liệu

cout << "Nhap cac he so a, b, c: ";

cin >> a >> b >> c;

// gọi sn là hàm giải của Pt bậc 2

sn = giaiPTBac2(a, b, c, &x1, &x2);

// Xử lí và xuất kết quả

if (sn == -1) {

cout << "Phuong trinh co vo so nghiem" << endl;

}

else if (sn == 0) {

cout << "Phuong trinh vo nghiem" << endl;

}

else {

cout << "Phuong trinh co " << sn << " nghiem" << endl;

cout << fixed << setprecision(2) << x1;

if (sn == 2) {

cout << " " << fixed << setprecision(2) << x2;

}

cout << endl;

}

return 0;

}

TEST CASE

|  |  |
| --- | --- |
| 1 -3 2 | Phương trình có 2 nghiệm 1.00 2.00 |
| 1 -4 4 | Phương trình có một nghiệm 4.00 |
| 1 2 5 | Phương trình vô nghiệm |
| 0 2 -4 | Phương trình có 1 nghiệm 2.00 |

**Bài 2**

#include <iostream>

#include <vector>

using namespace std;

// Hàm kiểm tra tính chẵn lẻ của dãy số

bool kiemTraChanLe(const vector<int>& arr) {

for (int i = 1; i < arr.size(); i++) {

if ((arr[i] % 2 == 0 && arr[i - 1] % 2 == 0) || (arr[i] % 2 != 0 && arr[i - 1] % 2 != 0)) {

return false; // Nếu có 2 phần tử liên tiếp cùng chẵn hoặc cùng lẻ, không thỏa mãn

}

}

return true; // Nếu tất cả các phần tử xen kẽ chẵn lẻ

}

// Hàm kiểm tra tính toàn chẵn của dãy số

bool kiemTraToanChan(const vector<int>& arr) {

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

if (arr[i] % 2 != 0) {

return false; // Nếu có phần tử lẻ, không thỏa mãn

}

}

return true; // Nếu tất cả các phần tử là số chẵn

}

int main() {

int n;

// Nhập số lượng phần tử

cout << "Moi ban nhap so luong phan tu: ";

cin >> n;

vector<int> a(n); // Khai báo vector với n phần tử

// Nhập các phần tử của dãy số

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

cout << "Phan tu " << i << ": ";

cin >> a[i];

}

// Kiểm tra các tính chất của dãy số

if (kiemTraChanLe(a)) {

cout << "+ Day co tinh chat chan le" << endl;

}

else {

cout << "+ Day khong co tinh chat chan le" << endl;

}

if (kiemTraToanChan(a)) {

cout << "+ Day co tinh chat toan chan" << endl;

}

else {

cout << "+ Day khong co tinh chat toan chan" << endl;

}

return 0;

}

**TEST CASE**

**DỮ LIỆU VÀO**

Moi ban nhap so luong phan tu: 6

Phan tu 0: 2

Phan tu 1: 3

Phan tu 2: 4

Phan tu 3: 5

Phan tu 4: 6

Phan tu 5: 7

**Kết quả mong muốn**:

+ Day co tinh chat chan le

+ Day khong co tinh chat toan chan

**DỮ LIỆU VÀO**

Moi ban nhap so luong phan tu: 4

Phan tu 0: 2

Phan tu 1: 4

Phan tu 2: 6

Phan tu 3: 8

**KẾT QUẢ**

+ Day khong co tinh chat chan le

+ Day co tinh chat toan chan

**Bài 3**

#include <iostream>

#include <vector> // Sử dụng vector để lưu trữ các ký tự

using namespace std;

// Hàm kiểm tra một ký tự có phải là nguyên âm không

bool kiemTraNguyenAm(char c) {

c = tolower(c); // Chuyển ký tự về chữ thường để đơn giản hóa việc so sánh

return (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');

}

// Hàm kiểm tra tính chất toàn chẵn của dãy ký tự

bool kiemTraToanChan(const vector<char>& arr) {

for (char c : arr) {

if (c < '0' || c > '9' || (c - '0') % 2 != 0) { // Kiểm tra xem ký tự có phải là số chẵn không

return false;

}

}

return true;

}

// Hàm kiểm tra tính chất toàn lẻ của dãy ký tự

bool kiemTraToanLe(const vector<char>& arr) {

for (char c : arr) {

if (c < '0' || c > '9' || (c - '0') % 2 == 0) { // Kiểm tra xem ký tự có phải là số lẻ không

return false;

}

}

return true;

}

int main() {

int n;

// Nhập số lượng phần tử

cout << "Moi ban nhap so luong phan tu: ";

cin >> n;

vector<char> a(n); // Khai báo vector chứa các ký tự

vector<char> b; // Khai báo vector chứa các ký tự nguyên âm

// Nhập các phần tử của dãy ký tự

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

cout << "Phan tu " << i << ": ";

cin >> a[i];

}

// Tạo dãy b chứa các ký tự nguyên âm của dãy a

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

if (kiemTraNguyenAm(a[i])) {

b.push\_back(a[i]); // Thêm ký tự nguyên âm vào dãy b

}

}

// Xuất dãy b ra màn hình

cout << "+ Day co tinh chat chan le" << endl;

// Kiểm tra và xuất thông báo về tính chất toàn chẵn hay toàn lẻ

if (kiemTraToanChan(a)) {

cout << "+ Day co tinh chat toan chan" << endl;

}

else {

cout << "+ Day khong co tinh chat toan chan" << endl;

}

return 0;

}

**TEST CASE**

**DỮ LIỆU ĐẦU VÀO**

Moi ban nhap so luong phan tu: 6

Phan tu 0: a

Phan tu 1: 2

Phan tu 2: e

Phan tu 3: 5

Phan tu 4: 7

Phan tu 5: o

**KẾT QUẢ**

+ Day co tinh chat chan le

+ Day khong co tinh chat toan chan

**DỮ LIỆU ĐẦU VÀO**

Moi ban nhap so luong phan tu: 4

Phan tu 0: 2

Phan tu 1: 4

Phan tu 2: 6

Phan tu 3: 8

**KẾT QUẢ**

+ Day co tinh chat chan le

+ Day co tinh chat toan chan

**Bài 4**

#include <iostream>

#include <vector>

using namespace std;

// Hàm hợp nhất hai dãy đã được sắp xếp tăng dần

void mergeArrays(const vector<float>& a, const vector<float>& b, vector<float>& c) {

int i = 0, j = 0;

// Hợp nhất dãy a và b vào dãy c

while (i < a.size() && j < b.size()) {

if (a[i] <= b[j]) {

c.push\_back(a[i++]);

}

else {

c.push\_back(b[j++]);

}

}

// Nếu còn phần tử trong dãy a

while (i < a.size()) {

c.push\_back(a[i++]);

}

// Nếu còn phần tử trong dãy b

while (j < b.size()) {

c.push\_back(b[j++]);

}

}

int main() {

int n, m;

// Nhập dãy a

cout << "Moi ban nhap so luong phan tu cho day a: ";

cin >> n;

vector<float> a(n);

cout << "+ Day so a" << endl;

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

cout << "Phan tu " << i << ": ";

cin >> a[i];

}

// Nhập dãy b

cout << "+ Day so b" << endl;

cout << "Moi ban nhap so luong phan tu cho day b: ";

cin >> m;

vector<float> b(m);

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

cout << "Phan tu " << i << ": ";

cin >> b[i];

}

// Dãy c chứa hợp nhất của a và b

vector<float> c;

// Hợp nhất hai dãy a và b vào dãy c

mergeArrays(a, b, c);

// Xuất dãy c ra màn hình

cout << "+ Day so c" << endl;

cout << "Day so co " << c.size() << " phan tu: ";

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

cout << c[i];

if (i != c.size() - 1) {

cout << " "; // Thêm dấu cách giữa các phần tử

}

}

cout << endl; // Đảm bảo in kết thúc dòng đúng cách

return 0;

}

**TEST CASE**

**DỮ LIỆU ĐẦU VÀO**

Moi ban nhap so luong phan tu cho day a: 0

+ Day so a

Moi ban nhap so luong phan tu cho day b: 3

+ Day so b

Phan tu 0: 1.1

Phan tu 1: 2.2

Phan tu 2: 3.3

**KẾT QUẢ**

+ Day so c

Day so co 3 phan tu: 1.1 2.2 3.3

**DỮ LIỆU ĐẦU VÀO**

Moi ban nhap so luong phan tu cho day a: 3

+ Day so a

Phan tu 0: 1.1

Phan tu 1: 3.3

Phan tu 2: 5.5

Moi ban nhap so luong phan tu cho day b: 2

+ Day so b

Phan tu 0: 2.2

Phan tu 1: 3.3

**KẾT QUẢ**

+ Day so c

Day so co 5 phan tu: 1.1 2.2 3.3 3.3 5.5

**Bài 5**

#include <iostream>

#include <vector>

using namespace std;

int main() {

int n;

// Nhập số lượng phần tử của dãy a

cout << "Moi ban nhap so luong phan tu: ";

cin >> n;

vector<int> a(n); // Dãy a chứa các số nguyên

vector<int> b; // Dãy b chứa các số chẵn

vector<int> c; // Dãy c chứa các số lẻ

// Nhập các phần tử của dãy a

cout << "+ Day so a" << endl;

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

cout << "Phan tu " << i << ": ";

cin >> a[i];

}

// Tách dãy a thành dãy b (chẵn) và dãy c (lẻ)

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

if (a[i] % 2 == 0) {

b.push\_back(a[i]); // Thêm phần tử chẵn vào dãy b

}

else {

c.push\_back(a[i]); // Thêm phần tử lẻ vào dãy c

}

}

// Xuất dãy b chứa số chẵn

cout << "+ Day so b chua so chan" << endl;

cout << "Day so co " << b.size() << " phan tu: ";

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

cout << b[i] << " ";

}

cout << endl;

// Xuất dãy c chứa số lẻ

cout << "+ Day so c" << endl;

cout << "Day so co " << c.size() << " phan tu: ";

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

cout << c[i] << " ";

}

cout << endl;

return 0;

}

**TEST CASE**

**DỮ LIỆU ĐẦU VÀO**

Moi ban nhap so luong phan tu: 5

+ Day so a

Phan tu 0: 1

Phan tu 1: 2

Phan tu 2: 3

Phan tu 3: 4

Phan tu 4: 5

**KẾT QUẢ**

+ Day so b chua so chan

Day so co 2 phan tu: 2 4

+ Day so c

Day so co 3 phan tu: 1 3 5

**DỮ LIỆU ĐẦU VÀO**

Moi ban nhap so luong phan tu: 4

+ Day so a

Phan tu 0: 2

Phan tu 1: 4

Phan tu 2: 6

Phan tu 3: 8

**KẾT QUẢ**

+ Day so b chua so chan

Day so co 4 phan tu: 2 4 6 8

+ Day so c

Day so co 0 phan tu: