

TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO

Học phần: Cấu Trúc Dữ Liệu Và Giải Thuật

Bài tập 01

Sinh viên thực hiện:

Tên SV: Vũ Bình Phước

Mã số sinh viên: 3121411171

Bài toán 1: Giải hệ phương trình bậc 2 ($ax^2 + bx + c = 0$)

Phân tích:

Main idea: Lấy a, b, c từ người dùng và in ra kết quả của phương trình.

+Input: a, b, c: float

-Output: sn, x1, x2

Với:

- sn = -1: “Phương trình vô số nghiệm”

- sn = 0: “Phương trình vô nghiệm”

- sn = 1, x1: “Phương trình có 1 nghiệm x =”

- sn = 2, x1, x2: “Phương trình có 2 nghiệm x1 =, x2=”

Quy ước: $x1 < x2$

Thiết kế:

$$\Delta = b^2 - 4ac$$

a	b	c	delta	Kết quả
0	0	0	None	sn = -1
0	0	!0	None	sn = 0
0	!0	!0	None	sn = 1, x1 = -c/n
!0	!0	!0	< 0	sn = 0
!0	!0	!0	= 0	sn = 1, x1 = -b/(2a)
!0	!0	!0	> 0	sn = 2, x1 = (-b- sqrt(delta))/(2a), X2 = (-b+ sqrt(delta))/(2a)

Cài đặt: PTBac2.cpp

```
#include <iostream>
```

```
#include <cmath>
```

```
using namespace std;
```

```
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;
```

```
            else
```

```
                sn = 0;
```

```
        }
```

```
    else { //  $bx + c = 0$ 
```

```
        sn = 1;
```

```
        x1 = -c / b;
```

```
    }
```

```
    } else {
```

```
        float delta = b * b - 4 * a * c;
```

```
        if(delta<0) sn = 0;
```

```
        else if(delta == 0) {
```

```

        sn = 1;

        x1 = -b / (2 * a);

    } else {

        sn = 2;

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

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

        if(x1>x2){

            float tmp = x1;

            x1 = x2;

            x2 = tmp;

        }

    }

}

return sn;

}

void test1(){

    float a1, b1, c1;

    int sn1; float x1, x2;


    a1=0; b1=0; c1=0; sn1 = giaiptbac2(a1, b1, c1, x1, x2);

    printf("sn=%d, x1=%g, x2=%g\n", sn1, x1, x2);

    printf("---\n");

```

```

a1=0; b1=0; c1=5; sn1 = giaiptbac2(a1, b1, c1, x1, x2);

printf("sn=%d, x1=%g, x2=%g\n", sn1, x1, x2);

printf("---\n");


a1=1; b1=-2; c1=1; sn1 = giaiptbac2(a1, b1, c1, x1, x2);

printf("sn=%d, x1=%g, x2=%g\n", sn1, x1, x2);

printf("---\n");


a1=1; b1=-3; c1=2; sn1 = giaiptbac2(a1, b1, c1, x1, x2);

printf("sn=%d, x1=%g, x2=%g\n", sn1, x1, x2);

}

```

```

int main(){

    float a, b, c;

    int sn; float x1, x2;

    test1();

    FILE *file = freopen("GiaIPTBac2.INP", "rt", stdin);

    if (!file) {

        printf("Khong the mo file GiaIPTBac2.INP\n");

        return 1;

    }
}

```

```
scanf("%f%f%f", &a, &b, &c);

sn = giaiptbac2(a, b, c, x1, x2);


if(sn == -1)

    printf("Phuong trinh co vo so nghiem\n");

else if(sn == 0)

    printf("Phuong trinh vo nghiem\n");

else {

    printf("Phuong trinh co %d nghiem\n", sn);

    printf("%.2f", x1);

    if(sn == 2)

        printf(" %.2f", x2);

    printf("\n");

}

return 0;

}
```

Kiểm thử:

Testcase 1: $a = 0, b = 0, c = 0$

-> Phương trình có vô số nghiệm

Testcase 2: $a = 0, b = 0, c = 5$

-> Phương trình vô nghiệm.

Testcase 3: $a = 1, b = -2, c = 1$

-> Phương trình có 1 nghiệm $x_1 = 1.0$.

Testcase 4: $a = 1, b = -3, c = 2$

-> Phương trình có 2 nghiệm $x_1 = 1.0, x_2 = 2.0$.

Bài toán 2: Giải hệ phương trình trùng phương ($ax^4 + bx^2 + c = 0$)

Phân tích:

Main idea: Lấy a, b, c từ người dùng và in ra kết quả của phương trình.

+Input: a, b, c: float

-Output: sn, x1, x2, x3, x4

Với:

- sn = -1: “Phương trình vô nghiệm”

- sn = 0: “Phương trình vô nghiệm”

- sn = 1, x1: “Phương trình có 1 nghiệm x =”

- sn = 2, x1, x2: “Phương trình có 2 nghiệm x1 =, x2=”

- sn = 3, x1, x2, x3: “Phương trình có 2 nghiệm x1 =, x2=, x3=”

- sn = 4, x1, x2, x3, x4: “Phương trình có 2 nghiệm x1 =, x2=, x3=, x4=”

Quy ước: $x1 < x2 < x3 < x4$

Thiết kế:

$$\text{Đặt } t = x^2 \Rightarrow x \geq 0$$

$$\text{Phương trình } \Leftrightarrow at^2 + bt + c = 0$$

Gọi hàm $\text{ptbac2}(a,b,c)$ ở bài toán 1

$$\Rightarrow \text{snt}, t1, t2$$

$$\Delta = b^2 - 4*a*c$$

snt	t1	t2	Kết quả
-1			sn = -1
0			sn = 0
1	< 0		sn = 0
1	= 0		sn = 1; x1 = 0
1	> 0		sn = 2; x1 = -sqrt(t1) x2 = sqrt(t1)
1		< 0	sn = 0
1		= 0	sn = 1; x2 = 0
1		> 0	sn = 2; x1 = -sqrt(t2) x2 = sqrt(t2)
2	= 0	> 0	sn = 3; x1 = 0 x2 = -sqrt(t2)

			$x3 = \sqrt{t2}$
2	> 0	$= 0$	$sn = 3;$ $x1 = 0$ $x2 = -\sqrt{t1}$ $x3 = \sqrt{t1}$
2	< 0	> 0	$sn = 2;$ $x1 = -\sqrt{t2}$ $x2 = \sqrt{t1}$
2	> 0	> 0	$sn = 4$ $x1 = -\sqrt{t1}$ $x2 = \sqrt{t1}$ $x3 = -\sqrt{t2}$ $x4 = \sqrt{t2}$

Cài đặt: PTTP.cpp

```
#include <iostream>
```

```
#include <cmath>
```

```
using namespace std;
```

```
void ptb2(float a, float b, float c, float &x1, float &x2, int &so_n) {
```

```
    x1 = x2 = 0;
```

```
    so_n = 0;
```

```
    if (a == 0) {
```

```
        if (b == 0) {
```

```
            if (c == 0) {
```

```
                so_n = -1; // Vô số nghiệm
```

```
            } else {
```

```
                so_n = 0; // Vô nghiệm
```

```
            }
```

```
        } else {
```

```
            so_n = 1; // Phương trình bậc nhất
```

```
            x1 = -c / b;
```

```
        }
```

```
    } else {
```

```
        float delta = b * b - 4 * a * c;
```

```
        cout << "Delta: " << delta << endl; // In ra giá trị delta để kiểm tra
```

```
        if (delta < 0) {
```

```
            so_n = 0; // Vô nghiệm
```

```

    } else if (delta == 0) {

        so_n = 1; // Nghiệm kép

        x1 = -b / (2 * a);

    } else {

        so_n = 2; // Hai nghiệm phân biệt

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

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

        if (x1 > x2) {

            swap(x1, x2);

        }

    }

}

int giaipttphuong(float a, float b, float c, float &x1, float &x2, float &x3, float
&x4) {

    float t1, t2;

    int so_nbp;

    ptb2(a, b, c, t1, t2, so_nbp);

    cout << "t1 = " << t1 << ", t2 = " << t2 << ", so_nbp = " << so_nbp << endl;

    int so_n = 0;

    if (so_nbp == -1) {

        so_n = -1;

    } else if (so_nbp == 0) {

```

```
    so_n = 0;
} else if (so_nbp == 1) {
    if (t1 > 0) {
        so_n = 2;
        x1 = -sqrt(t1);
        x2 = sqrt(t1);
    } else if (t1 == 0) {
        so_n = 1;
        x1 = 0;
    } else {
        so_n = 0;
    }
} else if (so_nbp == 2) {
    if (t1 == 0 && t2 > 0) {
        so_n = 3;
        x1 = 0;
        x2 = -sqrt(t2);
        x3 = sqrt(t2);
    } else if (t1 > 0 && t2 == 0) {
        so_n = 3;
        x1 = 0;
        x2 = -sqrt(t1);
        x3 = sqrt(t1);
    }
}
```

```

    } else if (t1 < 0 && t2 > 0) {

        so_n = 2;

        x1 = -sqrt(t2);

        x2 = sqrt(t2);

    } else if (t1 > 0 && t2 > 0) {

        so_n = 4;

        x1 = -sqrt(t1);

        x2 = sqrt(t1);

        x3 = -sqrt(t2);

        x4 = sqrt(t2);

    }

}

return so_n;

}

void testpttp() {

    float a, b, c;

    float x1 = 0, x2 = 0, x3 = 0, x4 = 0;

    int sn;

    // Test Case 1: Vô nghiệm

    a = 1; b = 0; c = 1; //  $x^4 + 1 = 0 \Rightarrow$  Vô nghiệm

    sn = giaipttpuong(a, b, c, x1, x2, x3, x4);

    printf("Test Case 1: Vô nghiệm\n");

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

```

```
printf("---\n");
```

```
// Test Case 2: 1 nghiệm
```

```
a = 1; b = 0; c = 0; //  $x^4 = 0 \Rightarrow x = 0$ 
```

```
sn = giaipttpuong(a, b, c, x1, x2, x3, x4);
```

```
printf("Test Case 2: 1 nghiệm\n");
```

```
printf("sn = %d, x1 = %.2f\n", sn, x1);
```

```
printf("---\n");
```

```
// Test Case 3: 2 nghiệm
```

```
a = 1; b = 0; c = -1; //  $x^4 - 1 = 0 \Rightarrow x = \pm 1$ 
```

```
sn = giaipttpuong(a, b, c, x1, x2, x3, x4);
```

```
printf("Test Case 3: 2 nghiệm\n");
```

```
printf("sn = %d, x1 = %.2f, x2 = %.2f\n", sn, x1, x2);
```

```
printf("---\n");
```

```
// Test Case 4: 3 nghiệm
```

```
a = 1; b = -1; c = 0; //  $x^4 + 1 = 0 \Rightarrow 3$  nghiệm
```

```
sn = giaipttpuong(a, b, c, x1, x2, x3, x4);
```

```
printf("Test Case 4: 3 nghiệm\n");
```

```
printf("sn = %d, x1 = %.2f, x2 = %.2f, x3 = %.2f\n", sn, x1, x2, x3);
```

```
printf("---\n");
```



```

// Test Case 5: 4 nghiệm

a = 1; b = -5; c = 4; //  $x^4 - 5x^2 + 4 = 0 \Rightarrow x = \pm 1, \pm 2$ 

sn = giaipttpuong(a, b, c, x1, x2, x3, x4);

printf("Test Case 5: 4 nghiệm\n");

printf("sn = %d, x1 = %.2f, x2 = %.2f, x3 = %.2f, x4 = %.2f\n", sn, x1, x2,
x3, x4);

printf("---\n");


// Test Case 6: Vô số nghiệm

a = 0; b = 0; c = 0; //  $0 = 0 \Rightarrow$  Vô số nghiệm

sn = giaipttpuong(a, b, c, x1, x2, x3, x4);

printf("Test Case 6: Vô số nghiệm\n");

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

printf("---\n");

}

```

```

int main() {

    float a, b, c;

    float x1 = 0, x2 = 0, x3 = 0, x4 = 0;

    int sn;

    testpttp();

    FILE *file = freopen("GiaiPTTrungPhuong.INP", "rt", stdin);

```

```

if (!file) {

    printf("Khong the mo file GiaiPTTrungPhuong.INP\n");

    return 1;

}

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

sn = giaipttphuong(a, b, c, x1, x2, x3, x4);

if (sn == -1) {

    cout << "Phương trình có vô số nghiệm." << endl;

} else if (sn == 0) {

    cout << "Phương trình vô nghiệm." << endl;

} else if (sn == 1) {

    cout << "Phương trình có 1 nghiệm: x = " << x1 << endl;

} else if (sn == 2) {

    cout << "Phương trình có 2 nghiệm: x1 = " << x1 << "; x2 = " << x2 <<
endl;

} else if (sn == 3) {

    cout << "Phương trình có 3 nghiệm: x1 = " << x1 << "; x2 = " << x2 << ";
x3 = " << x3 << endl;

} else if (sn == 4) {

    cout << "Phương trình có 4 nghiệm: x1 = " << x1 << "; x2 = " << x2 << ";
x3 = " << x3 << "; x4 = " << x4 << endl;

}

return 0;

}

```

Kiểm thử:

Testcase 1: $a = 1, b = 0, c = 1$

-> Phương trình vô nghiệm

Testcase 2: $a = 1, b = 0, c = 0$

-> Phương trình có 1 nghiệm $x_1 = 0$

Testcase 3: $a = 1, b = 0, c = -1$

-> Phương trình có 2 nghiệm $x_1 = -1, x_2 = 1$.

Testcase 4: $a = 1, b = -1, c = 0$

-> Phương trình có 4 nghiệm $x_1 = 0, x_2 = -1, x_3 = 1$

Testcase 5: $a = 1, b = -5, c = 4$

-> Phương trình có 4 nghiệm $x_1 = -1, x_2 = 1, x_3 = -2, x_4 = 2$

Testcase 6: $a = 0, b = 0, c = 0$

-> Phương trình có vô số nghiệm

Bài toán 3: Phần tử chung

Phân tích:

Main idea: Nhập 3 dãy số nguyên dương x, y, z từ người dùng và in ra số phần tử chung và liệt kê các giá trị chung của dãy x, y, z.

+Input: nx, ny, nz, x[] , y[] , z[]: int

-Output: count, common

Cài đặt:

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

```
#include <stdlib.h>
```

```
void nhapMangTuFile(FILE *file, int *arr, int *n) {  
  
    fscanf(file, "%d", n);  
  
    for (int i = 0; i < *n; i++) {  
  
        fscanf(file, "%d", &arr[i]);  
  
    }  
}
```

```
int compare(const void *a, const void *b) {  
  
    return (*(int *)a - *(int *)b);  
  
}
```

```
int isPresent(int *arr, int size, int value) {  
  
    for (int i = 0; i < size; i++) {
```

```

        if (arr[i] == value) {

            return 1;

        }

    }

    return 0;

}

```

```

int timPhanTuChung(int *x, int nx, int *y, int ny, int *z, int nz, int *common) {

    int count = 0;

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

        if (isPresent(y, ny, x[i]) && isPresent(z, nz, x[i]) && !isPresent(common,
count, x[i])) {

            common[count++] = x[i];

        }

    }

    return count;

}

```

```

void inKetQua(int *arr, int size) {

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

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

        printf("%d ", arr[i]);

    }

}

```

```
    printf("\n");
}

int main() {

    int nx, ny, nz;

    int x[1000], y[1000], z[1000];

    int common[1000];

    FILE *file = fopen("PTChung1.INP", "r");

    if (file == NULL) {

        printf("Khong mo duoc file!\n");

        return 1;

    }

    nhapMangTuFile(file, x, &nx);

    nhapMangTuFile(file, y, &ny);

    nhapMangTuFile(file, z, &nz);

    fclose(file);

    int count = timPhanTuChung(x, nx, y, ny, z, nz, common);

    qsort(common, count, sizeof(int), compare);
```

```
    inKetQua(common, count);  
  
    return 0;  
}
```

Kiểm thử:

Testcase 1:

5

1 2 4 5 3

4

5 6 1 4

4

5 3 5 1

-> count = 2, common = 1, 5

Testcase 2:

5

1 2 4 5 3

4

7 6 8 4

4

5 3 5 1

-> count = 0

