**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP KỸ THUẬT LẬP TRÌNH**

**SINH VIÊN THỰC HIỆN:**

**Nguyễn Duy Thuận LỚP: 24T\_Nhat2 NHÓM: 5**

**Hà Văn Việt LỚP: 24T\_Nhat2 NHÓM: 5**

**Võ Đăng Phước LỚP: 24T\_Nhat2 NHÓM: 5**

**Trương Thái Sâm LỚP: 24T\_Nhat2 NHÓM: 5**

**Lương Thanh Nhật LỚP: 24T\_Nhat2 NHÓM: 5**

**GIẢNG VIÊN HƯỚNG DẪN: ThS. Nguyễn Văn Nguyên**

**Đà Nẵng 1/1/2**

Contents

[1](#_Toc186895671)

[DANH SÁCH BÀI TẬP 6](#_Toc186895672)

[1.1. Tên đề bài 6](#_Toc186895673)

[1.2. Thuật toán 7](#_Toc186895674)

[1.3. Chương trình nguồn 7](#_Toc186895675)

[1.4. Kết quả chương trinh 7](#_Toc186895676)

[1.5. Nhận xét nếu có 8](#_Toc186895677)

[2.1 . Tên đề bài 8](#_Toc186895678)

[2.2 . Thuật toán 8](#_Toc186895679)

[2.3 . Chương trình nguồn 9](#_Toc186895680)

[2.4 . Kết quả chương trinh 9](#_Toc186895681)

[2.5 . Nhận xét nếu có 10](#_Toc186895682)

[3.1 . Tên đề bài 10](#_Toc186895683)

[3.2 . Thuật toán 10](#_Toc186895684)

[3.3 . Chương trình nguồn 10](#_Toc186895685)

[3.4 . Kết quả chương trinh 11](#_Toc186895686)

[3.5 . Nhận xét nếu có 11](#_Toc186895687)

[4.1 . Tên đề bài 11](#_Toc186895688)

[4.2 . Thuật toán 11](#_Toc186895689)

[4.3 . Chương trình nguồn 11](#_Toc186895690)

[4.4 . Kết quả chương trinh 12](#_Toc186895691)

[4.5 . Nhận xét nếu có 13](#_Toc186895692)

[5.1 . Tên đề bài 13](#_Toc186895693)

[5.2 . Thuật toán 13](#_Toc186895694)

[5.3 . Chương trình nguồn 13](#_Toc186895695)

[5.4 . Kết quả chương trinh 13](#_Toc186895696)

[5.5 . Nhận xét nếu có 14](#_Toc186895697)

[6.1 . Tên đề bài 14](#_Toc186895698)

[6.2 . Thuật toán 14](#_Toc186895699)

[6.3 . Chương trình nguồn 14](#_Toc186895700)

[6.4 . Kết quả chương trinh 15](#_Toc186895701)

[6.5 . Nhận xét nếu có 15](#_Toc186895702)

[7.1 . Tên đề bài 15](#_Toc186895703)

[7.2 . Thuật toán 15](#_Toc186895704)

[7.3 . Chương trình nguồn 15](#_Toc186895705)

[7.4 . Kết quả chương trinh 16](#_Toc186895706)

[7.5 . Nhận xét nếu có 16](#_Toc186895707)

[8.1 . Tên đề bài 16](#_Toc186895708)

[8.2 . Thuật toán 16](#_Toc186895709)

[8.3 . Chương trình nguồn 16](#_Toc186895710)

[8.4 . Kết quả chương trinh 17](#_Toc186895711)

[8.5 . Nhận xét nếu có 18](#_Toc186895712)

[9.1 . Tên đề bài 18](#_Toc186895713)

[9.2 . Thuật toán 18](#_Toc186895714)

[9.3 . Chương trình nguồn 18](#_Toc186895715)

[9.4 . Kết quả chương trinh 20](#_Toc186895716)

[9.5 . Nhận xét nếu có 20](#_Toc186895717)

[10.1 . Tên đề bài 20](#_Toc186895718)

[10.2 . Thuật toán 20](#_Toc186895719)

[10.3 . Chương trình nguồn 20](#_Toc186895720)

[10.4 . Kết quả chương trinh 21](#_Toc186895721)

[10.5 . Nhận xét nếu có 21](#_Toc186895722)

[11.1 . Tên đề bài 21](#_Toc186895723)

[11.2 . Thuật toán 21](#_Toc186895724)

[11.3 . Chương trình nguồn 21](#_Toc186895725)

[11.4 . Kết quả chương trinh 22](#_Toc186895726)

[11.5 . Nhận xét nếu có 22](#_Toc186895727)

[12.1 . Tên đề bài 22](#_Toc186895728)

[12.2 . Thuật toán 23](#_Toc186895729)

[12.3 . Chương trình nguồn 23](#_Toc186895730)

[12.4 . Kết quả chương trinh 25](#_Toc186895731)

[12.5 . Nhận xét nếu có 25](#_Toc186895732)

[13.1 . Tên đề bài 25](#_Toc186895733)

[13.2 . Thuật toán 25](#_Toc186895734)

[13.3 . Chương trình nguồn 25](#_Toc186895735)

[13.4 . Kết quả chương trinh 26](#_Toc186895736)

[13.5 . Nhận xét nếu có 26](#_Toc186895737)

[14.1 . Tên đề bài 26](#_Toc186895738)

[14.2 . Thuật toán 26](#_Toc186895739)

[14.3 . Chương trình nguồn 27](#_Toc186895740)

[14.4 . Kết quả chương trinh 27](#_Toc186895741)

[14.5 . Nhận xét nếu có 28](#_Toc186895742)

[15.1 . Tên đề bài 28](#_Toc186895743)

[15.2 . Thuật toán 28](#_Toc186895744)

[15.3 . Chương trình nguồn 28](#_Toc186895745)

[15.4 . Kết quả chương trinh 29](#_Toc186895746)

[15.5 . Nhận xét nếu có 29](#_Toc186895747)

[16.1 . Tên đề bài 29](#_Toc186895748)

[16.2 . Thuật toán 29](#_Toc186895749)

[16.3 . Chương trình nguồn 30](#_Toc186895750)

[16.4 . Kết quả chương trinh 30](#_Toc186895751)

[16.5 . Nhận xét nếu có 31](#_Toc186895752)

[17.1 . Tên đề bài 31](#_Toc186895753)

[17.2 . Thuật toán 31](#_Toc186895754)

[17.3 . Chương trình nguồn 31](#_Toc186895755)

[17.4 . Kết quả chương trinh 32](#_Toc186895756)

[17.5 . Nhận xét nếu có 33](#_Toc186895757)

[18.1 . Tên đề bài 33](#_Toc186895758)

[18.2 . Thuật toán 33](#_Toc186895759)

[18.3 . Chương trình nguồn 33](#_Toc186895760)

[18.4 . Kết quả chương trinh 34](#_Toc186895761)

[18.5 . Nhận xét nếu có 34](#_Toc186895762)

[19.1 . Tên đề bài 34](#_Toc186895763)

[19.2 . Thuật toán 35](#_Toc186895764)

[19.3 . Chương trình nguồn 35](#_Toc186895765)

[19.4 . Kết quả chương trinh 37](#_Toc186895766)

[19.5 . Nhận xét nếu có 37](#_Toc186895767)

[20.1 . Tên đề bài 37](#_Toc186895768)

[20.2 . Thuật toán 37](#_Toc186895769)

[20.3 . Chương trình nguồn 38](#_Toc186895770)

[20.4 . Kết quả chương trinh 39](#_Toc186895771)

[20.5 . Nhận xét nếu có 39](#_Toc186895772)

[21.1 . Tên đề bài 39](#_Toc186895773)

[21.2 . Thuật toán 39](#_Toc186895774)

[21.3 . Chương trình nguồn 39](#_Toc186895775)

[21.4 . Kết quả chương trinh 40](#_Toc186895776)

[21.5 . Nhận xét nếu có 41](#_Toc186895777)

[22.1 . Tên đề bài 41](#_Toc186895778)

[22.2 . Thuật toán 41](#_Toc186895779)

[22.3 . Chương trình nguồn 41](#_Toc186895780)

[22.4 . Kết quả chương trinh 42](#_Toc186895781)

[22.5 . Nhận xét nếu có 42](#_Toc186895782)

[23.1 . Tên đề bài 42](#_Toc186895783)

[23.2 . Thuật toán 42](#_Toc186895784)

[23.3 . Chương trình nguồn 43](#_Toc186895785)

[23.4 . Kết quả chương trinh 44](#_Toc186895786)

[23.5 . Nhận xét nếu có 44](#_Toc186895787)

[24.1 . Tên đề bài 44](#_Toc186895788)

[24.2 . Thuật toán 44](#_Toc186895789)

[24.3 . Chương trình nguồn 45](#_Toc186895790)

[24.4 . Kết quả chương trinh 46](#_Toc186895791)

[24.5 . Nhận xét nếu có 46](#_Toc186895792)

[25.1 . Tên đề bài 46](#_Toc186895793)

[25.2 . Thuật toán 47](#_Toc186895794)

[25.3 . Chương trình nguồn 47](#_Toc186895795)

[25.4 . Kết quả chương trinh 50](#_Toc186895796)

[25.5 . Nhận xét nếu có 50](#_Toc186895797)

[TÀI LIỆU THAM KHẢO 51](#_Toc186895798)

# DANH SÁCH BÀI TẬP

## Tên đề bài

Hãy viết chương trình có sử dụng hàm để tính lãi suất tiết kiệm. Bạn có một khoản tiền P để gửi vào quỹ tiết kiệm, mỗi tháng lãi I phần trăm.

Số tiền sau n tháng gửi sẽ là

F=P (1 + I/100)n

F chính là số tiền cả vốn lẫn lãi nhận được sau n tháng.

Tỷ số F/P thường được tính và in ra thành bảng để nhân viên ngân hàng hoặc người gửi tra cứu.

Tỷ số này được in ra thành bảng theo tỉ lệ lãi và theo số tháng. Hãy lập bảng sau:

n 5% 6% 7% 8% 9% 10%

1. 1.05000 1.06000 1.07000 1.08000 1.09000 1.10000

2 1.10250 1.12360 1.14490 1.16640 1.18810 1.21000

3 1.15763 1.19102 1.22504 1.25971 1.29503 1.33100

4 1.21551 1.26248 1.31080 1.36049 1.41158 1.46410

5 1.27628 1.33823 1.40255 1.46933 1.53862 1.61051

6 1.34010 1.41852 1.50073 1.58687 1.67710 1.77156

7 1.40710 1.50363 1.60578 1.71382 1.82804 1.94872

8 1.47746 1.59385 1.71819 1.85093 1.99256 2.14359

9 1.55133 1.68948 1.83846 1.99900 2.17189 2.35795

10 1.62889 1.79085 1.96715 2.15892 2.36736 2.59374

## Thuật toán

Hàng ngang: Dùng for để lặp từ 5 đến 10 để in ra 5% đến 10%.

Hàng dọc: Dùng for lặp từ 1 đến 10

Sau đó thêm 1 for vào để gọi hàm **cal** và tính lãi xuất của tháng đó từ 5% -> 10%

Sau đó xuống hàng và thực hiện tiếp tháng tiếp theo đến hết vòng lặp for.

## Chương trình nguồn

#include <stdio.h>

#include <math.h>

double cal(double l, int month) {

    return pow(1 + l / 100, month);

}

int main() {

    // Hang ngang

    printf(" n");

    for (double l = 5; l <= 10; l++) {

        printf("%8.0lf%%  ", l);

    }

    printf("\n");

    // Hang doc

    for (int month = 1; month <= 10; month++) {

        printf("%2d ", month);

        for (double l = 5; l <= 10; l++) {

            double c = cal(l, month);

            printf("%8.5lf   ", c);

        }

        printf("\n");

    }

    return 0;

}

## Kết quả chương trinh

n 5% 6% 7% 8% 9% 10%

1 1.05000 1.06000 1.07000 1.08000 1.09000 1.10000

2 1.10250 1.12360 1.14490 1.16640 1.18810 1.21000

3 1.15763 1.19102 1.22504 1.25971 1.29503 1.33100

4 1.21551 1.26248 1.31080 1.36049 1.41158 1.46410

5 1.27628 1.33823 1.40255 1.46933 1.53862 1.61051

6 1.34010 1.41852 1.50073 1.58687 1.67710 1.77156

7 1.40710 1.50363 1.60578 1.71382 1.82804 1.94872

8 1.47746 1.59385 1.71819 1.85093 1.99256 2.14359

9 1.55133 1.68948 1.83846 1.99900 2.17189 2.35795

10 1.62889 1.79085 1.96715 2.15892 2.36736 2.59374

## Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Tạo cây chữ số Viết chương trình hiện ra trên màn hình:

1

232

34543

4567654

567898765

67890109876

7890123210987

890123454321098

90123456765432109

0123456789876543210

## . Thuật toán

Cho n = 10 tương ứng với 10 dòng

Dùng for lặp i từ 1 đến n

Trong for đó lồng thêm 3 for nữa:

For đầu tiên lặp từ j = 1 đến n- i để tạo nên các khoảng trống tạo hình tam giác

For thứ hai lặp từ j = i đến i + i – 1 sau đó lấy phần dư của j để đảm bảo các số chỉ nằm trong khoảng từ 0 đến 9.

For thứ ba dùng lặp ngược từ j = i + i – 2 đến i lấy phần dư j tương tự giới hạn các số trong khoảng từ 0 đến 9.

Sau mỗi vòng lặp in xuống dòng và tạo hình tam giác đủ 10 dòng

## . Chương trình nguồn

#include <stdio.h>

int main () {

    int n = 10;

    for (int i = 1; i <= n; i++) {

        // Tao khoang trang

        for (int j = 1; j <= n - i; j++) {

            printf (" ");

        }

        // Tao tam giac

        for (int j = i; j < i + i; j++) {

            printf ("%d", j % 10);

        }

        // Tao tam giac

        for (int j = i + i - 2; j >=i; j--) {

            printf ("%d", j % 10);

        }

        printf ("\n");

    }

    return 0 ;

}

## . Kết quả chương trinh

1

232

34543

4567654

567898765

67890109876

7890123210987

890123454321098

90123456765432109

0123456789876543210

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Tính diện tích của một hình thang cân biết độ dài hai cạnh đáy là a và b và góc nhọn thuộc đáy lớn là α. Viết chương trình sử dụng hàm để giải bài toán này.

## . Thuật toán

Nhập vào độ dài cạnh a, b và góc nhọn α

Gọi hàm **dientich** để tính diện tích hình thang

Trong hàm ta đổi góc α từ độ về radiant, sau đó tính chiều cao h là áp dụng công thức tính diện tích hình thang để tính

## . Chương trình nguồn

#include <stdio.h>

#include <math.h>

double dientich(double a, double b, double alpha) {

    // Doi goc tu do sang rad

    double rad = alpha \* M\_PI / 180.0;

    // Tinh chieu cao h

    double h = (a - b) \* tan(rad);

    // Tinh dien tich

    double s = (a + b) \* h / 2.0;

    return s;

}

int main() {

    double a, b, alpha;

    scanf("%lf%lf%lf", &a, &b, &alpha);

    double s = dientich(a, b, alpha);

    printf("Dien tich: %.2lf\n", s);

    return 0;

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 15 8 32 | OUTPUT: 50.30 |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Cho 3 số thực x, y, z. Viết chương trình sử dụng hàm hãy tìm

a. max(x, y, z) và min(x, y, z)

b. min(x + y + z, xyz)

c. min2(x + y + z / 2, xyz) + 1

## . Thuật toán

Nhập vào 3 số thực x, y, z

Tạo hàm min max để tìm số nhỏ nhất và số lớn nhất trong 3 số x, y, z

1. Gọi hàm **min** để tìm số bé nhất
2. Gọi hàm **max** để tìm số lớn nhất
3. Tạo thêm 1 variable là m = 999999 vô cùng lớn để tìm min(x + y + z, x \* y \* z)
4. Tương tự c) gọi hàm **min** để tìm min2

## . Chương trình nguồn

#include <stdio.h>

#include <math.h>

double min(double x, double y, double z) {

    if (x < y) {

        if (x < z) return x;

        else return z;

    } else {

        if (y < z) return y;

        else return z;

    }

}

double max(double x, double y, double z) {

    if (x > y) {

        if (x > z) return x;

        else return z;

    } else {

        if (y > z) return y;

        else return z;

    }

}

int main() {

    double x,y,z;

    double m = 999999;

    scanf("%lf%lf%lf", &x, &y, &z);

    // a

    printf("%lf" , min(x,y,z));

    // b

    printf("\n%lf", max(x,y,z));

    // c

    printf("\n%lf", min(x + y + z, x\*y\*z, m));

    // d

    double result = pow(min((x + y + z) / 2, x\*y\*z, m), 2);

    printf("\n%lf", result + 1.0);

    return 0;

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 4 6 2 | OUTPUT: 2.0  6.0  12.0  37.0 |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Cho 3 số thực x, y, z. Hãy viết chương trình sử dụng hàm chỉ ra các số trong chúng thuộc khoảng (3, ).

## . Thuật toán

Cho mảng gồm 3 phần tử, sau đó nhập 3 số thực x, y, z vào mảng

Dùng For duyệt qua từng phần tử trong mảng và tìm xem phần tử nào nằm trong khoảng (3, 5) và in ra màn hình

## . Chương trình nguồn

#include <stdio.h>

int main() {

    double arr[3];

    for (int i = 0; i < 3; i++) {

        scanf("%lf", &arr[i]);

    }

    for (int i = 0; i < 3; i++) {

        if (arr[i] > 3 && arr[i] < 5) {

            printf("%lf ", arr[i]);

        }

    }

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 3.1 4.9 5 | OUTPUT: 3.1 4.9 |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Nhập vào 2 số thực a và x và số tự nhiên n. Hãy viết chương trình có sử dụng hàm để tính giá trị của biểu thức sau: ((. . .((x + a)2 + a2) +…+ a2) +a) + a với n dấu ngoặc.

## . Thuật toán

Nhập vào 2 số thực a, x và số tự nhiên n

Gọi hàm **cal**, khởi tạo ( x + a )2, sau đó dùng for lặp từ 1 đến n

Mỗi lần lặp sẽ tính ( x + a )2 + a2

Sau khi kết thúc vòng lặp sẽ cộng thêm a vào cuối

In ra kết quả

## . Chương trình nguồn

#include <stdio.h>

#include <math.h>

double cal(double x, double a, int n) {

    double result = pow(x + a, 2);

    for (int i = 1; i <= n; i++) {

        // Tinh n ngoac

        result +=  pow(a, 2);

    }

    result += a; // Cong them a vao cuoi cung

    return result;

}

int main() {

    int n;

    double a, x;

    do {

        scanf("%lf %lf %d", &x, &a, &n);

    } while (n < 1);

    double result = cal(x, a, n);

    printf("%lf", result);

    return 0;

}

## . Kết quả chương trinh

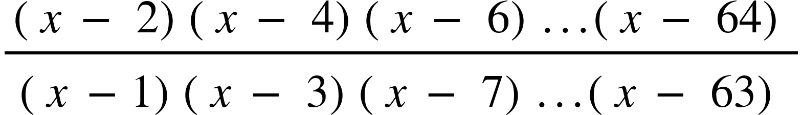
|  |  |
| --- | --- |
| INPUT: 3 6 2 | OUTPUT: 159.0 |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Nhập vào số thực x và hãy viết chương trình sử dụng hàm tính giá trị của biểu thức



## . Thuật toán

Nhập vào số thực x

Dùng For cho i = 1 lặp đến 64

{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mfrac><mn>1</mn><mfenced><mrow><mi>x</mi><mo>&#xA0;</mo><mo>-</mo><mo>&#xA0;</mo><mi>i</mi></mrow></mfenced></mfrac></mstyle></math>","origin":"MathType for Microsoft Add-in"}Nếu i là số chẵn thì sẽ lấy result =

Nếu i là số lẻ thì sẽ lấy kết quả số chẵn nhân với (x – i)

## . Chương trình nguồn

#include <stdio.h>

int main() {

    int n = 64, i;

    double result, x;

    scanf("%lf", &x);

    for (i = 1; i <= n; i++) {

        if (i % 2 != 0) {

            result = 1.0 / (x - i);

        }

        else {

            result = result \* (x - i);

        }

    }

    printf("%lf", result);

    return 0;

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 5 | OUTPUT: 1.017241 |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Giả sử n là số tự nhiên và n!! được định ngĩa như sau:

N!!=

Với n (n <= 2000) . Hãy viết chương trình sử dụng hàm để tính:

a) n!!

b) (-1)n + 1 n!!

## . Thuật toán

Nhập vào số nguyên n

Gọi hàm **giaithua** để tính n!!

Dùng For lặp ngược từ n về 0, nếu n lẻ thì sẽ lặp các số lẻ và ngược lại để tính n!!

Với (-1)n + 1 gọi hàm **cl\_giaithua** nếu n chẵn thì s = -1 nếu n chẵn và ngược lại s = 1 nếu n lẻ, sau đó dùng s nhân với hàm **giaithua** để cho ra kết quả (-1)n + 1 n!!

## . Chương trình nguồn

#include <stdio.h>

#define ll long long

// n!!

ll giaithua(int n) {

    ll result = 1;

    for (int i = n; i > 0; i -= 2) {

        result \*= i;

    }

    return result;

}

// (-1)^(n+1) \* n!!

ll cl\_giaithua(int n) {

    ll kq\_giaithua = giaithua(n);

    ll s;

    // Tinh (-1)^(n+1)

    if (n % 2 == 0) s = 1;  // Nếu n chan s = 1

    else s = -1; // Neu n le s = -1

    return s \* kq\_giaithua;

}

int main() {

    int n;

    scanf("%d", &n);

    if (n < 0 || n > 2000) return 1;

    ll kqa = giaithua(n);

    ll kqb = cl\_giaithua(n);

    printf("%lld", kqa);

    printf("\n%lld", kqb);

    return 0;

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 6 | OUTPUT: 48  -48 |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Viết chương trình giải phương trình bậc bốn trùng phương (ax4 + bx2 + c = 0, a≠0). Có đếm số nghiệm khác nhau của phương trình.

## . Thuật toán

Nhập vào các số a, b, c để tính phương trình bậc 4

Nếu a = 0, thì sẽ in ra thông báo “Khong phai phuong trinh bac 4”

Tính delta và xét các trường hợp

Tạo một variable để đếm số nghiệm khác nhau tùy vào trường hợp delta

## . Chương trình nguồn

#include <stdio.h>

#include <math.h>

int main() {

    double a, b, c;

    double y1, y2, x1, x2;

    int num = 0;

    scanf("%lf%lf%lf", &a, &b, &c);

    if (a == 0) {

        printf("Khong phai phuong trinh bac 4\n");

        return 1;

    }

    double delta = b \* b - 4 \* a \* c;

    if (delta < 0) {

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

    }

    else if (delta == 0) {

        y1 = -b / (2 \* a);

        if (y1 >= 0) {

            x1 = sqrt(y1);

            x2 = -sqrt(y1);

            if (x1 == x2) {

                num = 1;

                printf("Nghiem: %.3lf\n", x1);

            }

else {

                num = 2;

                printf("Nghiem 1: %.3lf\n", x1);

                printf("Nghiem 2: %.3lf\n", x2);

            }

        }

    }

    else {

        y1 = (-b + sqrt(delta)) / (2 \* a);

        y2 = (-b - sqrt(delta)) / (2 \* a);

        if (y1 >= 0) {

            x1 = sqrt(y1);

            x2 = -sqrt(y1);

            num += (x1 == x2) ? 1 : 2;

            printf("Nghiem 1: %.3lf\n", x1);

            if (x1 != x2) {

                printf("Nghiem 2: %.3lf\n", x2);

            }

        }

        if (y2 >= 0) {

            x1 = sqrt(y2);

            x2 = -sqrt(y2);

            num += (x1 == x2) ? 1 : 2;

            printf("Nghiem 3: %.3lf\n", x1);

            if (x1 != x2) {

                printf("Nghiem 4: %.3lf\n", x2);

            }

        }

    }

    printf("So nghiem khac nhau: %d\n", num);

    return 0;

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 1 -5 4 | OUTPUT: Nghiem 1: 2.0  Nghiem 2: -2.0  Nghiem 3: 1.0  Nghiem 4: -1.0  So nghiem khac nhau: 4 |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Viết chương trình nhập vào chuỗi số nguyên dương a và kiểm tra a có phải là chuỗi số tự đối xứng không?

Ví dụ:5,232,5775 tự đối xứng, 2342 thì không tự đối xứng. 01 là không đối xứng

## . Thuật toán

Nhập chuỗi số nguyên a

Gọi hàm **doixung** để kiểm tra xem có phải số đối xứng không

Nếu trả về False thì sẽ in ra a không phải số đối xứng, ngược lại với True.

## . Chương trình nguồn

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int doixung(char \*a) {

    int len = strlen(a);

    for (int i = 0; i < len / 2; i++) {

        if (a[i] != a[len - 1 - i]) {

            return 0; // False

        }

    }

    return 1; // True

}

int main() {

    char a[100];

    scanf("%s", a);

    if (doixung(a)) { // If true

        printf("%s la so doi xung", a);

    }

    else {

        printf("%s la so khong doi xung", a);

    }

    return 0;

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 232 | OUTPUT: 232 la so doi xung |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

.Viết chương trình tính N!!, N nhập từ bàn phím ( với 0<=N<=2000)

N chẵn: N!!=2\*4\*6\*…\*N với N lẻ: N!!=1\*3\*5\*…\*N

## . Thuật toán

Nhập vào số nguyên n và gọi hàm **giaithua** để tính n!!

Dùng For lặp ngược từ n về 0 với mỗi step -2

Nếu n chẵn thì hàm sẽ lặp i số chẵn và ngược lại sau đó nhân lần lượt với nhau để cho ra giai thừa n!!

## . Chương trình nguồn

#include <stdio.h>

#define ull unsigned long long

// n!!

ull giaithua(int n) {

    ull result = 1;

    for (int i = n; i > 0; i -= 2) {

        result \*= i;

    }

    return result;

}

int main() {

    int n;

    scanf("%d", &n);

    ull result = giaithua(n);

    printf("%llu", result);

    return 0;

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 6 | OUTPUT: 48 |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

a) Với 4 chữ số tự nhiên thuộc dãy số { 0; 1; 2; 3; 4; 5; 6; 7; 8; 9}. Hỏi lập được bao nhiêu chữ số thập phân có đủ mặt 4 chữ số đã cho mà có 2 chữ số ở phần nguyên? Hãy viết chương trình đếm bao nhiêu số thỏa mãn đề bài? Liệt kê các chữ số đó ?

b) Với 4 chữ số tự nhiên thuộc dãy số { 0; 1; 2; 3; 4; 5; 6; 7; 8; 9}. Hỏi lập được bao nhiêu chữ số thập phân có đủ mặt 4 chữ số mà có 1 chữ số ở phần nguyên và 03 chữ số phần thập phân? Hãy viết chương trình đếm bao nhiêu số thỏa mãn đề bài? Liệt kê các số đó ?

## . Thuật toán

Tạo array với dãy phần tử { 0; 1; 2; 3; 4; 5; 6; 7; 8; 9}

Dùng 4 For với điều kiện phần tử sau không trùng với phần tử trước để tạo thành dãy 4 chữ số có đủ mặt nhau

In ra dãy số với yêu cầu đề bài 2 phần nguyên,2 phần thực và 1 phần nguyên,3 phần thực .Đồng thời tạo một variable để đếm số lượng chữ số đó

## . Chương trình nguồn

**a)**

#include <stdio.h>

#include <stdbool.h>

void num() {

    int arr[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};

    int len = 10;

    int total = 0;

    for (int a = 0; a < len; a++) {

        for (int b = 0; b < len; b++) {

            if (a == b) continue;

            for (int c = 0; c < len; c++) {

                if (c == a || c == b) continue;

                for (int d = 0; d < len; d++) {

                    if (d == a || d == b || d == c) continue;

                    int pc = arr[a] \* 10 + arr[b];

                    int ps = arr[c] \* 10 + arr[d];

                    if (arr[a] != 0) {

                        printf("%d.%02d ", pc, ps);

                        total++;

                    }

                }

            }

        }

    }

    printf("\nTong so luong chu so: %d", total);

}

int main() {

    num();

    return 0;

}

**b)**

#include <stdio.h>

#include <stdbool.h>

void num() {

    int arr[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};

    int len = 10;

    int total = 0;

    for (int a = 0; a < len; a++) {

        for (int b = 0; b < len; b++) {

            if (a == b) continue;

            for (int c = 0; c < len; c++) {

                if (c == a || c == b) continue;

                for (int d = 0; d < len; d++) {

                    if (d == a || d == b || d == c) continue;

                    int pc = arr[a];

                    int ps = arr[b] \* 100 + arr[c] \* 10 + arr[d];

                    if (arr[a] != 0) {

                        printf("%d.%03d ", pc, ps);

                        total++;

                    }

                }

            }

        }

    }

    printf("\nTong so luong chu so: %d", total);

}

int main() {

    num();

    return 0;

}

## . Kết quả chương trinh

**a)** 10.23…… 98.76

Tong so luong chu so: 4536

**b)** 9.876

Tong so luong chu so: 4536

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Viết chương trình tính sin(x) triển khai theo chuỗi sau:

S = sin(x) = x / 1! – x3 / 3! + ... + (-1)n.x2n + 1 / (2n + 1)!

Số phần tử được chọn cho tới khi đạt độ chính xác: | x2n + 1 / (2n + 1)! | < EPS = 10-6

## . Thuật toán

Nhập vào giá trị x

Tạo một variable EPS = 10-6 theo điều kiện đề bài

Dùng while với điều kiện | Term | ≥ EPS

Dùng định lí Taylor trong toán cao cấp để tính sin(x)

## . Chương trình nguồn

#include <stdio.h>

#include <math.h>

#define ll long long

const double EPS = 1e-6;

double sinx(double x, double eps) {

    double term = x;

    double sum = term;

    int n = 1;

    while (fabs(term) >= eps) {

        term \*= (-1.0) \* x \* x / ((2 \* n) \* (2 \* n + 1));

        sum += term;

        n++;

    }

    return sum;

}

int main() {

    double x;

    scanf("%lf", &x);

    double result = sinx(x, EPS);

    printf("%lf", x, result);

    // So sanh voi ham chuan

    // printf("Gia tri sin(%lf) chuan: %lf\n", x, sin(x));

    return 0;

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 2 | OUTPUT: 0.909297 |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

.Viết chương trình tính cos(x) triển khai theo chuỗi số sau:

S = Cos(x) = 1 – x2 / 2! + x4 / 4! – … + (-1)n. ( (x2n ) / (2n)! )

Số phần tử được chọn cho tới khi đạt độ chính xác: | x2n / (2n)! | < EPS = 10-6

## . Thuật toán

Nhập vào giá trị x

Tạo một variable EPS = 10-6 theo điều kiện đề bài

Dùng while với điều kiện | Term | ≥ EPS

Dùng định lí Taylor trong toán cao cấp để tính cos(x)

## . Chương trình nguồn

#include <stdio.h>

#include <math.h>

#define ll long long

const double EPS = pow(10, -6);

double cosx(double x, double eps) {

    double term = 1.0;

    double sum = term;

    int n = 1;

    while (fabs(term) >= eps) {

        term \*= (-1.0) \* x \* x / (2 \* n \* (2 \* n - 1));

        sum += term;

        n++;

    }

    return sum;

}

int main() {

    double x;

    scanf("%lf", &x);

    double result = cosx(x, EPS);

    printf("%lf", x, result);

    // So sanh voi ham chuan

    // printf("Gia tri cos(%lf) chuan: %lf\n", x, cos(x));

    return 0;

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 2 | OUTPUT: -0.416147 |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Số tổ hợp có công thức truy hồi: = + với = = 1

Viết chương trình có sử dụng hàm để tính và in ra tam giác Pascal

## . Thuật toán

Nhập vào số nguyên r

Gọi hàm **tamgiac\_pascal** để dựng nên tam giác

Dùng For gán (i, k) vào hàm **cnk** để tính số tổ hợp bằng công thức truy hồi

## . Chương trình nguồn

#include <stdio.h>

int cnk(int n, int k) {

    if (k == 0 || k == n) {

        return 1;

    }

    return cnk(n - 1, k - 1) + cnk(n - 1, k);

}

void tamgiac\_pascal(int r) {

    for (int i = 0; i < r; i++) {

        for (int j = 0; j < r - i - 1; j++) {

            printf(" ");

        }

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

            printf("%d ", cnk(i, k));

        }

        printf("\n");

    }

}

int main() {

    int r;

    scanf("%d", &r);

    tamgiac\_pascal(r);

    return 0;

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 3 | OUTPUT: 1  1 1  1 2 1 |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Viết chương trình sử dụng hàm nhập số n (hệ thập phân) từ bàn phím và đổi sang cơ số c, cơ số c cũng nhập từ bàn phím ? In kết quả đổi sang cơ số c ra màn hình

## . Thuật toán

Nhập vào số nguyên n ( hệ thập phân ) và cơ số c

Gọi hàm **chuyencs** để chuyển số nguyên n sang cơ số c

Tạo 1 mảng result và index = 0 ( chỉ vị trí ban đầu trong mảng )

Lần lượt chia lấy dư của n với c và xét điều kiện nếu kết quả < 10 thì sẽ chuyển kết quả thành số bằng cách

(n % c) + ‘0’ ( ‘0’ = 48 theo mã ASCII ) gán vào mảng result và index tăng 1

Ngược lại nếu ≥ 10 thì sẽ chuyển kết quả thành kí tự A -> Z bằng cách

(n % c) - 10 + ‘A’ ( ‘A’ = 65 theo mã ACSII ) gán vào mảng result và index tăng 1

Sau đó chia lấy nguyên n để tiếp tục chia lấy dư đến khi n = 0 , dùng For đảo ngược mảng vừa lưu in ra kết quả

## . Chương trình nguồn

#include <stdio.h>

#include <stdlib.h>

void chuyencs(int n, int c) {

    char result[32];

    int index = 0;

    while (n > 0) {

        int s = n % c;

        if (s < 10) {

            result[index++] = s + '0';

        }

else {

            result[index++] = s - 10 + 'A';

        }

        n /= c;

    }

    for (int i = index - 1; i >= 0; i--) {

        printf("%c", (result[i]));

    }

}

int main() {

    int n, c;

    scanf("%d%d", &n, &c);

    chuyencs(n, c);

    return 0;

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 191 2 | OUTPUT: 10111111 |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Viết chương trình có sử dụng hàm

a. Viết chương trình nhập mảng một chiều A với n phần tử (n>=10). Xuất mảng A ra màn hình .

b. Xóa các số hoàn hảo trong A trên, xuất lại mảng A sau khi xóa các số hoàn hảo.

## . Thuật toán

Nhập số nguyên n tương ứng số lượng phần tử, sau đó nhập phần tử vào mảng và dùng For để in từng phần tử của mảng ra màn hình

Dùng For duyệt từng phần tử trong mảng và gọi hàm **sohoanhao** để kiểm tra từng phần tử trong mảng xem đâu là số hoàn hảo

Khi điều kiện bằng True thì dùng For đưa phần tử về cuối mảng sau đó giảm số lượng phần tử mảng đi 1 để xóa phần tử ở cuối mảng sau đó in ra màn hình mảng đã xóa số hoàn hảo.

## . Chương trình nguồn

#include <stdio.h>

#define ll long long

ll sohoanhao(ll n) {

    ll sum = 0;

    for (int i = 1; i < n; i++) {

        if (n % i == 0) {

            sum += i;

        }

    }

    if (sum == n) return 1; // True

    else return 0; // False

}

void nhap(ll a[], ll n ){

    for (int i = 0; i < n; i++) {

        scanf("%lld", &a[i]);

    }

}

void xuat(ll a[], ll n) {

    for (int i = 0; i < n; i++) {

        printf("%lld ", a[i]);

    }

}

int main() {

    ll n;

    scanf("%lld", &n);

    ll arr[n];

    nhap(arr, n);

    xuat(arr, n);

    printf("\n");

    // Xoa cac so hoan hoa ra khoi mang

    for (int i = 0; i < n; ) {

        if (sohoanhao(arr[i]) == 1) {

            for (int j = i; j < n - 1; j++) {

                arr[j] = arr[j + 1];

            }

            n--;

        }

        else i++; // Tranh viec bo qua phan tu

    }

    xuat(arr, n);

    return 0;

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 6  6 1 28 5 496 2 | OUTPUT: 6 1 28 5 496 2  1 5 2 |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Viết chương trình nhập mảng 1 chiều A nguyên có n phần tử có sử dụng hàm (với n nhập từ bàn phím).

a. Xuất các phần tử đã nhập ra màn hình.

b. Tìm ước số chung lớn nhất của các số trên

c. Tìm bội số chung nhỏ nhất của các số trên

## . Thuật toán

Nhập số nguyên n tương ứng số lượng phần tử mảng, nhập phần tử vào mảng sau đó in mảng đó ra màn hình

Dùng phương pháp đệ quy lồng nhau để tính UCLN và BCNN và in ra màn hình

## . Chương trình nguồn

#include<stdio.h>

#define ll long long

void nhap(ll a[], ll n ){

    for (int i = 0; i < n; i++) {

        scanf("%lld", &a[i]);

    }

}

void xuat(ll a[], ll n) {

    for (int i = 0; i < n; i++) {

        printf("%lld ", a[i]);

    }

}

ll ucln(ll a, ll b) {

    if (b == 0) return a;

    return ucln(b, a % b);

}

ll bcnn(ll a, ll b) {

    return a \* b / ucln(a, b);

}

int main() {

    ll n;

    scanf("%lld", &n);

    ll a[n];

    nhap(a, n);

    xuat(a, n);

    ll uc = a[0], bc = a[0];

    for (int i = 1; i < n; i++) {

        uc = ucln(uc, a[i]);

        bc = bcnn(bc, a[i]);

    }

    printf("\n%lld", uc);

    printf("\n%lld", bc);

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 4  6 18 12 4 | OUTPUT: 2  36 |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Viết chương trình có sử dụng hàm

a. Viết chương trình nhập mảng một chiều A với n phần tử (n>=15). Xuất mảng A ra màn hình.

b. Tính tổng các số nguyên tố trong mảng, xuất kết quả ra màn hình.

## . Thuật toán

Nhập số nguyên n tương ứng số lượng phần tử mảng, nhập phần tử vào mảng sau đó in mảng đó ra màn hình

Tạo một mảng c với số lương phần từ bằng phần tử lớn nhất của mảng arr nhằm tối ưu bộ nhớ

Áp dụng phương pháp sàn nguyên tố để giải bài toán, đặt tất cả phần tử = True tương ứng tất cả phần tử đều là số nguyên tố nhưng vị trí [0] và [1] thì bỏ qua.

Sau đó kiểm tra từng phần tử mảng arr xem có phải số nguyên tố không nếu không phải là số nguyên tố thì tại vị trí đó đánh dấu False vào mảng c.

Dùng For duyệt qua mảng với điều kiện = True, nếu phần tử nào = True thì là số nguyên tố sẽ cộng lần lượt vào variable sum để tính tổng tất cả số nguyên tố và in ra màn hình

## . Chương trình nguồn

#include <stdio.h>

#include <stdbool.h>

#define ll long long

// San so nguyen to ( O\*\*n6 )

// Python

// def snt(x):

//     c = []

//     p = [True]\*x

//     for i in range (2,x):

//         for j in range (i\*i,x,i):

//             p[j] = False

//     for i in range (2,x):

//         if p[i]: c.append(i)

//     return c

// result = []

// print(snt(n))

void inp(ll arr[], ll n ){

    for (int i = 0; i < n; i++) {

        scanf("%lld", &arr[i]);

    }

}

void out(ll arr[], ll n) {

    for (int i = 0; i < n; i++) {

        printf("%lld ", arr[i]);

    }

}

void snt(ll m, bool c[]) {

    for (int i = 0; i <= m; i++) {

        c[i] = true; // Tat cac phan tu la snt

    }

    c[0] = c[1] = false; // 0 và 1 khong phai snt

    for (int i = 2; i \* i <= m; i++) {

        if (c[i]) {

            for (int j = i \* i; j <= m; j += i) {

                c[j] = false; // Flag khong phai snt

            }

        }

    }

}

int main() {

    ll n, s, sum = 0;

    scanf("%lld", &n);

    if ( n >= 15 ) {

        ll arr[n];

        inp(arr, n);

        out(arr, n);

        // Tao bang snt bang phan tu lon nhat

        ll m = arr[0];

        for (int i = 1; i < n; i++) {

            if (arr[i] > m) {

                m = arr[i];

            }

        }

        bool c[m + 1];

        snt(m, c);

        for (int i = 0; i < n; i++) {

            if (c[arr[i]]) {

                sum += arr[i];

            }

        }

        printf("\n%lld", sum);

    }

    return 0;

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 15  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 | OUTPUT: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15  41 |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Viết chương trình nhập mảng 1 chiều A có N phần tử có sử dụng hàm(với N nhập từ bàn phím).

a. Xuất các phần tử đã nhập ra màn hình

b. Xuất ra màn hình các chính phương của mảng và tính tổng các số đó?

## . Thuật toán

Nhập số nguyên n tương ứng số lượng phần tử mảng, nhập phần tử vào mảng sau đó in mảng đó ra màn hình

Gọi hàm **scp**, dùng For duyệt từng phần tử trong mảng kiểm tra xem phần tử đó có phải số chính phương không và in ra màn hình

Đồng thời cộng lần lượt vào sum để tính tổng của các số chính phương đó và in ra màn hình

## . Chương trình nguồn

#include <stdio.h>

#include <math.h>

#define ll long long

// Python

// for i in range (1,N+1):

//     if int(math.sqrt(i)) \* int(math.sqrt(i)) == i:

//         print(i,end=' ')

void inp(ll arr[], ll n ){

    for (int i = 0; i < n; i++) {

        scanf("%lld", &arr[i]);

    }

}

void out(ll arr[], ll n) {

    for (int i = 0; i < n; i++) {

        printf("%lld ", arr[i]);

    }

}

void scp(ll arr[], ll n) {

    ll sum = 0;

    for (int i = 0; i < n; i++) {

        if (sqrt(arr[i]) \* sqrt(arr[i]) == arr[i]) {

            printf("%lld ", arr[i]);

            sum += arr[i];

        }

    }

    printf("\n%lld", sum);

}

int main() {

    ll n;

    scanf("%lld", &n);

    ll arr[n];

    inp(arr, n);

    out(arr, n);

    printf("\n");

    scp(arr, n);

    return 0;

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 6  8 2 4 23 13 16 | OUTPUT: 8 2 4 23 13 16  4 16  20 |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Viết chương trình có sử dụng hàm

a. Viết chương trình nhập mảng một chiều A.Xuất mảng A ra màn hình

b. Xóa các phần âm trong mảng A.Xuất mảng A sau khi xóa các phần tử âm ra màn hình.

## . Thuật toán

Nhập số nguyên n tương ứng số lượng phần tử mảng, nhập phần tử vào mảng sau đó in mảng đó ra màn hình

Dùng For để duyệt các phần tử kiểm tra xem phần tử nào âm thì gọi hàm **del** để đẩy phần tử đó về cuối mảng và giảm số lượng trong mảng đi 1 để xóa phần tử đó

## . Chương trình nguồn

#include <stdio.h>

#define ll long long

void inp(ll arr[], ll n ){

    for (int i = 0; i < n; i++) {

        scanf("%lld", &arr[i]);

    }

}

void out(ll arr[], ll n) {

    for (int i = 0; i < n; i++) {

        printf("%lld ", arr[i]);

    }

}

void del(ll arr[], ll \*n, ll index) {

    // Dich phan tu ve cuoi mang

    for (int i = index; i < \*n - 1; i++) {

        arr[i] = arr[i + 1];

    }

    // Giam so luong phan tu trong mang

    (\*n)--;

}

int main() {

    ll n;

    scanf("%lld", &n);

    ll arr[n];

    inp(arr, n);

    out(arr, n);

    printf("\n");

    for (int i = 0; i < n;) {

        if (arr[i] < 0) {

            del(arr, &n, i);

        }

        else i++;

    }

    out(arr, n);

    return 0;

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 6  1 -5 -8 4 -3 6 | OUTPUT: 1 -5 -8 4 -3 6  1 4 6 |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Viết chương trình có sử dụng hàm

a.Nhập mảng số nguyên A gồm n phần tử và nhập số nguyên k.

b. Xóa phần tử có chỉ số k ra khỏi mảng A. Xuất mảng A sau khi xóa chỉ số chỉ số k ra màn hình.

## . Thuật toán

Nhập số nguyên n tương ứng số lượng phần tử mảng và số nguyên k

Dùng For để duyệt các phần tử kiểm tra xem phần tử nào = k thì gọi hàm **del** để đẩy phần tử đó về cuối mảng và giảm số lượng trong mảng đi 1 để xóa phần tử đó

## . Chương trình nguồn

#include <stdio.h>

#define ll long long

void inp(ll arr[], ll n ){

    for (int i = 0; i < n; i++) {

        scanf("%lld", &arr[i]);

    }

}

void out(ll arr[], ll n) {

    for (int i = 0; i < n; i++) {

        printf("%lld ", arr[i]);

    }

}

void del(ll arr[], ll \*n, ll index) {

    // Dich phan tu ve cuoi mang

    for (int i = index; i < \*n - 1; i++) {

        arr[i] = arr[i + 1];

    }

    // Giam so luong phan tu trong mang

    (\*n)--;

}

int main() {

    ll n, k;

    scanf("%lld%lld", &n, &k);

    ll arr[n];

    inp(arr, n);

    for (int i = 0; i < n;) {

        if (arr[i] == k) {

            del(arr, &n, i);

        }

else i++;

    }

    out(arr, n);

    return 0;

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 6 4  2 4 6 4 4 2 | OUTPUT: 2 6 2 |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Viết chương trình có sử dụng hàm

a. Viết chương trình nhập hai mảng một chiều A và B.Xuất mảng A và B ra màn hình .

b. Nối mảng A vào B (không dùng mảng phụ và không đươc sắp xếp trước và sau) đảm bảo thứ tự tăng dần. Xuất mảng sau khi nối ra màn hình.

## . Thuật toán

Nhập số nguyên x, y tương ứng số lượng phần tử mảng, tạo mảng A với số lượng [x + 100] để có thể chứa thêm mảng B khi cộng vào và mảng B với y phần tử

Nhập phần tử vào 2 mảng sau đó in 2 mảng đó ra màn hình

Sau đó dùng For để cộng mảng A và B lại với nhau, sau đó gọi hàm **sortArr** để sắp xếp lại phần tử tăng dần trong mảng A mới cộng

Dùng 2 For duyệt phần tử với For đầu tiên i = 0 đến n – 2 và For thứ hai từ j = i + 1 đến n - 1

Kiểm tra xem nếu A[ i ] > A[ j ] thì sẽ chuyển phần tử A[ i ] về sau và phần tử A[ j ] về trước cứ thế đến khi sắp xếp hết tất cả phần tử tăng dần và in ra mảng đã sắp xếp ra màn hình.

## . Chương trình nguồn

#include <stdio.h>

#define ll long long

void inp(ll arr[], ll n ){

    for (int i = 0; i < n; i++) {

        scanf("%lld", &arr[i]);

    }

}

void out(ll arr[], ll n) {

    for (int i = 0; i < n; i++) {

        printf("%lld ", arr[i]);

    }

}

ll sortArr(ll arr[], ll n) {

    for (int i = 0; i < n - 1; i++) {

        for (int j = i + 1; j < n; j++) {

            if (arr[i] > arr[j]) {

                ll t = arr[i];

                arr[i] = arr[j];

                arr[j] = t;

            }

        }

    }

}

int main() {

    ll x,y;

    scanf("%lld%lld", &x, &y);

    ll arrA[x + 100], arrB[y];

    inp(arrA, x); // Nhap mang A

    inp(arrB, y); // Nhap mang B

    out(arrA, x); // Xuat mang A

    printf("\n");

    out(arrB, y); // Xuat mang B

    // Mang A + B

    for (int i = 0; i < y; i++) {

        arrA[x + i] = arrB[i];

    }

    sortArr(arrA, x + y);

    printf("\n");

    out(arrA, x + y); // Xuat mang A + B

    return 0;

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 5 4  1 2 3 4 5  2 3 4 5 | OUTPUT: 1 2 3 4 5  2 3 4 5  1 2 2 3 3 4 4 5 5 |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Viết chương trình có sử dụng hàm

a. Đọc file chương trình \*.C có các dấu chú thích ( /\*….\*/ hoặc // )

b. Xóa các dấu chú thích để có chương trình không có dấu chú thích ?

## . Thuật toán

Mở file test.c để xóa kí tự bằng hàm **fopen** với chế độ đọc file, sau đó tạo 2 mảng inp và out để lưu trữ nội dung file và nội dung đã xử lí

Sau đó dùng hàm **fread** đọc duyệt qua các dòng trong file để kiểm tra xem kí tự comment, nếu gặp comment sẽ bỏ qua và chỉ thêm những chuỗi khác vào mảng out

Sau khi kiểm tra tất cả các dòng và ghi lại vào mảng thì dùng hàm **fopen** để mở file với chế độ ghi và dùng hàm **fwrite** ghi nội dung đã xử lí các kí tự comment vào file đó và dùng hàm **fclose** để đóng file lại

## . Chương trình nguồn

#include <stdio.h>

#include <string.h>

#define size 10000 // Kich thuoc toi da cua file

void rComment(const char \*input) {

    // Ham mo file de doc file

    // FILE \*fopen(const char \*filename, const char \*mode);

    FILE \*file = fopen(input, "r");

    char inp[size]; // Luu tru noi dung file

    char out[size]; // Luu tru noi dung da xu li

    int index = 0; // Vi tri mang

    // Ham doc noi dung file

    // Xu li va ghi vao output

    // size\_t fread(void \*ptr, size\_t size, size\_t count, FILE \*stream);

    size\_t read = fread(inp, 1, size, file);

    inp[read] = '\0'; // Ket thuc chuoi

    for (int i = 0; i < read; i++) {

        if (inp[i] == '/' && inp[i + 1] == '/') {

            i++; // Bo qua

        }

        else if (inp[i] == '/' && inp[i + 1] == '\*') {

            i++; // Bo qua

        }

        else if (inp[i] == '\*' && inp[i + 1] == '/') {

            i++; // Bo qua

        }

        else {

            // Neu khong phai //, /\*, \*/ thi them vao inp

            out[index++] = inp[i];

        }

    }

    out[index] = '\0';

    // Ham mo file de ghi noi dung vao file

    file = fopen(input, "w");

    if (file == NULL) {

        return;

    }

    // Ham ghi noi dung moi vao file

    // size\_t fwrite(const void \*ptr, size\_t size, size\_t count, FILE \*stream);

    fwrite(out, 1, index, file);

    // Ham dong file

    // int fclose(FILE \*stream);

    fclose(file);

}

int main() {

    const char \*input = "test.c";

    rComment(input);

    return 0;

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: | OUTPUT: |

## . Nhận xét nếu có

|  |
| --- |
|  |

## . Tên đề bài

Tính toán số lớn. Viết chương trình xử lý tính toán các số lớn. Yêu cầu chỉ thực hiện hai phép toán cộng và trừ, sử dụng chuỗi ký tự để biểu diễn số lớn.

## . Thuật toán

Nhập 2 số nguyên lớn a, b và chuyển số nguyên thành chuỗi ký tự để thực hiện các phép tính toán số lớn

Tạo hàm **chen** thêm các ký tự '0' vào đầu chuỗi a để độ dài của nó bằng với độ dài của chuỗi b để có thể thực hiện cộng hoặc trừ chuỗi.

Hàm **tong** thực hiện phép cộng hai số nguyên lớn (dưới dạng chuỗi) a và b, lưu kết quả vào c:

* Dùng variable du để lưu giá trị dư khi cộng.
* Dùng For lặp từ cuối chuỗi đến đầu chuỗi:
* Cộng các chữ số từ a và b và cộng thêm du.
* Lấy kết quả dư (num % 10) để thêm vào mảng c.
* Cập nhật kết quả du bằng kết quả nguyên (num / 10).
* Nếu sau khi lặp xong vẫn còn dư (du > 0), chèn du vào đầu c.

Hàm **hieu** thực hiện phép trừ hai số nguyên lớn (dưới dạng chuỗi) a và b, lưu kết quả vào c:

* Dùng variable du để lưu giá trị mượn.
* Dùng For lặp từ cuối chuỗi đến đầu:
* Nếu chữ số của a nhỏ hơn (bi + du), thực hiện mượn.
* Ghi kết quả vào c sau khi trừ.
* Dùng hàm từ thư viện <string.h> để xóa các chữ số 0 dư thừa ở đầu chuỗi kết quả (**memmove**).
* Nếu a >= b, gọi hàm hieu(a, b, c) và in ra kết quả phép trừ. Nếu a < b, gọi hieu(b, a, c) và thêm dấu '-' vào kết quả để thể hiện số âm

## . Chương trình nguồn

#include <stdio.h>

#include <string.h>

void chen(char a[], char b[]) {

    int i;

    int lena = strlen(a), lenb = strlen(b);

    while (lena < lenb) {

        for (i = lena - 1; i >= 0; i--)

            a[i + 1] = a[i];

        lena++;

        a[0] = '0';

    }

    a[lena] = '\0';

}

void tong(char a[], char b[], char c[]) {

    int i;

    int du = 0;

    int n = strlen(a);

    for (i = n - 1; i >= 0; i--) {

        int num = (a[i] - '0') + (b[i] - '0') + du;

        c[i] = num % 10 + '0';

        du = num / 10;

    }

    if (du > 0) {

        for (i = n - 1; i >= 0; i--)

            c[i + 1] = c[i];

        c[0] = du + '0';

        c[n + 1] = '\0';

    }

else {

        c[n] = '\0';

    }

}

void hieu(char a[], char b[], char c[]) {

    int du = 0;

    int n = strlen(a);

    int i;

    for (i = n - 1; i >= 0; i--) {

        int ai = a[i] - '0';

        int bi = b[i] - '0';

        int num;

        if (ai < (bi + du)) {

            num = ai + 10 - bi - du;

            du = 1;

        }

else {

            num = ai - bi - du;

            du = 0;

        }

        c[i] = num + '0';

    }

    c[n] = '\0';

    for (i = 0; c[i] == '0' && c[i + 1] != '\0'; i++);

    if (i > 0) {

        memmove(c, c + i, strlen(c) - i + 1);

    }

}

int main() {

    char a[10000], b[10000], c[10000];

    int i, len;

    scanf("%s", a);

    scanf("%s", b);

    int lena = strlen(a), lenb = strlen(b);

    if (lena < lenb) chen(a, b);

    else chen(b, a);

    tong(a, b, c);

    printf("%s", c);

    c[0] = '\0';

    if (strcmp(a, b) >= 0) {

        hieu(a, b, c);

    }

else {

        hieu(b, a, c);

        len = strlen(c);

        for (i = len; i > 0; i--)

            c[i] = c[i - 1];

        c[0] = '-';

        c[len + 1] = '\0';

    }

    printf("\n%s", c);

    return 0;

}

## . Kết quả chương trinh

|  |  |
| --- | --- |
| INPUT: 987654321987  123456789987 | OUTPUT: 1111111111974  864197532000 |

## . Nhận xét nếu có

|  |
| --- |
|  |

TÀI LIỆU THAM KHẢO

[1] Tên tác giả, *Tên tài liệu*, Tên nhà xuất bản, năm xuất bản

[2] Tên chủ sở hữu, *Tên bài viết*, url, ngày truy cập

Github: [practical-tutorials/project-based-learning: Curated list of project-based tutorials](https://github.com/practical-tutorials/project-based-learning?tab=readme-ov-file#cc)

VNOJ – NTH: [hoaitrinhnth - VNOJ: VNOI Online Judge](https://oj.vnoi.info/organization/nt-188)

Leetcode: [Problems - LeetCode](https://leetcode.com/problemset/)

Stack Overflow: [Newest Questions - Stack Overflow](https://stackoverflow.com/questions)

W3School: [C Tutorial](https://www.w3schools.com/c/index.php)

Tiktok – Code4Life: [CODE4LIFE (@code4life.official) | TikTok](https://www.tiktok.com/@code4life.official)