A. BÀI TẬP MINH HỌA

Bài 1. Cho hai số nguyên a và b. Viết chương trình hoán đổi hai số (dùng con trỏ).

Moi ban nhap so a: 10

Moi ban nhap so b: 20

Truoc khi hoan vi, a = 10 va b = 20.

Sau khi hoan vi, a = 20 va b = 10.



#include <stdio.h>

#include <stdlib.h>

/\*

Hoan vi hai so nguyen duong

+ Vao: &a, &b

+ Ra : \*a, \*b

+ Vi du: a[][MAXC] = {{1,2,3},{4,5,6}}, nd = 2, nc=3

\*/

void HoanVi(int \*a, int \*b)

{

    int tmp = (\*a);

    (\*a) = (\*b);

    (\*b) = tmp;

}

int main()

{

    /\* Khai bao bien \*/

    int x, y;

    /\* Nhap du lieu \*/

    printf("Moi ban nhap so a: ");

    scanf("%d", &x);

    printf("Moi ban nhap so b: ");

    scanf("%d", &y);

    /\* Xu ly\*/

    printf("Truoc khi hoan vi, a = %d va b = %d.\n", x, y);

    HoanVi(&x, &y);

    printf("Sau khi hoan vi, a = %d va b = %d.\n", x, y);

    /\* Dung man hinh xem ket qua \*/

    system("PAUSE");

    return 0;

} // main

Bài 2. Cho số nguyên dương n. Hãy kiểm tra n có phải là số chính phương (dùng con trỏ). Biết số

chính phương là bình phương của một số.

Moi ban nhap so nguyen n: 9

So 9 la so chinh phuong.



#include <stdio.h>

#include <math.h>

// Hàm kiểm tra số chính phương

int isPerfectSquare(int \*n)

{

    int sqrtN = (int)sqrt(\*n);    // Tính căn bậc hai và ép kiểu về số nguyên

    return (sqrtN \* sqrtN == \*n); // Kiểm tra bình phương lại có bằng n không

}

int main()

{

    int n;

    printf("Nhập số nguyên dương n: ");

    scanf("%d", &n);

    // Kiểm tra nếu n <= 0

    if (n <= 0)

    {

        printf("Vui lòng nhập số nguyên dương!\n");

        return 1; // Thoát chương trình với mã lỗi

    }

    // Kiểm tra số chính phương

    if (isPerfectSquare(&n))

    {

        printf("%d là số chính phương.\n", n);

    }

    else

    {

        printf("%d không phải là số chính phương.\n", n);

    }

    return 0;

}

Bài 3. Cho dãy số nguyên a có n phần tử (n ≤ 100). Hãy tạo dãy b chứa các số chẳn của a và xuất b ra

màn hình.

Moi ban nhap so luong phan tu: 3

Phan tu 0: 2

Phan tu 1: 3

Phan tu 2: 6

+ Day chua cac so chan

Day so co 2 phan tu: 2 6



#include <stdio.h>

#include <stdlib.h>

#define MAX 100 // so luong phan tu toi da cua day

/\*

Nhap day so nguyen tu ban phim

+ Vao: ban phim

+ Ra : (\*a), n

+ Vi du: (\*a) = {1,2,3}, n = 3

\*/

void NhapMang(int \*\*a, int \*n)

{

    printf("Moi ban nhap so luong phan tu: "); // nhap (\*n)

    scanf("%d", n);

    while ((\*n) <= 0 || (\*n) > MAX) // kiem tra nhap (\*n)

    {                               // nhap lai (\*n)

        printf("Nhap sai! Nhap lai: ");

        scanf("%d", n);

    } // while n

    (\*a) = new int[(\*n)];               // cap phat mang (\*a) co (\*n) phan tu

    for (int i = 0; i <= (\*n) - 1; i++) // duyet tu vi tri 0 den (\*n)-1

    {                                   // nhap \*((\*a)+i) <-> (\*a)[i]

        printf("Phan tu %d: ", i);

        scanf("%d", (\*a) + i); // scanf("%d",&(\*a)[i]);

    } // for i

} // NhapMang

/\*

Xuat day so nguyen ra man hinh

+ Vao: (\*a), n

+ Ra : man hinh

+ Vi du:

\*a = {2,5,3}, n = 3

=> man hinh = 2 5 3

\*/

void XuatMang(int \*a, int n)

{

    printf("Day so co %d phan tu: ", n); // xuat so luong phan tu

    for (int i = 0; i <= n - 1; i++)     // duyet tu vi tri 0 den n-1 lam

    {

        printf("%d ", \*(a + i)); // xuat a[i] hay printf("%d",a[i]);

    } // for i

    printf("\n"); // xuong dong

} // XuatMang

/\*

Dem so chan co tron day

+ Vao: \*a, n

+ Ra : dem

\*/

int DemChan(int \*a, int n)

{

    int dem = 0;

    for (int i = 0; i <= n - 1; i++)

    {

        if (\*(a + i) % 2 == 0)

            dem++;

    }

    return dem;

} // DemChan

/\*

Tach cac so chan

+ Vao: \*a, n

+ Ra : \*b, m

\*/

void TachChan(int \*a, int n, int \*\*b, int \*m)

{

    (\*m) = DemChan(a, n); // dem so luong phan tu day (\*b)

    (\*b) = new int[(\*m)]; // cap phat day so (\*b) co (\*m) phan tu

    int cs = 0;

    for (int i = 0; i <= n - 1; i++)

    {

        if (\*(a + i) % 2 == 0) // a[i]%2

        {

            \*((\*b) + cs) = \*(a + i); // (\*b)[cs] = a[i]

            cs++;

        }

    } // for

} // TachChan

int main()

{

    /\* Khai bao bien \*/

    int \*b = NULL, k = 0;

    int \*aChan = NULL, nChan;

    /\* Nhap du lieu \*/

    NhapMang(&b, &k);

    /\* Xu ly\*/

    TachChan(b, k, &aChan, &nChan);

    /\* Xuat du lieu \*/

    printf("+ Day chua cac so chan\n");

    XuatMang(aChan, nChan);

    /\* QUAN TRONG - Huy vung nho\*/

    delete[] b;

    delete[] aChan;

    /\* Dung man hinh xem ket qua \*/

    system("PAUSE");

    return 0;

} // main

Bài 4. Cho dãy số thực a có n phần tử (n ≤ 100). Hãy kiểm tra các tính chất sau của dãy a:

 Tính đan dấu (các phần tử có dấu xen kẽ nhau),

 Tính đơn điệu (dãy số tăng hoặc giảm),

 Tính chất đối xứng.



#include <stdio.h>

#include <stdbool.h>

#define MAX 100

// Hàm kiểm tra tính đan dấu

bool isAlternatingSign(double \*a, int n)

{

    for (int i = 0; i < n - 1; i++)

    {

        if (a[i] \* a[i + 1] >= 0) // Hai số cùng dấu

            return false;

    }

    return true;

}

// Hàm kiểm tra tính đơn điệu (tăng hoặc giảm)

bool isMonotonic(double \*a, int n)

{

    bool increasing = true, decreasing = true;

    for (int i = 0; i < n - 1; i++)

    {

        if (a[i] > a[i + 1])

            increasing = false;

        if (a[i] < a[i + 1])

            decreasing = false;

    }

    return increasing || decreasing;

}

// Hàm kiểm tra tính đối xứng

bool isSymmetric(double \*a, int n)

{

    for (int i = 0; i < n / 2; i++)

    {

        if (a[i] != a[n - 1 - i])

            return false;

    }

    return true;

}

int main()

{

    int n;

    double a[MAX];

    // Nhập số phần tử

    printf("Nhập số phần tử của dãy (n <= 100): ");

    scanf("%d", &n);

    // Kiểm tra giới hạn n

    if (n <= 0 || n > MAX)

    {

        printf("Số phần tử không hợp lệ!\n");

        return 1;

    }

    // Nhập mảng

    printf("Nhập dãy số:\n");

    for (int i = 0; i < n; i++)

    {

        printf("a[%d] = ", i);

        scanf("%lf", &a[i]);

    }

    // Kiểm tra các tính chất

    printf("\nKiểm tra các tính chất của dãy số:\n");

    if (isAlternatingSign(a, n))

        printf("- Dãy có tính đan dấu.\n");

    else

        printf("- Dãy không có tính đan dấu.\n");

    if (isMonotonic(a, n))

        printf("- Dãy có tính đơn điệu.\n");

    else

        printf("- Dãy không có tính đơn điệu.\n");

    if (isSymmetric(a, n))

        printf("- Dãy có tính đối xứng.\n");

    else

        printf("- Dãy không có tính đối xứng.\n");

    return 0;

}

Bài 5. Cho dãy số nguyên a có n phần tử (n ≤ 100). Hãy tạo dãy b chứa các số nguyên tố của a và xuất

b ra màn hình.

Moi ban nhap so luong phan tu: 3

Phan tu 0: 2

Phan tu 1: 3

Phan tu 2: 6

+ Mang chua cac so nguyen to

Day so co 2 phan tu: 2 3



#include <stdio.h>

#include <stdbool.h>

#include <math.h>

#define MAX 100

// Hàm kiểm tra số nguyên tố

bool isPrime(int num)

{

    if (num < 2)

        return false;

    if (num == 2 || num == 3)

        return true;

    if (num % 2 == 0 || num % 3 == 0)

        return false;

    for (int i = 5; i \* i <= num; i += 6)

    {

        if (num % i == 0 || num % (i + 2) == 0)

            return false;

    }

    return true;

}

int main()

{

    int a[MAX], b[MAX], n, m = 0;

    int \*pa = a, \*pb = b; // Con trỏ trỏ đến mảng a và b

    // Nhập số phần tử

    printf("Nhập số phần tử của dãy (n ≤ 100): ");

    scanf("%d", &n);

    if (n <= 0 || n > MAX)

    {

        printf("Số phần tử không hợp lệ!\n");

        return 1;

    }

    // Nhập dãy số a

    printf("Nhập dãy số:\n");

    for (int i = 0; i < n; i++)

    {

        printf("a[%d] = ", i);

        scanf("%d", (pa + i)); // Dùng con trỏ nhập dữ liệu

    }

    // Tạo dãy b chứa số nguyên tố

    for (int i = 0; i < n; i++)

    {

        if (isPrime(\*(pa + i)))

        {                          // Kiểm tra số nguyên tố

            \*(pb + m) = \*(pa + i); // Gán vào mảng b

            m++;

        }

    }

    // Xuất dãy b

    printf("\nDãy số nguyên tố:\n");

    if (m == 0)

    {

        printf("Không có số nguyên tố trong dãy!\n");

    }

    else

    {

        for (int i = 0; i < m; i++)

        {

            printf("%d ", \*(pb + i)); // Xuất dãy b bằng con trỏ

        }

        printf("\n");

    }

    return 0;

}

B. BÀI TẬP PHẢI LÀM

Con trỏ cho đối tượng

Bài 1. Cho hệ số a, b, c của phương trình bậc hai. Viết chương trình giải phương trình bậc hai (dùng

con trỏ).

Moi ban nhap he so a, b, c: 1 2 1

Phuong trinh co 1 nghiem kep x = -1.



#include <stdio.h>

#include <math.h>

// Hàm giải phương trình bậc hai, sử dụng con trỏ để trả kết quả

void giaiPTBacHai(double \*a, double \*b, double \*c, double \*x1, double \*x2, int \*soNghiem)

{

    // Trường hợp đặc biệt: a = 0 (phương trình bậc nhất hoặc vô nghiệm)

    if (\*a == 0)

    {

        if (\*b == 0)

        {

            \*soNghiem = (\*c == 0) ? -1 : 0; // -1: Vô số nghiệm, 0: Vô nghiệm

        }

        else

        {

            \*soNghiem = 1;

            \*x1 = -(\*c) / (\*b);

        }

        return;

    }

    // Tính biệt số Delta

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

    if (delta < 0)

    {

        \*soNghiem = 0; // Vô nghiệm

    }

    else if (delta == 0)

    {

        \*soNghiem = 1; // Nghiệm kép

        \*x1 = -(\*b) / (2 \* (\*a));

    }

    else

    {

        \*soNghiem = 2; // Hai nghiệm phân biệt

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

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

    }

}

int main()

{

    double a, b, c, x1, x2;

    int soNghiem;

    // Nhập hệ số a, b, c

    printf("Nhập hệ số a, b, c: ");

    scanf("%lf %lf %lf", &a, &b, &c);

    // Gọi hàm giải phương trình

    giaiPTBacHai(&a, &b, &c, &x1, &x2, &soNghiem);

    // Xuất kết quả

    if (soNghiem == -1)

    {

        printf("Phương trình có vô số nghiệm.\n");

    }

    else if (soNghiem == 0)

    {

        printf("Phương trình vô nghiệm.\n");

    }

    else if (soNghiem == 1)

    {

        printf("Phương trình có nghiệm duy nhất: x = %.2lf\n", x1);

    }

    else

    {

        printf("Phương trình có hai nghiệm phân biệt:\n");

        printf("x1 = %.2lf\n", x1);

        printf("x2 = %.2lf\n", x2);

    }

    return 0;

}

Con trỏ cho mảng một chiều

Bài 2. Cho dãy số nguyên a có n phần tử (n ≤ 100). Hãy kiểm tra các tính chất sau của dãy a:

 Tính chẵn lẻ (các phần tử xen kẽ nhau chẵn lẻ),

 Tính toàn chẵn (dãy số chứa toàn số chẵn),

Moi ban nhap so luong phan tu: 5

Phan tu 0: 2

Phan tu 1: 5

Phan tu 2: 6

Phan tu 3: 5

Phan tu 4: 2

+ Day co tinh chat chan le

+ Day khong co tinh chat toan chan



#include <stdio.h>

// Hàm kiểm tra dãy có xen kẽ chẵn lẻ không (dùng con trỏ)

int kiemTraXenKeChanLe(int \*arr, int n)

{

    for (int i = 1; i < n; i++)

    {

        if ((\*(arr + i) % 2) == (\*(arr + i - 1) % 2))

        {

            return 0; // Nếu hai phần tử liên tiếp cùng chẵn hoặc cùng lẻ => Không thỏa mãn

        }

    }

    return 1; // Dãy xen kẽ chẵn lẻ

}

// Hàm kiểm tra dãy có toàn số chẵn không (dùng con trỏ)

int kiemTraToanChan(int \*arr, int n)

{

    for (int i = 0; i < n; i++)

    {

        if (\*(arr + i) % 2 != 0)

        {

            return 0; // Nếu có ít nhất một số lẻ => Không phải dãy toàn chẵn

        }

    }

    return 1; // Dãy toàn chẵn

}

int main()

{

    int n;

    // Nhập số lượng phần tử

    printf("Moi ban nhap so luong phan tu: ");

    scanf("%d", &n);

    // Kiểm tra điều kiện hợp lệ

    if (n <= 0 || n > 100)

    {

        printf("So luong phan tu khong hop le!\n");

        return 1;

    }

    int arr[n];

    // Nhập dãy số

    for (int i = 0; i < n; i++)

    {

        printf("Phan tu %d: ", i);

        scanf("%d", arr + i); // Nhập giá trị vào mảng bằng con trỏ

    }

    // Kiểm tra tính chẵn lẻ xen kẽ

    if (kiemTraXenKeChanLe(arr, n))

    {

        printf("+ Day co tinh chat chan le\n");

    }

    else

    {

        printf("+ Day khong co tinh chat chan le\n");

    }

    // Kiểm tra tính toàn chẵn

    if (kiemTraToanChan(arr, n))

    {

        printf("+ Day co tinh chat toan chan\n");

    }

    else

    {

        printf("+ Day khong co tinh chat toan chan\n");

    }

    return 0;

}

Bài 3. Cho dãy ký tự a có n phần tử (n ≤ 100). Hãy tạo dãy b chứa các ký tự nguyên âm của a và xuất b

ra màn hình.

Moi ban nhap so luong phan tu: 5

Phan tu 0: a

Phan tu 1: b

Phan tu 2: c

Phan tu 3: d

Phan tu 4: e

+ Day co tinh chat chan le

+ Day khong co tinh chat toan chan



#include <stdio.h>

#include <ctype.h> // Thư viện xử lý ký tự

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

int laNguyenAm(char c)

{

    c = tolower(c); // Chuyển về chữ thường để kiểm tra

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

}

// Hàm lọc nguyên âm vào mảng b

void locNguyenAm(char \*a, int n, char \*b, int \*m)

{

    \*m = 0; // Số phần tử của mảng b ban đầu là 0

    for (int i = 0; i < n; i++)

    {

        if (laNguyenAm(\*(a + i)))

        {

            \*(b + (\*m)) = \*(a + i); // Thêm nguyên âm vào mảng b

            (\*m)++;                 // Tăng số lượng phần tử của b

        }

    }

}

int main()

{

    int n;

    // Nhập số lượng phần tử

    printf("Moi ban nhap so luong phan tu: ");

    scanf("%d", &n);

    getchar(); // Loại bỏ ký tự xuống dòng sau khi nhập số

    // Kiểm tra điều kiện hợp lệ

    if (n <= 0 || n > 100)

    {

        printf("So luong phan tu khong hop le!\n");

        return 1;

    }

    char a[n], b[n]; // Mảng gốc và mảng chứa nguyên âm

    int m;           // Số lượng phần tử trong mảng b

    // Nhập dãy ký tự

    for (int i = 0; i < n; i++)

    {

        printf("Phan tu %d: ", i);

        scanf("%c", a + i); // Nhập ký tự vào mảng bằng con trỏ

        getchar();          // Xử lý ký tự xuống dòng

    }

    // Lọc nguyên âm vào mảng b

    locNguyenAm(a, n, b, &m);

    // Xuất mảng b (các nguyên âm)

    printf("Cac ky tu nguyen am trong day: ");

    if (m == 0)

    {

        printf("Khong co ky tu nguyen am.");

    }

    else

    {

        for (int i = 0; i < m; i++)

        {

            printf("%c ", \*(b + i));

        }

    }

    printf("\n");

    return 0;

}

Bài 4. Cho dãy số thực a có n phần tử và b có m phần tử (n,m ≤ 100) tăng dần. Hãy tạo dãy c từ hai

dãy a, b sao cho tăng dần (không dùng sắp xếp).

+ Day so a

Moi ban nhap so luong phan tu: 3

Phan tu 0: 1.1

Phan tu 1: 3.2

Phan tu 2: 5.3

+ Day so b

Moi ban nhap so luong phan tu: 4

Phan tu 0: 2.1

Phan tu 1: 3.2

Phan tu 2: 5.4

Phan tu 3: 8.4

+ Day so c

Day so co 7 phan tu: 1.1 2.1 3.2 3.2 5.4 5.4 8.4



#include <stdio.h>

// Hàm trộn hai mảng tăng dần thành một mảng tăng dần

void tronMang(float \*a, int n, float \*b, int m, float \*c, int \*p)

{

    int i = 0, j = 0, k = 0;

    while (i < n && j < m)

    {

        if (\*(a + i) < \*(b + j))

        {

            \*(c + k) = \*(a + i);

            i++;

        }

        else

        {

            \*(c + k) = \*(b + j);

            j++;

        }

        k++;

    }

    // Sao chép các phần tử còn lại của mảng a

    while (i < n)

    {

        \*(c + k) = \*(a + i);

        i++;

        k++;

    }

    // Sao chép các phần tử còn lại của mảng b

    while (j < m)

    {

        \*(c + k) = \*(b + j);

        j++;

        k++;

    }

    \*p = k; // Cập nhật số lượng phần tử của mảng c

}

int main()

{

    int n, m, p;

    // Nhập số lượng phần tử của mảng a

    printf("+ Day so a\nMoi ban nhap so luong phan tu: ");

    scanf("%d", &n);

    // Kiểm tra điều kiện hợp lệ

    if (n <= 0 || n > 100)

    {

        printf("So luong phan tu khong hop le!\n");

        return 1;

    }

    float a[n];

    for (int i = 0; i < n; i++)

    {

        printf("Phan tu %d: ", i);

        scanf("%f", a + i); // Nhập trực tiếp vào con trỏ

    }

    // Nhập số lượng phần tử của mảng b

    printf("+ Day so b\nMoi ban nhap so luong phan tu: ");

    scanf("%d", &m);

    if (m <= 0 || m > 100)

    {

        printf("So luong phan tu khong hop le!\n");

        return 1;

    }

    float b[m];

    for (int i = 0; i < m; i++)

    {

        printf("Phan tu %d: ", i);

        scanf("%f", b + i);

    }

    // Mảng c có tối đa n + m phần tử

    float c[n + m];

    // Gộp hai mảng vào mảng c

    tronMang(a, n, b, m, c, &p);

    // Xuất mảng c

    printf("+ Day so c\nDay so co %d phan tu: ", p);

    for (int i = 0; i < p; i++)

    {

        printf("%.1f ", \*(c + i));

    }

    printf("\n");

    return 0;

}

Bài 5. Cho dãy số nguyên a có n phần tử (n ≤ 100). Hãy tách dãy a thành dãy b chứa số chẵn và dãy c

chứa số lẻ..

+ Day so a

Moi ban nhap so luong phan tu: 3

Phan tu 0: 2

Phan tu 1: 5

Phan tu 2: 8

+ Day so b chua so chan

Day so co 2 phan tu: 2 8

+ Day so c

Day so co 1 phan tu: 5



#include <stdio.h>

// Hàm tách mảng a thành mảng b (số chẵn) và mảng c (số lẻ)

void tachMang(int \*a, int n, int \*b, int \*m, int \*c, int \*p)

{

    \*m = 0; // Đếm số phần tử của mảng b (chẵn)

    \*p = 0; // Đếm số phần tử của mảng c (lẻ)

    for (int i = 0; i < n; i++)

    {

        if (\*(a + i) % 2 == 0)

        { // Nếu số chẵn

            \*(b + \*m) = \*(a + i);

            (\*m)++;

        }

        else

        { // Nếu số lẻ

            \*(c + \*p) = \*(a + i);

            (\*p)++;

        }

    }

}

int main()

{

    int n, m, p;

    // Nhập số lượng phần tử của mảng a

    printf("+ Day so a\nMoi ban nhap so luong phan tu: ");

    scanf("%d", &n);

    // Kiểm tra điều kiện hợp lệ

    if (n <= 0 || n > 100)

    {

        printf("So luong phan tu khong hop le!\n");

        return 1;

    }

    int a[n], b[n], c[n]; // Mảng b và c có thể chứa tối đa n phần tử

    // Nhập mảng a

    for (int i = 0; i < n; i++)

    {

        printf("Phan tu %d: ", i);

        scanf("%d", a + i);

    }

    // Gọi hàm tách mảng

    tachMang(a, n, b, &m, c, &p);

    // Xuất mảng b (chứa số chẵn)

    printf("+ Day so b chua so chan\n");

    printf("Day so co %d phan tu: ", m);

    for (int i = 0; i < m; i++)

    {

        printf("%d ", \*(b + i));

    }

    printf("\n");

    // Xuất mảng c (chứa số lẻ)

    printf("+ Day so c chua so le\n");

    printf("Day so co %d phan tu: ", p);

    for (int i = 0; i < p; i++)

    {

        printf("%d ", \*(c + i));

    }

    printf("\n");

    return 0;

}