

```

/*
Thông tin cá nhân
*/
#include<iostream>
using namespace std;

// NGUYÊN MẪU HÀM
void nhap_mang(int[], int);
void xuat_mang(int[], int);
int tong_mang(int[], int);
int tinh_tong_pt_chan(int[], int);
int tinh_tong_pt_vt_le(int[], int);
int tinh_tong_pt_duong(int[], int);
int tinh_tong_pt_am(int[], int);
int la_so_nguyen_to(int);
int tinh_tong_pt_nguyen_to(int[], int);
int la_so_hoan_thien(int);
int tinh_tong_pt_hoan_thien(int[], int);
float tinh_trung_binh_cong_mang(int[], int);
int tim_x(int[], int, int);
int tim_gt_lon_nhat(int[], int);
int dem_x(int[], int, int);
int tim_vt_x_cuoi_cung(int[], int, int);
int tim_gt_nho_nhat(int[], int);
int tim_gt_duong_nho_nhat(int[], int);
int tim_gt_am_lon_nhat(int[], int);
void hoan_vi(int&, int&);
void sap_xep_tang_dan(int[], int);
void sap_xep_giam_dan(int[], int);
void sap_xep_chan_tang_dan(int[], int);
void sap_xep_le_giam_dan(int[], int);
void them_cuoi(int[], int&, int);
void them_dau(int[], int&, int);
void them_vt_k(int[], int&, int, int);
void xoa_cuoi(int[], int&);
void xoa_dau(int[], int&);
void xoa_vt_k(int[], int&, int);

// HÀM MAIN
void main()
{
    int arr[100];
    int sl;
    // Nhập số lượng phần tử mảng
    do
    {
        cout << "Nhap so luong phan tu mang: ";
        cin >> sl;
    } while (sl < 1 || sl > 100);

    // Nhập mảng (Nhập các giá trị vào mảng)
    nhap_mang(arr, sl);

    // In mảng (in các giá trị của mảng)
    xuat_mang(arr, sl);

    // Tính tổng mảng
    int tong = tong_mang(arr, sl);
    cout << "Tong cac phan tu: " << tong << endl;
}

```

```

        sap_xep_le_giam_dan(arr, sl);
        xuat_mang(arr, sl);

        system("pause");
    }

// ĐỊNH NGHĨA HÀM
void nhap_mang(int a[], int n)
{
    for (int i = 0; i < n; i++)
    {
        cout << "Nhập a[" << i << "] = ";
        cin >> a[i];
    }
}

void xuat_mang(int a[], int n)
{
    for (int i = 0; i < n; i++)
    {
        cout << "a[" << i << "] = " << a[i] << endl;
    }
}

int tong_mang(int a[], int n)
{
    int s = 0;
    for (int i = 0; i < n; i++)
    {
        s += a[i];
    }

    return s;
}

int tinh_tong_pt_chan(int a[], int n)
{
    int s = 0;
    for (int i = 0; i < n; i++)
    {
        if (a[i] % 2 == 0)
        {
            s += a[i];
        }
    }

    return s;
}

```

```
int tinh_tong_pt_vt_le(int a[], int n)
{
    int s = 0;
    for (int i = 1; i < n; i+=2)
    {
        s += a[i];
    }

    return s;
}

int tinh_tong_pt_duong(int a[], int n)
{
    int s = 0;
    for (int i = 0; i < n; i++)
    {
        if (a[i] > 0)
        {
            s += a[i];
        }
    }

    return s;
}

int tinh_tong_pt_am(int a[], int n)
{
    int s = 0;
    for (int i = 0; i < n; i++)
    {
        if (a[i] < 0)
        {
            s += a[i];
        }
    }

    return s;
}

int la_so_nguyen_to(int n)
{
    int dem = 0;
    for (int i = 1; i <= n; i++)
    {
        if (n%i == 0)
        {
            dem++;
        }
    }

    if (dem == 2)
        return 1;
    return 0;
}
```

```

int tinh_tong_pt_nguyen_to(int a[], int n)
{
    int s = 0;
    for (int i = 0; i < n; i++)
    {
        if (la_so_nguyen_to(a[i]) == 1)
        {
            s += a[i];
        }
    }

    return s;
}

int la_so_hoan_thien(int n)
{
    int tong = 0;
    for (int i = 1; i < n; i++)
    {
        if (n%i == 0)
        {
            tong += i;
        }
    }

    if (tong == n)
        return 1;
    return 0;
}

int tinh_tong_pt_hoan_thien(int a[], int n)
{
    int s = 0;
    for (int i = 0; i < n; i++)
    {
        if (la_so_hoan_thien(a[i]) == 1)
        {
            s += a[i];
        }
    }

    return s;
}

float tinh_trung_binh_cong_mang(int a[], int n)
{
    return (float)tong_mang(a, n) / n;
}

```

```

int tim_x(int a[], int n, int x)
{
    for (int i = 0; i < n; i++)
    {
        if (a[i] == x)
        {
            return i;
        }
    }
    return -1;
}

int tim_gt_lon_nhat(int a[], int n)
{
    int max = a[0];
    for (int i = 1; i < n; i++)
    {
        if (a[i] > max)
        {
            max = a[i];
        }
    }

    return max;
}

int dem_x(int a[], int n, int x)
{
    int dem = 0;
    for (int i = 0; i < n; i++)
    {
        if (a[i] == x)
        {
            dem++;
        }
    }
    return dem;
}

int tim_vt_x_cuoi_cung(int a[], int n, int x)
{
    for (int i = n - 1; i >= 0; i--)
    {
        if (a[i] == x)
        {
            return i;
        }
    }
    return -1;
}

```

```

int tim_gt_nho_nhat(int a[], int n)
{
    int min = a[0];
    for (int i = 1; i < n; i++)
    {
        if (a[i] < min)
        {
            min = a[i];
        }
    }

    return min;
}

int tim_gt_duong_nho_nhat(int a[], int n)
{
    int min = 0;
    for (int i = 0; i < n; i++)
    {
        if (a[i] > 0 && (min == 0 || a[i] < min))
        {
            min = a[i];
        }
    }

    return min; // min = 0 ==> không tìm được giá trị dương nhỏ nhất
}

int tim_gt_am_lon_nhat(int a[], int n)
{
    int max = 0;
    for (int i = 0; i < n; i++)
    {
        if (a[i] < 0 && (max == 0 || a[i] > max))
        {
            max = a[i];
        }
    }

    return max; // min = 0 ==> không tìm được giá trị âm lớn nhất
}

void hoan_vi(int& a, int& b)
{
    int tam = a;
    a = b;
    b = tam;
}

```

```

void sap_xep_tang_dan(int a[], int n)
{
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = i + 1; j < n; j++)
        {
            if (a[i] > a[j])
            {
                hoan_vi(a[i], a[j]);
            }
        }
    }
}

void sap_xep_giam_dan(int a[], int n)
{
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = i + 1; j < n; j++)
        {
            if (a[i] < a[j])
            {
                hoan_vi(a[i], a[j]);
            }
        }
    }
}

void sap_xep_chan_tang_dan(int a[], int n)
{
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = i + 1; j < n; j++)
        {
            if (a[i] % 2 == 0 && a[j] % 2 == 0 && a[i] > a[j])
            {
                hoan_vi(a[i], a[j]);
            }
        }
    }
}

void sap_xep_le_giam_dan(int a[], int n)
{
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = i + 1; j < n; j++)
        {
            if (a[i] % 2 != 0 && a[j] % 2 != 0 && a[i] < a[j])
            {
                hoan_vi(a[i], a[j]);
            }
        }
    }
}

```

```

void them_cuoi(int a[], int& n, int x)
{
    // Kiểm tra còn thêm vào mảng được không?
    if (n < 100)
    {
        // Thêm vào cuối
        a[n] = x;
        n++;
    }
    else
    {
        cout << "Mang da day! Khong them them vao!\n";
    }
}

void them_dau(int a[], int& n, int x)
{
    // Kiểm tra còn thêm vào mảng được không?
    if (n < 100)
    {
        // Dịch chuyển các phần tử về phía sau 1 ô
        for (int i = n-1; i >= 0; i--)
        {
            a[i + 1] = a[i];
        }
        // Thêm vào đầu
        a[0] = x;
        n++;
    }
    else
    {
        cout << "Mang da day! Khong them them vao!\n";
    }
}

void them_vt_k(int a[], int& n, int x, int k)
{
    // Kiểm tra còn thêm vào mảng được không?
    if (n < 100)
    {
        // Dịch chuyển các phần tử về phía sau 1 ô bắt đầu từ vị trí k
        for (int i = n - 1; i >= k; i--)
        {
            a[i + 1] = a[i];
        }
        // Thêm vào vị trí k
        a[k] = x;
        n++;
    }
    else
    {
        cout << "Mang da day! Khong them them vao!\n";
    }
}

```



```
void xoa_cuoi(int a[], int& n)
{
    // giảm số lượng phần tử đi 1 đơn vị
    n--;
}

void xoa_dau(int a[], int& n)
{
    // Dịch chuyển các phần tử về phía trước 1 ô
    for (int i = 0; i < n; i++)
    {
        a[i] = a[i + 1];
    }
    // giảm số lượng phần tử đi 1 đơn vị
    n--;
}

void xoa_vt_k(int a[], int& n, int k)
{
    // Dịch chuyển các phần tử về phía trước 1 ô tính từ vị trí k trở về sau
    for (int i = k; i < n; i++)
    {
        a[i] = a[i + 1];
    }
    // giảm số lượng phần tử đi 1 đơn vị
    n--;
}
```