//1

#include <iostream>

using namespace std;

int square(int i){

    int sq = i \* i;

    return sq;

}

int main(){

    int i;

    cin >> i;

    int res = square(i);

    cout << res;

}

//2

#include <iostream>

#include <cmath>

#include <stdlib.h>

using namespace std;

float sqrti(float x) {

  if (x < 0) {

    cout << "Negative argument";

    exit(0);

  } else {

  return sqrt(x);

  }

}

int main() {

  int a;

  cout << "Введите число: ";

  cin >> a;

  cout << sqrti(a);

  return 0;

}

//3

#include <iostream>

#include <cmath>

#include <stdlib.h>

using namespace std;

double multiplyOrAdd1(double a, double b, bool add) {

  if (add) {

    return a + b;

  } else {

    return a \* b;

  }

}

double multiplyOrAdd2(double a, double b, bool add = true) {

  if (add) {

    return a + b;

  } else {

    return a \* b;

  }

}

int main() {

  double a = 5, b = 3;

  cout << "Сумма: " << multiplyOrAdd1(a, b, true) << endl;

  cout << "Произведение: " << multiplyOrAdd1(a, b, false) << endl;

  cout << "Сумма по умолчанию: " << multiplyOrAdd2(a, b) << endl;

  return 0;

}

//4

#include <iostream>

#include <cmath>

#include <stdlib.h>

using namespace std;

int countOdd(int arr[], int size) {

  int count = 0;

  for (int i = 0; i < size; i++) {

    if (arr[i] % 2 == 0) {

      count++;

    }

  }

  return count;

}

int main() {

  int n;

  cout << "Введите количество элементов массива: ";

  cin >> n;

  int\* arr = new int[n];

  for (int i = 0; i < n; i++) {

    cout << "Введите элемент массива: ";

    cin >> arr[i];

  }

  int oddCount = countOdd(arr, n);

  cout << "Количество четных элементов в массиве: " << oddCount << endl;

  delete[] arr;

  return 0;

}

//5

#include <iostream>

#include <stdlib.h>

using namespace std;

int sum(int n, int size, int arr[]) {

    if (n < 0 || n >= size) {

    cout << "Ошибка: индекс n вне допустимого диапазона." << endl;

    exit(0);

  }

  return arr[n] + n;

}

int main() {

  int arr[] = {1, 2, 3, 4};

  int n;

  int size = sizeof(arr) / sizeof(arr[0]);

  cout << "Введите номер элемента в массиве n: ";

  cin >> n;

  cout << "Сумма n-го элемента массива и n: " << sum(n,size, arr) << endl;

  return 0;

}

//6

#include <iostream>

using namespace std;

void deln(int& n, int arr[]){

    for (int i = 0; i < 4; i++){

        if (i==n){

            arr[i] = 0;

        }

    }

    n++;

    for (int i = 0; i < 4; i++){

        cout << arr[i] << " ";

    }

    cout << endl << n;

}

int main(){

    int n = 2;

    int arr[] = {1, 2, 3, 4};

    deln(n, arr);

    return 0;

}

//7

#include <iostream>

int compareStrings(const char\* str1, const char\* str2) {

    const char\* ptr1 = str1;

    while (\*ptr1 != '\0') {

        ++ptr1;

    }

    int length1 = ptr1 - str1;

    const char\* ptr2 = str2;

    while (\*ptr2 != '\0') {

        ++ptr2;

    }

    int length2 = ptr2 - str2;

    std::cout << length1 << std::endl;

    std::cout << length2 << std::endl;

    if (length1 == length2) {

        return 0;

    } else if (length1 < length2) {

        return -1;

    } else {

        return 1;

    }

}

int main() {

    const char\* str1 = "Broken";

    const char\* str2 = "World!";

    int result = compareStrings(str1, str2);

    std::cout << result << std::endl;

    return 0;

}

//8

#include <iostream>

int\* findCommonElements(int\* arr1, int size1, int\* arr2, int size2, int& commonSize) {

    static int commonElements[99];

    commonSize = 0;

    for (int\* ptr1 = arr1; ptr1 < arr1 + size1; ++ptr1) {

        for (int\* ptr2 = arr2; ptr2 < arr2 + size2; ++ptr2) {

            if (\*ptr1 == \*ptr2) {

                commonElements[commonSize++] = \*ptr1;

                break;

            }

        }

    }

    return commonElements;

}

int main() {

    int arr1[] = {1, 2, 3, 4, 5};

    int arr2[] = {3, 4, 5, 6, 7};

    int commonSize = 0;

    int\* commonElements = findCommonElements(arr1, 5, arr2, 5, commonSize);

    std::cout << "Общие элементы: ";

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

        std::cout << commonElements[i] << " ";

    }

    std::cout << "\nКоличество: " << commonSize << std::endl;

    return 0;

}

//9

#include <iostream>

void rotateArray(int\* arr, int size, int positions) {

    positions = positions % size;

    int\* temp = new int[positions];

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

        temp[i] = arr[i];

    }

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

        arr[i] = arr[i + positions];

    }

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

        arr[size - positions + i] = temp[i];

    }

    delete[] temp;

}

int main() {

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

    int size = sizeof(arr) / sizeof(arr[0]);

    int positions = 3;

    rotateArray(arr, size, positions);

    std::cout << "Сдвинутый массив: ";

    for (int i = 0; i < size; ++i) {

        std::cout << arr[i] << " ";

    }

    std::cout << std::endl;

    return 0;

}

//10

#include <iostream>

int mergeAndCompare(int\* arr1, int size1, int\* arr2, int size2, int\*& mergedArray, int& mergedSize) {

    if (size1 != size2) {

        return -1;

    }

    mergedSize = size1 + size2;

    mergedArray = new int[mergedSize];

    for (int i = 0; i < size1; ++i) {

        mergedArray[i] = arr1[i];

    }

    for (int i = 0; i < size2; ++i) {

        mergedArray[size1 + i] = arr2[i];

    }

    return 0;

}

int main() {

    int size1 = 5;

    int size2 = 5;

    int\* arr1 = new int[size1]{1, 2, 3, 4, 5};

    int\* arr2 = new int[size2]{6, 7, 8, 9, 10};

    int\* mergedArray = nullptr;

    int mergedSize = 0;

    int result = mergeAndCompare(arr1, size1, arr2, size2, mergedArray, mergedSize);

    if (result == -1) {

        std::cout << result << std::endl;

    } else {

        std::cout << "Объединённый массив: ";

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

            std::cout << mergedArray[i] << " ";

        }

        std::cout << "\nРазмер: " << mergedSize << std::endl;

    }

    delete[] arr1;

    delete[] arr2;

    delete[] mergedArray;

    return 0;

}