# Лучше всего использовать Вики-учебник «Реализации алгоритмов» <https://ru.wikibooks.org/wiki/%D0%A0%D0%B5%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%BE%D0%B2> или английский вариант <https://en.wikibooks.org/wiki/Algorithm_Implementation>

**Там много программ на всех более или менее известных языках.**

**Нужно выбрать 10 или более программ Вот три из них**

# Алгоритм Брона — Кербоша (<https://ru.wikibooks.org/wiki/%D0%A0%D0%B5%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%BE%D0%B2/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%91%D1%80%D0%BE%D0%BD%D0%B0_%E2%80%94_%D0%9A%D0%B5%D1%80%D0%B1%D0%BE%D1%88%D0%B0> )

Реализация на C++ с использованием стека.

**list<set<int> >kerbosh(int \*\*&a,int SIZE)**

**{**

**set <int> M,G,K,P;**

**list<set<int> > REZULT;**

**for (int i=0; i<SIZE;i++)**

**{**

**K.insert(i);**

**}**

**int v,Count=0,cnt=0;**

**int Stack1[100];**

**std::set<int> Stack2[100];**

**std::set<int>::iterator theIterator;**

**theIterator=K.begin();**

**while ((K.size()!=0)||(M.size()!=0))**

**{**

**if (K.size()!=0)**

**{**

**theIterator=K.begin();**

**v=\*theIterator;**

**Stack2[++Count]=M;**

**Stack2[++Count]=K;**

**Stack2[++Count]=P;**

**Stack1[++cnt]=v;**

**M.insert(v);**

**for (int i=0;i<SIZE;i++)**

**{**

**if (a[v][i])**

**{**

**theIterator=K.find(i);**

**if (theIterator!=K.end())**

**{**

**K.erase(theIterator);**

**}**

**theIterator=P.find(i);**

**if (theIterator!=P.end())**

**{**

**P.erase(theIterator);**

**}**

**}**

**}**

**theIterator=K.find(v);**

**if (theIterator!=K.end())**

**{**

**K.erase(theIterator);**

**}**

**}**

**else**

**{**

**if (P.size()==0)**

**{**

**REZULT.push\_back(M);**

**}**

**v=Stack1[cnt--];**

**P=Stack2[Count--];**

**K=Stack2[Count--];**

**M=Stack2[Count--];**

**theIterator=K.find(v);**

**if (theIterator!=K.end())**

**{**

**K.erase(theIterator);**

**}**

**P.insert(v);**

**}**

**}**

**return REZULT;**

**}**

**Алгори́тм Де́йкстры** — алгоритм на графах для нахождения кратчайшего расстояния от одной из вершин графа до всех остальных. Алгоритм работает только для графов без рёбер отрицательного веса. (<https://ru.wikibooks.org/wiki/%D0%A0%D0%B5%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%BE%D0%B2/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%94%D0%B5%D0%B9%D0%BA%D1%81%D1%82%D1%80%D1%8B> )

**//Алгоритм Дейкстры**

**void Djecstra(int st)**

**{**

**int \*\*w= new int\*[n];**

**for (int i=0;i<n;i++)**

**\*(w+i)=new int[n];**

**bool visited[n];**

**int D[n];**

**for(int i=0;i<n;i++)**

**{**

**D[i]=\*(\*(w+st)+i);**

**visited[i]=false;**

**}**

**D[st]=0;**

**int index=0,u=0;**

**for (int i=0;i<n;i++)**

**{**

**int min=INT\_MAX;**

**for (int j=0;j<n;j++)**

**{**

**if (!visited[j] && D[j]<min)**

**{**

**min=D[j];**

**index=i;**

**}**

**}**

**u=index;**

**visited[u]=true;**

**for(int j=0;j<n;j++)**

**{**

**if (!visited[j] && \*(\*(w+u)+j)!=INT\_MAX && D[u]!=INT\_MAX && (D[u]+\*(\*(w+u)+j)<D[j]))**

**{**

**D[j]=D[u]+\*(\*(w+u)+j);**

**}**

**}**

**}**

**cout<<"Стоимость пути из начальной вершины до остальных(Алгоритм Дейкстры):\t\n";**

**for (i=0; i<n; i++)**

**{**

**if (D[i]!=INT\_MAX)**

**cout<<st<<" -> "<<i<<" = "<<D[i]<<endl;**

**else**

**cout<<st<<" -> "<<i<<" = "<<"маршрут недоступен"<<endl;**

**}**

**for(int j=0;i<n;j++)**

**delete \*(w+j);**

**delete w;**

**}**

**Алгоритм Кнута — Морриса — Пратта**

### Реализация алгоритма на языке [Си](https://ru.wikibooks.org/w/index.php?title=%D0%A1%D0%B8&action=edit&redlink=1)

**int seek\_substring\_KMP (char s[], char p[])**

**{**

**int i, j, N, M;**

**N = strlen(s);**

**M = strlen(p);**

**int \*d =(int\*)malloc(M\*sizeof(int)); /\* динамический массив длины М\*/**

**/\* Вычисление префикс-функции \*/**

**d[0]=0;**

**for(i=1,j=0;i<M;i++)**

**{**

**while(j>0 && p[j]!=p[i])**

**j = d[j-1];**

**if(p[j]==p[i])**

**j++;**

**d[i]=j;**

**}**

**/\* поиск \*/**

**for(i=0,j=0;i<N; i++)**

**{**

**while(j>0 && p[j]!=s[i])**

**j=d[j-1];**

**if(p[j]==s[i])**

**j++;**

**if (j==M)**

**{**

**free (d); /\* освобождение памяти массива d \*/**

**return i-j+1;**

**}**

**}**

**free (d); /\* освобождение памяти массива d \*/**

**return -1;**

Можно также использовать Wiki-учебник «язык си в примерах» https://ru.wikibooks.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA\_%D0%A1%D0%B8\_%D0%B2\_%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D0%B0%D1%85

Вот еще две программы из интернета (не помню, откуда, но в случае чего найду)

**Программа №1**

**0 000**

**1 001**

**2 011**

**3 010**

**4 110**

**Ну а теперь полная формулировка задания:**

**Реализовать программу на с++, которая будет считавать последовательно два числа, представляющих из себя четырехбитные числа. И нужно будет определить являются ли введенные числа соседними числами по коду Грея.**

**{Соседние числа по коду Грея - числа отличающиеся на один бит}**

**Собственно реализация на с++**

**// 666.cpp: определяет точку входа для консольного приложения.**

**//**

**#include "stdafx.h"**

**#include**

**#include**

**#include**

**const int N=5;// количество символов в строке**

**char lin1[N],lin2[N];// строки содержащие коды Грея**

**int main()**

**{**

**int err(0);// переменная для проверки, что в введенных строках**

**// присутствуют только числа 0 и 1**

**setlocale(LC\_CTYPE,"rus");**

**std::cout<<"Введите первое число:";**

**std::cin.getline(lin1,N);// считывание символов в первую строку**

**// условие проверки, что в введенных строках**

**// присутствуют только числа 0 и 1**

**for(int i=0;i<4;i++){**

**if(lin1[i]=='1' || lin1[i]=='0' );**

**else{std::cout<<"error\n";**

**err++;**

**break;**

**}**

**}**

**std::cout<<"Введите второе число:";**

**std::cin.getline(lin2,N);// считываем вторую строку**

**for(int i=0;i<4;i++){**

**if(lin2[i]=='1' || lin2[i]=='0');**

**else{std::cout<<"error\n";**

**err++;**

**break;**

**}**

**}**

**int sum2(0),sum1(0);//сюда будем записывать**

**// сумму чисел содержащихся в коде Грея**

**for(int i=0;i<4;i++){**

**sum1+=lin1[i];**

**sum2+=lin2[i];**

**}**

**if(err==0){**

**if(sum1==sum2+1 || sum2==sum1+1)std::cout<<"Числа являются соседними по коду ГреЯ\n";**

**else std::cout<<"Числа НЕ являются соседними по коду ГреЯ";**

**}**

**return 0;**

**Программа №2**

[Хортон «Visual C++ Полный курс»](http://ucxodnuku.ru/skachat-knigi-po-programmirovaniyu/a-horton-visual-c-2010-polnyiy-kurs.html)(стр.88-92)

**#include "stdafx.h"**

**#include <iostream>**

**#include <fstream>**

**#include <clocale>**

**using namespace std;**

**int main(int argc, \_TCHAR\* argv[])**

**{**

**setlocale(LC\_CTYPE,"rus");**

**char nazv[100];// строковой массив в которое собственно говоря будет записываться название файла**

**cout<<"Введите название файла:";**

**cin>>nazv;**

**ifstream inpfile(nazv);// открываем файл для чтения**

**if(!inpfile.is\_open())**

**cout<<"Файл не может быть открыт";**

**else**

**{**

**int kolotkrck(0);//счетчик количества {**

**int kolzakrck(0);//счетчик количества } изначально равный нулю**

**char symv;// переменная для считывания символов из входного файла**

**inpfile>>symv; // считываем один символ из входного файла**

**while(inpfile)**

**{**

**if(symv=='{')kolotkrck++;**

**if(symv=='}')**

**{**

**kolzakrck++;**

**if(kolzakrck>kolotkrck)break;**

**}**

**inpfile>>symv;//считываем следующий символ**

**}**

**inpfile.close();**

**ofstream outfile("out.txt");//открываем выходной файл для записи**

**if(!outfile.is\_open())cout<<"Невозможно открыть файл";**

**if(kolotkrck==0 && kolzakrck==0)**

**{**

**cout<<"Открывающихся и закрывающихся скобок не наблюдается";**

**outfile<<"Открывающихся и закрывающихся скобок не наблюдается";**

**}**

**else**

**{**

**if(kolotkrck==kolzakrck)**

**{**

**cout<<"количество Открывающихся и закрывающихся скобок одинаково";**

**outfile<<"количество Открывающихся и закрывающихся скобок одинаково";**

**}**

**else**

**{**

**cout<<"количество Открывающихся и закрывающихся скобок различно";**

**outfile<<"количество Открывающихся и закрывающихся скобок различно";**

**}**

**}**

**}**

**return 0;**

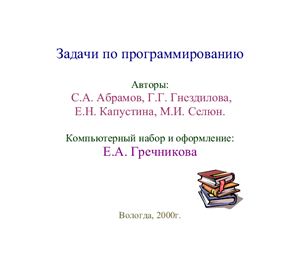
**}**

[**http://www.twirpx.com/file/66536/**](http://www.twirpx.com/file/66536/)

**Решебник. Абрамов С.А. Задачи по программированию *DOC* *JPG* *PDF* *TXT***

[Яндекс.Директ](https://direct.yandex.ru/?partner)

|  |
| --- |
| [Готовые домашние задания 2015](http://an.yandex.ru/count/DsI0nZ1glE440000ZhlSwau5XPZ21PK2cm5kGxS2Am68lGvwQmQ9kLoM6m6O__________yFdhS40004dQ-W3XESlaYAP0UMgWDVmBh-I8faBxXt_WgQiI4V0W6cQugsGR7O0RswtHmW0ge1fQXdS0IHkqhl4mkFjrQFBGwVjAMhH0MJXGsPLqACbTS3jf0d0RM0-g2Tk0QlbTS3ivYr0RI0-fIN71Adam21gAy0-pAam0000104hlrbuo5cy_OF1R2WgFtWiG6oWD2vhw0E4xlvITjzohSog071__________yFU1i0?test-tag=66657&stat-id=1073741895)ГДЗ для всех классов и предметов! Выбирайте решебник онлайн!18+[gdz2015.ru](http://an.yandex.ru/count/DsI0nZ1glE440000ZhlSwau5XPZ21PK2cm5kGxS2Am68lGvwQmQ9kLoM6m6O__________yFdhS40004dQ-W3XESlaYAP0UMgWDVmBh-I8faBxXt_WgQiI4V0W6cQugsGR7O0RswtHmW0ge1fQXdS0IHkqhl4mkFjrQFBGwVjAMhH0MJXGsPLqACbTS3jf0d0RM0-g2Tk0QlbTS3ivYr0RI0-fIN71Adam21gAy0-pAam0000104hlrbuo5cy_OF1R2WgFtWiG6oWD2vhw0E4xlvITjzohSog071__________yFU1i0?test-tag=66657&stat-id=1073741895) |



Есть много решенных задач. А так же в архиве есть сам сборник задач! Ко многим задачам приведены блок-схемы!  
Решенные номера: 21, 53, 54, 55, 56, 57А, 60В, 81, 83А, 93, 116В, 116Д, 119, 119Б, 119Д, 261Б, 269в, 314, 315А, 316А, 316В, 316, 316Д, 332, 336Г, 341, 345, 343, 350,  
350А, 350В, 367, 371, 374А, 389, 390Б, 391, 412, 420, 431, 444, 446, 460, 463, 480А, 482, 483, 485, 495, 496Б, 846Г, 854и, 866, 866А, 895, 901, 943, 951, 954, 957, 961, 962Б,

[**http://ptaskbook.com/ru/**](http://ptaskbook.com/ru/)

Programming Taskbook М. Э. Абрамян (Южный федеральный университет), 1998–2015

## Решения задач по программированию

<http://iproc.ru/2010/12/cs-solve/>

Aбрамов {Бесплатные решения}

http://abramov.org.ua/ru/

http://amberv.ru/skachat-zadachi-po-programmirovaniyu-na-paskale.php

**Программа №1**

**0 000**

**1 001**

**2 011**

**3 010**

**4 110**

**Ну а теперь полная формулировка задания:**

**Реализовать программу на с++, которая будет считавать последовательно два числа,**

**представляющих из себя четырехбитные числа. И нужно будет определить являются**

**ли введенные числа соседними числами по коду Грея.**

**{Соседние числа по коду Грея - числа отличающиеся на один бит}**

**Собственно реализация на с++**

**// 666.cpp: определяет точку входа для консольного приложения.**

**//**

**#include "stdafx.h"**

**#include**

**#include**

**#include**

**const int N=5;// количество символов в строке**

**char lin1[N],lin2[N];// строки содержащие коды Грея**

**int main()**

**{**

**int err(0);// переменная для проверки, что в введенных строках**

**// присутствуют только числа 0 и 1**

**setlocale(LC\_CTYPE,"rus");**

**std::cout<<"Введите первое число:";**

**std::cin.getline(lin1,N);// считывание символов в первую строку**

**// условие проверки, что в введенных строках**

**// присутствуют только числа 0 и 1**

**for(int i=0;i<4;i++){**

**if(lin1[i]=='1' || lin1[i]=='0' );**

**else{std::cout<<"error\n";**

**err++;**

**break;**

**}**

**}**

**std::cout<<"Введите второе число:";**

**std::cin.getline(lin2,N);// считываем вторую строку**

**for(int i=0;i<4;i++){**

**if(lin2[i]=='1' || lin2[i]=='0');**

**else{std::cout<<"error\n";**

**err++;**

**break;**

**}**

**}**

**int sum2(0),sum1(0);//сюда будем записывать**

**// сумму чисел содержащихся в коде Грея**

**for(int i=0;i<4;i++){**

**sum1+=lin1[i];**

**sum2+=lin2[i];**

**}**

**if(err==0){**

**if(sum1==sum2+1 || sum2==sum1+1)std::cout<<"Числа являются соседними по коду ГреЯ\n";**

**else std::cout<<"Числа НЕ являются соседними по коду ГреЯ";**

**}**

**return 0;**

**Программа №2**

[Хортон «Visual C++ Полный курс»](http://ucxodnuku.ru/skachat-knigi-po-programmirovaniyu/a-horton-visual-c-2010-polnyiy-kurs.html)(стр.88-92)

**#include "stdafx.h"**

**#include <iostream>**

**#include <fstream>**

**#include <clocale>**

**using namespace std;**

**int main(int argc, \_TCHAR\* argv[])**

**{**

**setlocale(LC\_CTYPE,"rus");**

**char nazv[100];// строковой массив в которое собственно говоря будет записываться название файла**

**cout<<"Введите название файла:";**

**cin>>nazv;**

**ifstream inpfile(nazv);// открываем файл для чтения**

**if(!inpfile.is\_open())**

**cout<<"Файл не может быть открыт";**

**else**

**{**

**int kolotkrck(0);//счетчик количества {**

**int kolzakrck(0);//счетчик количества } изначально равный нулю**

**char symv;// переменная для считывания символов из входного файла**

**inpfile>>symv; // считываем один символ из входного файла**

**while(inpfile)**

**{**

**if(symv=='{')kolotkrck++;**

**if(symv=='}')**

**{**

**kolzakrck++;**

**if(kolzakrck>kolotkrck)break;**

**}**

**inpfile>>symv;//считываем следующий символ**

**}**

**inpfile.close();**

**ofstream outfile("out.txt");//открываем выходной файл для записи**

**if(!outfile.is\_open())cout<<"Невозможно открыть файл";**

**if(kolotkrck==0 && kolzakrck==0)**

**{**

**cout<<"Открывающихся и закрывающихся скобок не наблюдается";**

**outfile<<"Открывающихся и закрывающихся скобок не наблюдается";**

**}**

**else**

**{**

**if(kolotkrck==kolzakrck)**

**{**

**cout<<"количество Открывающихся и закрывающихся скобок одинаково";**

**outfile<<"количество Открывающихся и закрывающихся скобок одинаково";**

**}**

**else**

**{**

**cout<<"количество Открывающихся и закрывающихся скобок различно";**

**outfile<<"количество Открывающихся и закрывающихся скобок различно";**

**}**

**}**

**}**

**return 0;**

**}**