

Спортивное программирование. Вводная лекция.

Плотников Даниил Михайлович, Закарлюка Иван Владимирович

Санкт-Петербургский Государственный Университет

Оглавление

① Ресурсы

② Асимптотика

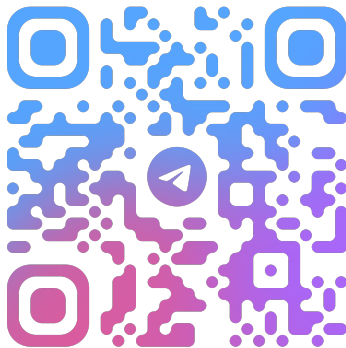
③ Основы

④ pref_sum

Связь


Материалы и обратная связь находятся в телеграм чате

<https://t.me/+HProdShRLdliZDZi>



Ресурсы


codeforces [codeforces.com]

 **CODEFORCES**
Sponsored by TON

Enter | Register

HOME TOP CATALOG CONTESTS GYM PROBLEMSET GROUPS RATING EDU API CALENDAR HELP RAVAN 🏆

Codeforces Round 1012 (Div.1, Div. 2, based on THUPC 2025 — Finals)

By [fengzhengwei](#), 2 days ago, 

There will, and there shall be a THU round!

Nihao Codeforces! After the unfortunate event which led to [Codeforces Round 1010 \(Div. 1, Unrated\)](#) and [Codeforces Round 1010 \(Div. 2, Unrated\)](#) being unrated last week, we're back and stronger than ever!

Tsinghua University's Student Association of Algorithmic Competitions (THUSAAC) is pleased to invite you to participate in [Codeforces Round 1012 \(Div. 1\)](#) and [Codeforces Round 1012 \(Div. 2\)](#) on [Sunday, March 23, 2025 at 08:35](#). This round will be **RATED** for everyone. You will be given 2 hours and 30 minutes to solve 5 problems for division 1 and 6 problems for division 2, and two of each will have subtasks. **Note the unusual time of the round.**

Our team of amazing problem setters include: [lam1789](#), [Ecrade_](#), [E.Space](#), [QuietBeautifulThoughts](#), [gyh20](#), [SpiritualKhorosho](#), [dapingguo8](#), [myee](#), [abruce](#), [_Lucien](#). Some of the problems prepared were not used in the final contest but we would still like to thank them nonetheless.

We would also like to express our gratitude to the following individuals:

- [KAN](#) for helping to host the round;
- [FairyWinx](#) for coordination and helping with the problem preparation;
- The team of problem setters for crosschecking the problems and also [donghanwen](#), [SnowySummer](#), [wangjinbo](#), [xiaoziyao](#), [turmax](#), [SomethingNew](#), [I_love_kirill22](#), [temporary1](#), [LeoPro](#), [gmusya](#), [KiruxaLight](#), [18o3](#), [mzt1](#) for testing the contest and providing us very useful feedback;

→ Pay attention

Before contest
[Codeforces Round 1013 \(Div. 3\)](#)
45:45:56
[Register now »](#)

→ Top rated

#	User	Rating
1	jiangly	3845
2	tourist	3798
3	orzdevinwang	3706
4	jqdai0815	3682
5	ksun48	3589
6	Ormlis	3532
7	Benq	3468
8	Radewoosh	3463
9	ecnerwala	3451
10	Um_nik	3450

[Countries](#) | [Cities](#) | [Organizations](#) [View all →](#)

acmp [acmp.com]

Школа программиста [English] 3/23/2025, 7:52:56 PM

новости [задачи] [курсы] [олимпиады]

информация

- О школе
- Правила
- Олимпиады
- Фотоальбом
- Гостевая
- Форум
- Паспорт
- Мои задачи
- Баланс
- Архив олимпиад

задачник

- Архив задач
- Состояние системы
- Рейтинг
- Задачи
- Курсы

методичка

- Новичкам
- Работа в системе
- Курсы ККДП
- Дистрибутивы
- Статьи
- Ссылки

статистика

новости

[09.01] Работа основного сайта восстановлена. Решения, отправленные на резервном сайте с 01.01.2025 по 09.01.2025, можно извлечь по адресу <https://w.acmp.ru>, они не будут там храниться вечно.

[25.12] Определены пороговые баллы для прохождения на региональный этап ВсОШ по информатике и ИКТ, полученные в результате участия на муниципальном этапе: 9 класс - 200, 10 класс - 300, 11 класс - 342. Региональный этап будет проходить на платформе codeforces.com.

[14.12] Завершилась ВКОШП - 2024/2025: [Результаты]

[29.11] Завершился муниципальный этап ВсОШ по информатике за 7-8 классы в Красноярском крае. Результаты: [7-8 классы] [9-11 классы]. Задачи: [7-8 классы] [9-11 классы]. Разборы: [7-8 классы] [9-11 классы].

[26.11] Определены пороговые баллы для прохождения учащихся г. Красноярска на муниципальный этап всероссийской олимпиады школьников по информатике: 7-е классы - 300 баллов, 8 классы - 400 баллов, 9-е - 160 баллов, 10-е классы - 200 баллов, 11-е классы - 205 баллов.

[17.11] Завершился полуфинал ВКОШП для Восточной Сибири: [Задачи] [Результаты].

[26.10] Завершился школьный этап ВсОШ по информатике: [Задачи]. Видеоразбор задач: [5-6 классы] [7-8 классы] [9-11 классы]. Результаты: [5-6 классы] [7-8 классы] [9-11 классы].

[23.10] Муниципальный этап ВсОШ Красноярского края по информатике состоится 29.11.2024 на данном ресурсе.

[04.10] Обновлено компиляторы: MinGW GNU C++ до 14.2.0, Python до 3.12.7, PyPy до 3.10 v7.3.17, PascalABC.NET до 3.10.0.3547 от 25.09.2024 и Go до 1.20.14.

[16.09] Приглашаем принять участие в командном соревновании по алгоритмическому программированию, которое состоится 28-29 сентября в ИКИТ. В этот раз участники смогут разрабатывать свои решения на современных (игровых) компьютерах. Подробнее: [Скачать].

[23.04] Красноярский краевой Дворец пионеров 03.05.2024 (пятница) проводит командную олимпиаду по программированию для школьников Красноярского края в рамках проекта "Миллигиб Еп+*". [Подробнее]

[22.01] Завершился региональный этап ВсОШ по информатике и ИКТ: [Результаты] [Задачи] [Фото] [Разбор].

1 2 3 4 5 6 7 8 9 10 11 12 »

Ученик: Плотноков Д.М. Выход

поиск по сайту

сейчас в online

- Агаева Э.Г.
- Адылбек
- Арсен Е.п.
- Артёмов К.А.
- Гаджимурадов Т.А.
- Зайцев Г.А.
- Канатбек С.
- Кривоносов П.С.
- Кузмина Ю.С.
- Неизвестный
- Неизвестный
- Неизвестный
- Оралбеков Нурасыл
- Павел В.П.
- Падалько И.Е.
- Плюшкин Е.А.
- Плюшкин Я.П.
- Покупай К.К.
- Пономарев И.Л.
- Райхон И.
- Сергей Н.П.
- Труфанов А.М.
- Федосов Я.Р.
- Чибирев Ахмадчон

Всего: 25

рейтинг

1	Андрей и К.	41914
2	Андрей Мясник	41914
3	Владимир И.Л.	41914
4	Зубков О.В.	41914
5	Тер-Саркисов Б.О.	41914
6	Степанов И.Д.	41862
7	Демидченко Виталий	41843
8	Пашментов Н.В.	41839
9	Яндулов Богдан	41744
10	Неизвестный	41371

Плотноков Д.М., Закарука И.В. (СПбГУ)

вводная лекция

2025

6 / 44

timus [acm.timus.ru]

ENG

RUS

Timus Online Judge



Online Judge
[О системе](#)
[Часто задаваемые вопросы](#)
[Новости сайта](#)
[Форум](#)
[Ссылки](#)

Задачи

[Архив задач](#)
[Отправить на проверку](#)
[Состояние проверки](#)
[Руководство](#)

Авторы

[Регистрация](#)
[Исправить данные](#)
[Рейтинг авторов](#)

Соревнования

[Текущее соревнование](#)
[Расписание](#)
[Прошедшие соревнования](#)
[Правила](#)

Добро пожаловать на Timus Online Judge,
архив задач с проверяющей системой

Timus Online Judge — это крупнейший в России архив задач по программированию с автоматической проверяющей системой. Основной источник задач для архива — соревнования Уральского федерального университета, Чемпионаты Урала, Уральские четвертьфиналы ICPC, Петрозаводские сборы по программированию. Чтобы начать решать задачи, прочитайте [руководство](#).

Timus Online Judge позволяет принять участие в онлайн-версиях большинства соревнований, которые регулярно проходят в Уральском федеральном университете. Перед участием в онлайн-соревнованиях ознакомьтесь с [правилами](#) их проведения.

Сайт создан и поддерживается студентами и выпускниками Уральского федерального университета.

Если вы хотите разместить свои задачи в архиве или провести онлайн-соревнование, напишите по адресу timus_support@acm.timus.ru.

© 2000-2025 [Timus Online Judge Team](#). Все права защищены.

Плотников Д.М., Закарюка И.В. (СПбГУ)

вводная лекция

2025

7 / 44

e-maxx [e-maxx.ru/algo]

MAXimal

[home](#)[algo](#)[bookz](#)[forum](#)[about](#)

algo

Здесь представлено 145 алгоритмов. Ко всем алгоритмам даны краткие описания и программы на C++.

Показать: последние добавленные, последние редактированные алгоритмы.

Вы также можете скачать PDF-книгу, в которой все статьи собраны в виде одного большого файла.


Алгебра (23)

элементарные алгоритмы (20)

- Функция Эйлера и её вычисление [TeX]
- Бинарное возведение в степень за $O(\log N)$ [TeX]
- Алгоритм Евклида нахождения НОД (наибольшего общего делителя) [TeX]
- Решето Эратосфена [TeX]
- Расширенный алгоритм Евклида [TeX]
- Числа Фибоначчи и их быстрое вычисление [TeX]
- Обратный элемент в кольце по модулю [TeX]
- Код Грея [TeX]
- Длинная арифметика [TeX]
- Дискретное логарифмирование по модулю M алгоритмом baby-step-giant-step Шэнкса за $O(\sqrt{M} \log M)$ [TeX]
- Диофантовы уравнения с двумя неизвестными: $AX+BY=C$ [TeX]
- Модульное линейное уравнение первого порядка: $AX=B$ [TeX]
- Китайская теорема об остатках. Алгоритм Гарнера [TeX]
- Нахождение степени делителя факториала [TeX]
- Трёхная сбалансированная система счисления [TeX]
- Вычисление факториала $N!$ по модулю P за $O(P \log N)$ [TeX]
- Перебор всех подмасок данной маски. Оценка 3^N для суммарного количества подмасок всех масок [TeX]
- Первообразный корень. Алгоритм нахождения [TeX]
- Дискретное квадратичное уравнение [TeX]

algorithmica [ru.algorithmica.org]





Алгоритмика

Алгоритмы

Анализ алгоритмов

Вычислительная сложность

- Асимптотический анализ
- Мастер-теорема

Сортировки

- Сортировка пузырьком
- Сортировка выбором
- Сортировка вставками
- Сортировка кучей
- Сортировка подсчетом
- Цифровая сортировка

Бинариск и интерактивки

- Бинарный поиск
- Бинарный поиск по ответу

Последовательности

- Сжатие координат

Декомпозиция задач

- Сканирующая прямая
- Отказывание состояний
- Корневые эвристики
- Алгоритмы Mo
- Meet-in-the-middle

Арифметика

- Битовое представление чисел
- Векторизация

Структуры данных

Базовые структуры данных

- Массивы и кортежи
- Итераторы
- Динамический массив
- Двоичная куча

Деревья поиска

- Деревья в STL
- Декартово дерево
- Неполный куч

Структуры для множеств

- Битсет и битовое сжатие
- Система непересекающихся множеств

usaco [usaco.org]

USA Computing Olympiad

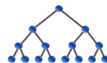
OVERVIEW

DETAILS / FAQ

TRAINING

HISTORY

RESOURCES



US OPEN CONTEST

The USACO 2025 US Open contest, our national championship, is available from March 21 through March 24. The contest is 5 hours in length, and can be taken any time during the larger 4-day contest window (see details on the contest page for special timing of "certified" contests for our gold and platinum divisions --- these are released at noon ET on March 22). The link below will take you to the contest page, containing further instructions as well as a button you can push to start your own personal contest timer.

[Contest Page](#)

2024-2025 COMPETITION SCHEDULE RELEASED

We are looking forward to a competition season in 2024-2025 with four contests, much like in previous years. At the end of the season, teams will be chosen to represent the USA at the International Olympiad in Informatics in Bolivia and the European Girls Olympiad in Informatics in Germany.

Important notes regarding changes to contest structure for this season, in the interest of ensuring academic integrity:

- Last year, we introduced the concept of a *certified* contest result --- obtained by taking the contest in a specific common time window when problems are first released. Since this has worked well, we are extending the use of certified contests to the gold division as well as the platinum division. In these divisions, problems will be first released at 12:00pm (noon) eastern time (ET) on the Saturday of the larger Friday-Monday contest timeline. Any student who begins the contest between 12:00pm and 12:15pm ET on that day will receive a *certified* score (treated by USACO staff as having higher potential credibility and weight than a non-certified score). Certified results are only possible in gold and platinum; the bronze and silver divisions will run as in the past, with problems available at the beginning of the Friday of the contest timeframe. In order to promote from gold to platinum, your score must be certified. In order to be considered as a contender for invitation to our summer training camp, you must have at least 3 certified scores out of our 4 monthly contests, one of which should be the final US Open contest (USACO coaches will consider any exceptional situations outside these parameters on a case-by-case basis).
- Recall an important addition to our contest *rules* from last year clarifying that generative AI is prohibited during our contests, and that USA students are not allowed to use VPNs to obscure their IP addresses.

YOUR ACCOUNT

Not currently logged in.

Username:**Password:**[Forgot password?](#)[Login](#)[Register for New Account](#)

2024-2025 SCHEDULE

Dec 13-16: First Contest

Jan 24-27: Second Contest

Feb 21-24: Third Contest

Mar 21-24: US Open

For each contest, USA students

wishing to receive a *certified* score in the *platinum* or *gold* contest must start Saturday at 12:00 ET, when problems in those divisions will be first released (see contest details for more detailed instructions)

May 22-31: Training Camp

Dates TBD: EGOI (Jul 14-20, Germany)

Dates TBD: IOI (Jul 27-Aug 3, Bolivia)

Асимптотика

Логарифм

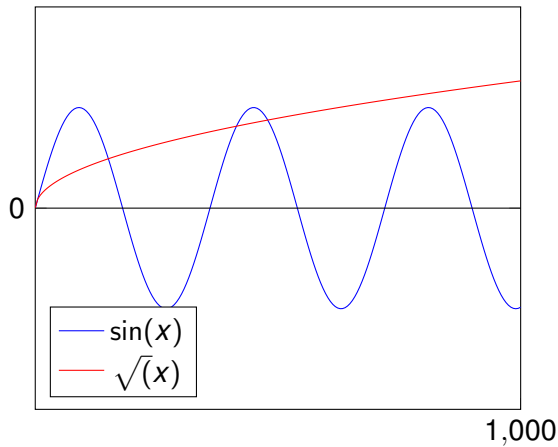
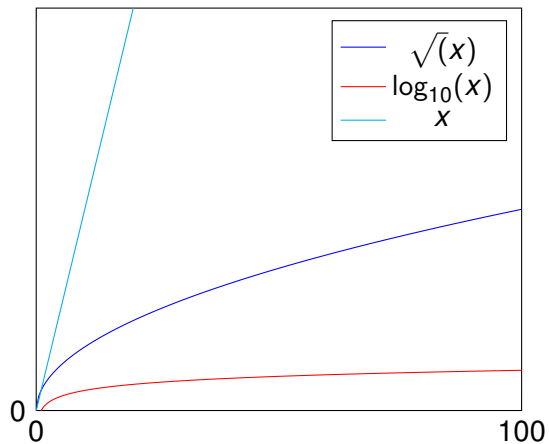
$x = \log_a b$ это такое число x , что $x^a = b$
 $4 = \log_4 16$

Определение

$$g(x) = O(f(x)) \Rightarrow \exists C = \text{const} : \exists x_1 : \forall x \geq x_1 \Rightarrow Cg(x) \geq f(x)$$

Функция $g(x)$ является асимптотической оценкой функции $f(x)$ тогда, когда существует константа, при которой найдётся такой x_1 , что для любого x , не меньшего x_1 , выполняется неравенство $Cg(x) \geq f(x)$

Определение



Частые асимптотики

Асимптотика	Возможный размер данных
a^n	примерно никогда
$n!$	10
n^3	500
n^2	10000
$n \log(n)$	10^6
n	10^8
\sqrt{n}	не хватит памяти
$\log n$	не хватит памяти
1	ваще жесьть

Как считать?

- 1 Заведём функцию сложности алгоритма
- 2 Посчитаем примерное количество операций
- 3 Держим в уме накладные расходы
- 4 Придумываем худший случай

Свойства

$$g(x) = O(f(x)) \Rightarrow \exists C = \text{const} : \exists x_1 : \forall x \geq x_1 \Rightarrow Cg(x) \geq f(x)$$

① $a = \text{const} \Rightarrow O(af(x)) = OO(f(x))$

② $O(n^2 + n + 8) = O(n^2)$

Важно помнить что на самом деле и константы и малые функции всё ещё влияют на временную сложность и могут привести к неверное оценке алгоритма

Почему нет основания логарифма?

$$O(\log_a n) = O\left(\frac{\log_b n}{\log_b a}\right) = \frac{1}{\log_b a} \log_b n$$

$$\frac{1}{\log_b a} = \text{const} \Rightarrow O\left(\frac{1}{\log_b a} \log_b n\right) = O(\log_b(n)) \Rightarrow O(\log_a n) = O(\log_b(n))$$

Опять же, стоит понимать что от основания на самом деле влияет скорость выполнения и для точной оценки его следует писать

Упражнения

Сортировка пузырьком

```
1  for (int i = 0; i < arr.size(); i++) {  
2      for (int j = 0; j < arr.size() - 1; j++) {  
3          if (arr[j] > arr[j + 1]) {  
4              int tmp = arr[j];  
5              arr[j] = arr[j + 1];  
6              arr[j + 1] = tmp;  
7          }  
8      }  
9  }
```

Упражнения

Сортировка пузырьком

```
1   for (int i = 0; i < arr.size(); i++) {  
2       for (int j = 0; j < arr.size() - 1; j++) {  
3           if (arr[j] > arr[j + 1]) {  
4               int tmp = arr[j];  
5               arr[j] = arr[j + 1];  
6               arr[j + 1] = tmp;  
7           }  
8       }  
9   }
```

Ответ: $O(n^2)$

Упражнения

```
1  int recursion(n){  
2      if(n < 1){  
3          return 1;  
4      }  
5  
6      int ans = 0;  
7      for(int i = 0; i < n){  
8          ans += recursion(n);  
9      }  
10  
11     return ans;  
12 }  
13
```

Упражнения

```
1  int recursion(int n){  
2      if(n < 1){  
3          return 1;  
4      }  
5  
6      int ans = 0;  
7      for(int i = 0; i < n){  
8          ans += recursion(n);  
9      }  
10  
11     return ans;  
12 }  
13
```

Ответ: $O(n^n)$

Упражнения

Дерево Фенвика

```
1  int sum (int r){  
2      int result = 0;  
3      for (int i = r; i >= 0; r = (i & (i+1)) - 1){  
4          result += t[r];  
5      }  
6      return result ;  
7  }  
8
```

Упражнения

Дерево Фенвика

```
1  int sum (int r){
2      int result = 0;
3      for (int i = r; i >= 0; r = (i & (i+1)) - 1){
4          result += t[r];
5      }
6      return result ;
7  }
8
```

Ответ: $O(\log n)$

Упражнения

Дерево Отрезков

```
1  int sum (int v, int tl, int tr, int l, int r) {  
2      if (l > r){  
3          return 0;  
4      }  
5      if (l == tl && r == tr){  
6          return t[v];  
7      }  
8      int tm = (tl + tr) / 2;  
9      return sum (v*2, tl, tm, l, min(r,tm))  
10     + sum (v*2+1, tm+1, tr, max(l,tm+1), r);  
11 }  
12
```

Упражнения

Дерево Отрезков

```
1  int sum (int v, int tl, int tr, int l, int r) {  
2      if (l > r){  
3          return 0;  
4      }  
5      if (l == tl && r == tr){  
6          return t[v];  
7      }  
8      int tm = (tl + tr) / 2;  
9      return sum (v*2, tl, tm, l, min(r,tm))  
10     + sum (v*2+1, tm+1, tr, max(l,tm+1), r);  
11 }  
12
```

Ответ: $O(\log n)$

Ссылка на тест в телеграме



Основы

Целые числа

существуют
полезные

Целые числа. Представление числа

Числа представляются в двоичной системе счисления

- $0 = 00000000$
- $8 = 00001000$
- $53 = 00110101$
- $255 = 01111111$

Целые числа. Побитовые операции

Побитовое не \sim

a	$\sim a$
0	1
1	0

Побитовое и $\&$

a	b	$a \& b$
0	0	0
0	1	0
1	0	0
1	1	1

Побитовое и $\&$

a	b	$a \& b$
0	0	0
0	1	0
1	0	0
1	1	1

Целые числа. Побитовые операции

Побитовое или |

a	b	a&b
0	0	0
0	1	1
1	0	1
1	1	1

Побитовый xor ^

a	b	a&b
0	0	0
0	1	1
1	0	1
1	1	0

Целые числа. Побитовые операции

Побитовый сдвиг вправо » и побитовый сдвиг влево «:

$$00010101 \ll 1 = 00101010$$

$$21 \ll 1 = 42 = 21 \times 2^1$$

$$0000011010100111 \gg 1 = 0000001101010011$$

$$1703 \gg 1 = 851 = 1703 / 2^1$$

Целочисленные типы данных в языке C++

Тип	Размер	Значения
short int	16	-32,768 — 32,767
unsigned short int	16	0 — 65,535
int	32	-2,147,483,648 — 2,147,483,647
unsigned int	32	0 — 4,294,967,295
long int	32	-2,147,483,648 — 2,147,483,647
unsigned long int	32	0 — 4,294,967,295
long long int	64	-9,223,372,036,854,775,808 — 9,223,372,036,854,775,807
unsigned long long int	64	0 — 18,446,744,073,709,551,615

Модификаторы: short, long, signed, unsigned

При подключении заголовочного файла `<cstdint>` можно писать `int[8/16/32/64/128]_t` и `uint[8/16/32/64/128]_t`

Отрицательные целые числа

Очевидная запись: от всех чисел отнимать константу. Эта константа будет минимальных числом.

Пример: пусть константа 52, а в памяти записано в двоичном виде число 42. Значит фактически число в этой переменной $42 - 52 = -10$

Решение рабочее, но медленное и неудобное.

Как же хранится число на самом деле?

Первый(самый старший) бит числа отвечает за знак. Тогда чтобы из положительного получить отрицательное нужно сделать операцию $-a = \sim a - 1$ но чем же это лучше?

Переполнения целых чисел

Что будет если в восьмибитном знаковом числе сложить 89 и 100?

Максимальное число это 127, что меньше этой суммы. Ответом будет -67.

Вернёмся к представлению числа чтобы понять почему так происходит.

$$87 = 01010111$$

$$100 = 01100100$$

$100 + 87 = 187 = 10111011$, что при первом бите ответственном за знак является $10111011 = -(01000100 - 00000001) = -67$. Не выглядит удобным. В чём же польза?

Переполнения целых чисел

Рассмотрим сумму чисел разных знаков

$$-87 = 10101001$$

$$100 = 01100100$$

$10101001 + 01100100 = 00001101 = 13$. Сумма сработала без каких либо дополнительных операций. Подобным свойством обладают также разность, умножение и деление. Фактически реализация разности и деления вовсе необязательна. На этом факте строиться Дерево Фенвика из примера асимптотического анализа алгоритма.

Ввод и вывод

```
1  #include <iostream>
2  using namespace std;
3  int main(){
4      std::ios_base::sync_with_stdio(0);
5      cin.tie(0);
6      cout.tie(0);
7
8      string s;
9      cin >> s;
10     cout << "Hello, mister " << s << "!";
11
12     cin.flush();
13     cout << "\n" << std::endl;
14
15     return 0;
16 }
17
```

Ввод и вывод

Ускорение ввода и вывода

```
1 std::ios_base::sync_with_stdio(0);  
2 cin.tie(0);  
3 cout.tie(0);  
4
```

Очистка буфера

```
1 cout.flush();  
2
```

НЕ ОДНО И ТО ЖЕ!!!

```
1 cout << "\n";  
2 cout << endl;  
3
```

Строки

Под капотом строки на самом деле это не более чем массив символов с одной особенностью. В конце находится терминальный символ с значением 0. Такую запись можно представить так:

```
1 char a[4] = {'1', '2', '3', 0};  
2 char b[4] = {'1', '2', '3', '\0'};  
3
```


Массив префиксных сумм

Задача

Есть массив. Требуется найти сумму всех элементов, индексы которых находятся в промежутке $[l, r]$.

Входные данные:

```
1 10
2 1 2 3 4 5 6 7 8 9 10
3 4
4 1 4
5 3 6
6 6 7
7 1 10
8
```

```
1 int n; cin >> n;
2 vector<int> a(n), b(n+1);
3 for (int i = 0; i < n; ++i){
4     cin >> a[i];
5 }
6 b[0] = 0;
7 for (int i = 1; i <= n; ++i){
8     b[i] = b[i-1] + a[i-1];
9 }
10 int t; cin >> t;
11 for (int i = 0; i < n; ++i){
12     int l, r; cin >> l >> r;
13     cout << b[r] - b[l-1] << '\n';
14 }
15
```