

Памятка студенту

Для ввода и вывода данных следует использовать стандартные потоки ввода и вывода. В каждом исходном файле **укажите, используя комментарии, ФИО (полностью), номер группы, номер варианта и букву задачи** (возможно, на некоторых компьютерах придется писать латиницей). Например:

```
/* student : Ivanov Sergey Petrovich
 * group   : 771
 * variant : 2
 * problem : A
 */
#include <stdio.h>
int main() {
    int n;
    scanf ("%d", &n);
    printf ("%d\n", n * n);
    return 0;
}
```

Решения должны корректно работать на всех входных данных, удовлетворяющих ограничениям задачи. Создайте директорию с именем **XXX-Familiyname**, где **XXX** — номер группы, а **Familiyname** — Ваша фамилия, набранная латиницей. Исходные файлы назовите **a.c**, **b.c**, **c.c**, **d.c** и поместите в эту директорию. Убедитесь с помощью тестирования, что ваши программы работают верно хотя бы для простых и крайних случаев. Сделайте это, даже если вы используете автоматическую проверяющую систему.

Подсказки и замечания

- Во многих задачках удобно использовать тип **unsigned**, который может хранить целые числа из полуинтервала $[0, 2^{32})$ (это верно для компьютеров в наших учебных классах). Этому типу соответствует формат **"%u"**.
- Следите, чтобы время работы вашей программы не превысило 1 секунды на одном тесте. Можно грубо считать, что 10 млн. простых операций соответствует одной секунде. Оцените время работы своей программы (желательно экспериментально) на самых больших допустимых входных данных.
- Проверьте работы своей программы как минимум на 5 различных входных данных самостоятельно. Помните, что за неудачную посылку снимается 4 балла.
- Если вам пришел ответ, что ваше посланное решение неверно, то ваше решение действительно неверно с вероятностью 0.999.
- На данной контрольной во многих задачах можно обойтись без динамического выделения памяти, так как ограничения на входные данные заранее известно. Ваше умение динамически выделять память под массивы будет отдельно проверяться преподавателем во время сдачи следующего задания.
- Не нужно проверять корректность входных данных. Считайте, что входные данные удовлетворяют условиям, описанным в задаче.
- Число и тип пробельных символов, которые вы использовали для разделения чисел на выходе, не учитываются проверяющей системой в большинстве задач. Если вы получили “Неверный ответ” (Wrong Answer), то скорее всего ошибка в самом результате, а не в форматировании.
- Ошибки, связанные с неправильным форматированием, система обычно отмечает как “Presentation Error”. Например, система ответит сообщением “Presentation Error”, если в ответе ожидается число, а вы ничего не вывели или вывели что-то, что не может интерпретироваться как число.

Критерии проверки

На контрольной будет 4 задачи. Для получения оценки 5 необходимо набрать 3 балла. Первые три задачи в каждом варианте имеют ценность 1 балл, а четвёртая задача — 2 балла. Общая формула оценки имеет вид

$$\langle \text{оценка} \rangle = 2 + \langle \text{число баллов} \rangle.$$

Таким образом, оценка — это дробное число от 2 до 7. Оценки 6 и 7 соответствуют 5+ и 5++.

Программа считается решением задачи, если

- программа компилируется компилятором языка Си — ANSI C, 99 (не Си++);
- не используются заголовочные файлы, отличные от `math.h`, `stdio.h`, `stdlib.h`, `ctype.h`, `alloc.h` (`malloc.h`), `string.h`;
- программа верно и эффективно работает на всех входных данных, удовлетворяющих условиям задачи, то есть менее чем за 1 секунду выводит на поток вывода требуемый ответ и завершает работу без ошибок выполнения.

Если данные критерии не удовлетворены, то за программу можно получить лишь половину или менее от полного количества баллов.

Если программа удовлетворяет данным критериям, то за неё можно получить половину от полного количества баллов или более.

Программы будут дополнительно оцениваться проверяющими по ряду критериев, определяемых проверяющим. Соответственно, количество баллов, поставленное за задачу может быть дробным. Примерный список критериев, которым *желательно* удовлетворить, следующий:

- Соответствие форматам ввода и вывода данных, указанных в условии задачи.
- Программа написана в хорошем стиле, с использованием отступов, для выделения блоков, и пробелов, для выделения бинарных операторов.
- Программа имеет простую, очевидную структуру, не нуждающуюся в комментариях.

Проверяющая система

Некоторым группам будет предоставлена возможность воспользоваться автоматической проверяющей системой:

<http://acm.mipt.ru/twiki/bin/view/Cintro/CTestMIPT2007>

Каждый студент должен войти по ссылке на свой вариант контрольной работы. Через web-интерфейс системы можно будет отправлять исходные файлы на языке C на проверку. Система автоматически скомпилирует их и запустит на нескольких входных данных.

Если все тесты пройдены успешно, то студент получит 50 или 100 очков. Первые три задачи имеют ценность 50 очков, а последняя – 100 очков. Возможно частичное решение задачи, когда часть тестов пройдено, а часть нет.

Обратите внимание, что при автоматической проверке необходимо скрупулёзно следовать указанным форматам входных и выходных данных.

Памятка преподавателям

Кто у кого проверяет, указано в файле

<http://acm.mipt.ru/twiki/pub/Info/CTestMIPT2007/instructors.doc>

Огромная просьба всем проверяющим прийти на контрольную, проследить за порядком, собрать результаты, проверить и послать результаты проверки контрольной на email'ы `petrov@mipt.ru` и `acm@acm.mipt.ru` в течение 7 рабочих дней после контрольной по форме

Группа: N

Основной преподаватель: ФИО

Проверяющий преподаватель: ФИО, контакты

- | | | | | | | | | |
|----|------|---------|----|----|----|----|-------|--------|
| 1. | ФИО | вариант | B1 | B2 | B3 | B4 | сумма | оценка |
| 2. | ФИО | вариант | B1 | B2 | B3 | B4 | сумма | оценка |
| 3. | | | | | | | | |

Здесь

- **ФИО** — фамилия, имя и отчество студента полностью;
- **вариант** — номер варианта набора задач;
- **B1, B2, B3, B4** — количество баллов, набранное за задачи A, B, C, D; количество баллов для каждой из первых трех задач — это действительное число из отрезка $[0, 1]$; количество баллов четвертой задачи — это действительное число из отрезка $[0, 2]$;
- **сумма** — сумма баллов за задачи;
- **оценка** — оценка за контрольную.

Распечатанные листки с задачами контрольной работы следует взять на кафедре заранее до занятий. После контрольной необходимо их собрать.

К заданию приложены “Памятка преподавателю”, “Памятка студентам” (5 штук) и “Критерии оценки” (1 штука). Содержимое последних двух необходимо довести до студентов.

С уважением, методическая комиссия 1-го курса.

Инструкции

Для предотвращения списывания и обмена решениями необходимо тщательно следить за выполнением правил.

- В первую очередь, студенты должны открыть в браузерах страницу с контрольной на сайте <http://acm.mipt.ru>. На этой странице есть правила участия и ссылки для входа в систему.
- Необходимо помочь открыть эту страницу, войти в систему автоматической проверки, зарегистрироваться незарегистрированным.
- Необходимо строго указать студентам следовать правилам, приведённым на странице контрольной. В частности:
 - Нельзя пользоваться локальной сетью.
 - Нельзя в браузере открывать страницы, отличные от страницы контрольной и системы автоматической проверки задач.
 - Нельзя переговариваться и обмениваться хранителями информации.
 - Каждый студент должен логиниться только под своим логином ровно с одного компьютера. Преподаватель должен следить, что в контекстной таблице нет лишних участников.
 - Все студенты должны указать в поле “Имя участника” свои настоящие имя и фамилию. При возникновении дубликатов или участников со странными именами, необходимо заходить в систему (ссылка “Вход для преподавателей” на странице рядом с вашим вариантом контрольной) и банить данного участника (открыть страницу “Посмотреть список обычных пользователей”, отметить на ней подозрительного участника и нажать кнопку “Установить флаг ЗАБЛОКИРОВАННОСТИ”).
 - Все студенты должны указать в поле “Имя участника” свои настоящие имя и фамилию. При возникновении дубликатов или участников с неизвестными именами, необходимо заходить в систему <http://su5.mipt.ru/cgi-bin/serve-control> (ссылка “Вход для преподавателей” на странице рядом с вашим вариантом контрольной) под логином viewer и паролем test2007 и банить данного участника.
- “Вход для преподавателей” позволяет смотреть решения участников, информацию о входных и выходных данных и корректировать набранные баллы.

Для доступа преподавателей следует использовать логин viewer, пароль test2007. В качестве “турнира” следует указывать номер группы.

- Необходимо обратить внимание студентов на страницу контрольной на <http://acm.mipt.ru> и раздел “Подсказки и замечания”.

“Вход для преподавателей” позволяет смотреть решения участников и информацию о входных и выходных данных.

При этом проверяющему преподавателю **разрешено и ставится в обязанность**:

- Отвечать на вопросы, касающиеся работы с системой.
- Отвечать на вопросы по условию задач.
- Пояснять условия задач и давать дополнительные примеры входных и выходных данных.
- Подсказывать возможные крайние значения входных данных, которые требуют особого рассмотрения.
- Подсказывать верные математические факты, которые необходимы для решения задачи.

Необходимо давать всем нужное количество подсказок и сообщать студентам о причинах непрохождения их решений, чтобы лучшие студенты в группе решили по крайней мере 3 задачи.

Контрольная в значительной степени имеет целью повысить мотивацию студентов к более глубокому изучению языка Си, а не проверить их текущие знания.

Вариант №1.

Задача А. (Простые числа) Напишите программу, которая из списка натуральных чисел выделяет простые. Число 1 не считается простым.

Формат ввода. Число N , $N \in \mathbb{N}$, $0 < N < 10000$ и N натуральных чисел из отрезка $[1, 100000]$.

Формат вывода. Простые числа из данных в том порядке, в котором они даны на входе.

| stdin | stdout |
|---------------|--------|
| 4 3 9 11 3 | 3 11 3 |

Задача В. (Число, свободное от квадратов) Напишите программу, которая определяет, есть ли среди делителей числа полные квадраты больше 1.

Формат ввода. Натуральное число N , $0 < N < 2^{31}$.

Формат вывода. Выведите YES (если есть) или NO (иначе).

| stdin | stdout |
|-------|--------|
| 4 | YES |
| 12 | YES |
| 15 | NO |

Задача С. (Взвешивание) Напишите программу, которая проверяет, можно ли данное число “взвесить” на двухчашечных весах гирьками с весами $1, 4^1, 4^2, \dots$. Каждая гирька есть только в одном экземпляре. Известно, что числа вида $4 \cdot n + 2$ невозможно взвесить указанным образом. Числа $4 \cdot n$, $4 \cdot n + 1$, $4 \cdot n - 1$ можно взвесить тогда и только тогда, когда можно взвесить n .

Формат ввода. Натуральное число N , $0 \leq N < 2^{31}$.

Формат вывода. Одна строка, содержащая YES или NO.

| stdin | stdout |
|-------|--------|
| 59 | YES |
| 58 | NO |

Задача D. (Таблица количеств хороших слов) Напишите программу, которая выводит таблицу чисел $b(n, k)$ ($0 \leq n, k \leq N$), где $b(n, k)$ есть количество бинарных слов длины n , в которых ровно k единиц и нет двух подряд идущих единиц.

Формат ввода. Целое число N , $0 \leq N < 48$.

Формат вывода. Выведите $N + 1$ строку. Строка с номером n ($n = 0, 1, \dots, N$) содержит числа $b(n, 0), b(n, 1), \dots$ до последнего ненулевого числа включительно.

| stdin | stdout |
|-------|-----------------------------------|
| 4 | 1 1 1 1 2 1 3 1 1 4 3 |

Вариант №2.

Задача А. (Транспонирование) Напишите программу, которая транспонирует введенную таблицу целых чисел. Память для хранения чисел выделяйте динамически.

Формат ввода. В первой строке дано натуральное число N — размер таблицы. $0 < N < 50$. В следующих N строках дано по N чисел, разделённых пробелом. Каждое число по модулю менее 2^{31} .

Формат вывода. Таблица чисел, в которой число, стоящее в i -й строке в j -м столбце равно числу, стоящее в j -й строке в i -м столбце данной таблицы.

| stdin | stdout |
|-------|--------|
| 3 | 1 4 7 |
| 1 2 3 | 2 5 8 |
| 4 5 6 | 3 6 9 |
| 7 8 9 | |

Задача В. (Нормализация корня) Нормализацией корня \sqrt{n} , $n \in \mathbb{N}$, назовём представление $\sqrt{n} = b\sqrt{a}$, где $a, b \in \mathbb{N}$ и b максимально. Напишите программу, которая для данного натурального числа n находит пару чисел (a, b) .

Формат ввода. Натуральное число n , $1 \leq n < 2^{32}$.

Формат вывода. Выведите числа a и b , разделив их пробелом.

| stdin | stdout |
|-------|--------|
| 24 | 6 2 |
| 36 | 1 6 |

Задача С. (N -е простое число) Напишите программу, которая вычисляет N -е простое число. Число 1 не считается простым.

Формат ввода. Натуральное число N , $0 < N \leq 10000$.

Формат вывода. Выведите простое число p_N .

| stdin | stdout |
|-------|--------|
| 5 | 11 |
| 1 | 2 |

Задача D. (Число разложений без повторений) Напишите программу, которая считает количество разложений $Q(N)$ данного натурального числа N на неупорядоченные натуральные слагаемые без повторений. Например, для $N = 5$ есть 3 различных разложения: $5 = 5 = 4 + 1 = 3 + 2$. Разложения считаются различными если множества слагаемых различаются. Реализуйте функцию $q(n, k)$, возвращающую количество разложений числа n на неповторяющиеся слагаемые, которые меньше либо равны k . Заметьте, что для $1 < k \leq n$ верна формула $q(n, k) = q(n - k, k - 1) + q(n, k - 1)$. Кроме того, $q(1, k) = q(0, k) = 1$ при $k > 0$, и $q(n, k) = 0$ при $k < 0$.

На основе этих соотношений напишите рекурсивную функцию q с запоминанием вычисленных значений. Для хранения вычисленных значений $q(n, k)$ используйте двумерный глобальный массив. Искомое число равно $Q(N) = q(N, N)$.

Формат ввода. Натуральное число N , $0 < N < 500$.

Формат вывода. Число $Q(N)$.

| stdin | stdout |
|-------|--------|
| 5 | 3 |
| 15 | 27 |

Вариант №3.

Задача А. (НОК трёх чисел) Напишите программу, которая находит наименьшее общее кратное трех данных чисел.

Формат ввода. Три натуральных числа a , b и c , $0 < a, b, c < 2^{32}$.

Формат вывода. $\text{НОК}(a, b, c)$. Известно, что $\text{НОК}(a, b, c)$ не превосходит $2^{32} - 1$

| stdin | stdout |
|----------|--------|
| 24 10 3 | 120 |
| 12 45 18 | 180 |

Задача В. (Вывод простых чисел по номеру) Напишите программу, которая вычисляет простые числа по их номерам. Число 1 не считается простым.

Формат ввода. Натуральное число N , $0 < N \leq 10000$, и N натуральных чисел a_1, \dots, a_N из отрезка $[1, 10000]$.

Формат вывода. Выведите N простых чисел p_{a_1}, \dots, p_{a_N} , разделив их пробелом.

| stdin | stdout |
|------------------|-------------|
| 3 3 2 1 | 5 3 2 |
| 2 1000 100000 | 7919 104729 |

Задача С. (Сумма четырёх квадратов) Известно, что любое натуральное число N можно представить в виде суммы четырёх квадратов неотрицательных чисел. Напишите программу, которая находит это разложение.

Формат ввода. Натуральное число N , $0 < N \leq 2\,000\,000$.

Формат вывода. Четыре полных квадрата a^2, b^2, c^2, d^2 , разделенных пробелами, такие что $a^2 + b^2 + c^2 + d^2 = N$, $0 \leq a^2 \leq b^2 \leq c^2 \leq d^2$. Если существует несколько разложений, то выведите одно из них.

| stdin | stdout |
|-------|-----------|
| 7 | 1 1 1 4 |
| 100 | 0 1 25 81 |

Задача D. (Максимальный квадрат) Напишите программу, которая находит на данной квадратной карте $N \times N$ белый квадрат максимального размера. Карта разбита на ячейки 1×1 . Каждая ячейка покрашена в чёрный или белый цвет. Реализуйте рекурсивную функцию `int max_square(int x, int y)` с запоминанием вычисленных значений, которая возвращает размер максимального белого квадрата с правой нижней вершиной в ячейке с координатами (x, y) (координаты увеличиваются вниз (y) и вправо (x)). Подсказка: выразите `max_square(x, y)` через `max_square(x-1, y)`, `max_square(x, y-1)`, `max_square(x-1, y-1)` и цвет ячейки с координатами (x, y) .

Формат ввода. Натуральное число N , $0 < N < 100$, а затем N строк по N символов. Символ '#' обозначает чёрный цвет, а символ '.' — белый.

Формат вывода. Размер стороны максимального белого квадрата и координаты его правой нижней вершины. Координаты отсчитываются с нуля. Если есть несколько вариантов ответа, выведите тот из них, который имеет меньшую координату x , а при равных x — тот, который имеет меньшую координату y .

| stdin | stdout |
|--|----------|
| 5 #...# ##... #.... #.### | 3 4 3 |

Вариант №4.

Задача А. (НОК трёх чисел) Напишите программу, которая находит наименьшее общее кратное трех данных чисел.

Формат ввода. Три натуральных числа a, b и c , $0 < a, b, c < 2^{32}$.

Формат вывода. НОК(a, b, c). Известно, что НОК(a, b, c) не превосходит $2^{32} - 1$

| stdin | stdout |
|----------|--------|
| 24 10 3 | 120 |
| 12 45 18 | 180 |

Задача В. (Фильтр символов) Напишите программу, которая считывает с входного потока строки символов, выводит их в той же последовательности и осуществляет следующие преобразования каждой строки:

- пропускает знаки препинания,
- последовательность подряд идущих пробельных символов (символов ' ', '\t') заменяет на один пробел,
- большие латинские буквы преобразует в малые,
- цифры заменяет на символ '#'

Формат ввода. Несколько строк текста (до поступления символа конца потока EOF).

Формат вывода. Преобразованные строки.

| stdin | stdout |
|------------------|----------------|
| Hi, 15 students! | hi ## students |

Задача С. (Перевоорачивание слов) Напишите программу, которая перевоорачивает все слова в строке.

Формат ввода. Одна строка, завершающаяся символом конца строки '\n'. Длина строки менее 1000.

Формат вывода. Та же самая строка, только все слова должны быть “перевернуты”. Символы, отличные от латинских букв должны остаться на своих местах.

| stdin | stdout |
|---|---|
| Answer to Life, the Universe, and Everything! | rewsnA ot efiL, eht esrevinU, dna gnihtyrevE! |

Задача D. (Число разложений без повторений) Напишите программу, которая считает количество разложений $Q(N)$ данного натурального числа N на неупорядоченные натуральные слагаемые без повторений. Например, для $N = 5$ есть 3 различных разложения: $5 = 5 = 4 + 1 = 3 + 2$. Разложения считаются различными если множества слагаемых различаются. Реализуйте функцию $q(n, k)$, возвращающую количество разложений числа n на неповторяющиеся слагаемые, которые меньше либо равны k . Заметьте, что для $1 < k \leq n$ верна формула $q(n, k) = q(n - k, k - 1) + q(n, k - 1)$. Кроме того, $q(1, k) = q(0, k) = 1$ при $k > 0$, и $q(n, k) = 0$ при $k < 0$.

На основе этих соотношений напишите рекурсивную функцию `q` с запоминанием вычисленных значений. Для хранения вычисленных значений $q(n, k)$ используйте двумерный глобальный массив. Искомое число равно $Q(N) = q(N, N)$.

Формат ввода. Натуральное число N , $0 < N < 500$.

Формат вывода. Число $Q(N)$.

| stdin | stdout |
|-------|--------|
| 5 | 3 |
| 15 | 27 |

Вариант №5.

Задача А. (Сумма ряда экспоненты) Напишите программу, которая находит частичную сумму ряда $a_k = \frac{x^k}{k!}$:

$$S(n, x) = \sum_{k=0}^n \frac{x^k}{k!}.$$

Формат ввода. Строка, содержащая натуральное число N и действительное число x .
 $0 < N \leq 10^6$, $-10.0 \leq x \leq 10.0$.

Формат вывода. Значение $S(N, x)$ с точностью до 6 знаков после десятичной точки.

| stdin | stdout |
|--------------|----------|
| 2 3 | 8.500000 |
| 100000 0.346 | 1.413403 |

Задача В. (Сумма делителей) Напишите программу, которая находит сумму всех делителей натурального числа N , включая 1 и N .

Формат ввода. Натуральное число N , $0 < N < 2^{28}$.

Формат вывода. Сумма делителей числа N , включая 1 и N .

| stdin | stdout |
|-------|--------|
| 5 | 6 |
| 10 | 18 |

Задача С. (Упорядочивание студентов) Напишите программу, которая считывает информацию о студентах (имя, оценка по информатике) и выводит их сначала в порядке, отсортированном по фамилии (в алфавитном порядке), а затем в порядке отсортированном по оценке (по убыванию). Все имена разные, состоят из маленьких латинских букв и имеют длину не более 29 букв. При упорядочивании по оценке в случае, если два студента имеют одну и ту же оценку, необходимо первым вывести того, у кого имя идет в алфавитном порядке раньше.

Формат ввода. Натуральное число N , $0 < N < 10000$. Затем N строк, каждая из которых содержит имя студента и его оценку – целое число из отрезка $[1, 10]$.

Формат вывода. Выведите $2 \cdot N + 1$ строк. Между двумя списками выведите пустую строку.

| stdin | stdout |
|---------------------------------|--|
| 3 lev 5 alla 4 boris 5 | alla 4 boris 5 lev 5 boris 5 lev 5 alla 4 |

Задача D. (Число хороших слов) Напишите программу, которая считает количество бинарных слов $b(N, K)$ длины N , в которых ровно K единиц и нет двух подряд идущих единиц.

Формат ввода. Два натуральных числа N и K , $0 \leq N$, $K < 48$.

Формат вывода. Одно целое число – ответ к задаче.

| stdin | stdout |
|-------|--------|
| 4 2 | 3 |
| 15 4 | 495 |

Вариант №6.

Задача А. (Зеркальные отображения матрицы) Напишите программу, которая зеркально отображает введенную таблицу целых чисел по горизонтали и по вертикали. Память для хранения чисел выделяйте динамически.

Формат ввода. В первой строке даны натуральные числа N и M — размеры таблицы. $0 < N, M < 50$. В следующих M строках дано по N чисел, разделённых пробелом. Каждое число по модулю менее 2^{31} .

Формат вывода. Таблица чисел, в которой число, стоящее в i -й строке в j -м столбце равно числу, стоящее в $M - i + 1$ -й строке в $N - j + 1$ -м столбце данной таблицы.

| stdin | stdout |
|----------|----------|
| 3 4 | 12 11 10 |
| 1 2 3 | 9 8 7 |
| 4 5 6 | 6 5 4 |
| 7 8 9 | 3 2 1 |
| 10 11 12 | |

Задача В. (Четырёхзначные числа) Напишите программу, которая находит все четырёхзначные числа (натуральные числа из отрезка $[1000, 9999]$), сумма цифр которых равна заданному числу N .

Формат ввода. Целое число N , $0 \leq N \leq 10000$.

Формат вывода. Выведите все найденные числа в возрастающем порядке. Затем выведите их число.

| stdin | stdout |
|-------|---|
| 33 | 6999 7899 7989 7998 8799 8889 8898 8979 8988 8997 9699 9789 9798 9879 9888 9897 9969 9978 9987 9996 20 |

Задача С. (Числа, свободные от квадратов) Напишите программу, которая среди введённых чисел находит числа, среди делителей которых нет полных квадратов больше 1.

Формат ввода. Натуральное число N , $0 < N \leq 200$, и N натуральных чисел из отрезка $[1, 2^{31} - 1]$.

Формат вывода. Выведите введенные числа, которые не имеют полных квадратов среди своих делителей, в том порядке, в котором они даны на входе.

| stdin | stdout |
|--------|--------|
| 3 | 5 |
| 4 5 12 | |

Задача D. (Число разложений) Напишите программу, которая считает количество разложений $P(N)$ данного натурального числа N на неупорядоченные натуральные слагаемые. Например, для $N = 5$ есть 7 различных разложений: $5 = 4 + 1 = 3 + 2 = 3 + 1 + 1 = 2 + 2 + 1 = 2 + 1 + 1 + 1 = 1 + 1 + 1 + 1 + 1$. Разложения считаются различными если множества слагаемых различаются. Реализуйте функцию $p(n, k)$, возвращающую количество разложений числа n на слагаемые, которые меньше либо равны k . Заметьте, что для $1 \leq k \leq n$ верна формула $p(n, k) = p(n - k, k) + p(n, k - 1)$, и $p(n, 1) = 1$. На основе этих соотношений напишите рекурсивную функцию p с запоминанием вычисленных значений. Для хранения вычисленных значений $p(n, k)$ используйте двумерный глобальный массив. $P(N) = p(N, N)$.

Формат ввода. Натуральное число N , $0 < N < 100$.

Формат вывода. Число $P(N)$.

| stdin | stdout |
|-------|--------|
| 5 | 7 |
| 15 | 176 |

Вариант №7.

Задача А. (Сумма степеней двойки) Напишите программу, которая для данного натурального числа N предьявляет его разложение на сумму неповторяющихся степеней двойки.

Формат ввода. Натуральное число N , $0 < N < 2^{31}$.

Формат вывода. Степени двойки, на сумму которых разлагается N , перечисленные через пробел в возрастающем порядке.

| stdin | stdout |
|-------|----------------------|
| 31 | 1 2 4 8 16 |
| 10 | 2 8 |
| 256 | 256 |
| 255 | 1 2 4 8 16 32 64 128 |

Задача В. (Максимально удалённые точки) Напишите программу, которая во множестве введённых точек на плоскости находит пару максимально удалённых друг от друга точек. Для хранения точек используйте динамически выделенную память.

Формат ввода. В первой строке дано число точек N , $1 < N < 1000$. Затем следуют N строк, в каждой из которых дана пара вещественных координат точек с точностью до двух значащих цифр после запятой. Координаты точек по модулю не превосходят 10000.

Формат вывода. Выведите номера двух искомых точек. Точки нумеруются от 1 до N в соответствии с порядком, в котором они задавались во входе. Если максимально удаленных пар точек несколько, выведите любую из них. Выведите также значение расстояния между ними с точностью до 6 значащих цифр после запятой.

| stdin | stdout |
|-----------------------------------|------------------|
| 4 1.5 2 -3 4 4 6 6 -1 | 4 2 10.295630 |

Задача С. (Вывод простых чисел по номеру) Напишите программу, которая вычисляет простые числа по их номерам. Число 1 не считается простым.

Формат ввода. Натуральное число N , $0 < N \leq 10000$, и N натуральных чисел a_1, \dots, a_N из отрезка $[1, 10000]$.

Формат вывода. Выведите N простых чисел p_{a_1}, \dots, p_{a_N} , разделив их пробелом.

| stdin | stdout |
|------------------|-------------|
| 3 3 2 1 | 5 3 2 |
| 2 1000 100000 | 7919 104729 |

Задача D. (Генерация хороших слов) Напишите программу, выводит бинарные слова длины N , в которых ровно K единиц и нет двух подряд идущих единиц. Объявите глобальный массив `char word[30]` и реализуйте рекурсивную функцию $g(n, k, i)$, которая в “подмассиве” `word[i], word[i+1], ..., word[i+n]` перебирает различные варианты хороших слов длины n с k единицами.

Формат ввода. Два натуральных числа N и K : $0 \leq N, K < 25, K \leq (N + 1)/2$.

Формат вывода. Все хорошие слова, упорядоченные по убыванию. Каждое слово выведите в отдельное строке.

| stdin | stdout |
|-------|----------------------|
| 4 2 | 1010 1001 0101 |

Вариант №8.

Задача А. (Таблица степеней по модулю N) Напишите программу, которая выводит таблицу остатков от деления чисел a^k на N . В k -й строке в столбце с номером a должен стоять остаток от деления числа a^k на число N , $1 \leq a \leq (N-1)$, $0 \leq k \leq (N-1)$. Считайте, что строки нумеруются с 0, а столбцы с 1.

Формат ввода. Натуральное число N , $2 \leq N \leq 100$.

Формат вывода. N строк. В каждой строке $(N-1)$ чисел, разделённых пробелами.

| stdin | stdout | | | |
|-------|--------|---|---|---|
| 5 | 1 | 1 | 1 | 1 |
| | 1 | 2 | 3 | 4 |
| | 1 | 4 | 4 | 1 |
| | 1 | 3 | 2 | 4 |
| | 1 | 1 | 1 | 1 |

Задача В. (Полные квадраты) Напишите программу, которая среди введённых чисел находит числа, являющиеся полными квадратами. При решении реализуйте и используйте функцию `int is_square(int n)`, которая возвращает 1, если натуральное число n является квадратом некоторого натурального числа k , а иначе возвращает 0.

Формат ввода. Натуральное число N , $0 < N < 100$, и N натуральных чисел меньших 2^{31} .

Формат вывода. Выведите те введённые числа, которые являются полными квадратами, в том порядке, в котором они даны на входе.

| stdin | stdout |
|---|----------|
| 18 | 1 4 9 16 |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 | |
| 18 | |

Задача С. (Сумма четырёх квадратов) Известно, что любое натуральное число N можно представить в виде суммы четырёх квадратов неотрицательных чисел. Напишите программу, которая находит это разложение.

Формат ввода. Натуральное число N , $0 < N \leq 2\,000\,000$.

Формат вывода. Четыре полных квадрата a^2 , b^2 , c^2 , d^2 , разделённых пробелами, такие что $a^2 + b^2 + c^2 + d^2 = N$, $0 \leq a^2 \leq b^2 \leq c^2 \leq d^2$. Если существует несколько разложений, то выведите одно из них.

| stdin | stdout |
|-------|-----------|
| 7 | 1 1 1 4 |
| 100 | 0 1 25 81 |

Задача D. (Количество чисел с данной суммой цифр) Напишите программу, которая находит количество $c(N, K)$ упорядоченных наборов K цифр (элементов множества $\{0, 1, \dots, 9\}$) с суммой, равной N . При $k \geq 1$ и $n \geq 9$ верна следующая рекуррентная формула:

$$c(n, k) = c(n, k-1) + c(n-1, k-1) + \dots + c(n-9, k-1).$$

Определите, чему равны значения $c(n, 1)$ и $c(0, k)$ и уточните рекуррентную формулу для $n < 9$. Для хранения вычисленных значений $c(n, k)$ используйте глобальный массив.

Формат ввода. Целые числа N и K , $0 < K \leq 10$, $0 \leq N \leq 90$.

Формат вывода. Число $c(N, K)$.

| stdin | stdout |
|-------|--------|
| 10 3 | 63 |

Вариант №9.

Задача А. (Таблица наибольших общих делителей) Напишите программу, которая выводит таблицу наибольших общих делителей $\text{НОД}(a, b)$ натуральных чисел a и b , для всех a и b таких, что $1 \leq a \leq N$, $1 \leq b \leq N$.

Формат ввода. Натуральное число N , $0 < N < 20$.

Формат вывода. Таблица чисел $\text{НОД}(a, b)$. Столбцы должны быть выровнены по левому краю.

Задача В. (Поиск повторений) Напишите программу, которая считывает N чисел и проверяет, есть ли среди них повторяющиеся. Для хранения чисел используйте динамически выделенную память.

Формат ввода. В первой строке дано натуральное число N , $0 < N < 1000$. В следующей строке даны N натуральных чисел, не превосходящих по модулю 2^{31} .

Формат вывода. Список чисел, которые во входе присутствуют более 1-го раза, в том порядке, в котором они встретились во входе. Каждое число выведите в отдельной строке и через пробел укажите число раз, которое оно присутствует во входе.

Задача С. (Сумма больших двоичных чисел) Напишите программу, которая суммирует большие натуральные числа, заданные в двоичной системе счисления.

Формат ввода. Две строки, содержащие двоичные записи натуральных чисел a и b . Число цифр в каждой записи не более 1000.

Формат вывода. Двоичная запись числа $a + b$.

Во входе двоичные числа могут иметь перед значащими цифрами несколько нулей.

| stdin | stdout |
|----------------|--------|
| 11111 1 | 100000 |
| 10 11 | 101 |
| 10000 01111 | 11111 |

Задача D. (Генерация разложений без повторений) Напишите программу, генерирует разложения данного натурального числа N на неупорядоченные натуральные слагаемые без повторений. Например, для $N = 5$ есть 3 различных разложения: $5 = 5 = 4 + 1 = 3 + 2$. Разложения считаются различными если множества слагаемых различаются.

Формат ввода. Натуральное число N , $0 < N \leq 60$.

Формат вывода. Все разложения числа N , упорядоченные в лексикографическом порядке по убыванию. Каждое разложение необходимо вывести в отдельной строке, разделив слагаемые разложения пробелом. Числа в каждой строке должны идти по убыванию.

| stdin | stdout |
|-------|--------------------------|
| 6 | 6 5 1 4 2 3 2 1 |

Вариант №10.

Задача А. (Таблица степеней по модулю N) Напишите программу, которая выводит таблицу остатков от деления чисел a^k на N . В k -й строке в столбце с номером a должен стоять остаток от деления числа a^k на число N , $1 \leq a \leq (N-1)$, $0 \leq k \leq (N-1)$. Считайте, что строки нумеруются с 0, а столбцы с 1.

Формат ввода. Натуральное число N , $2 \leq N \leq 100$.

Формат вывода. N строк. В каждой строке $(N-1)$ чисел, разделённых пробелами.

| stdin | stdout | | | |
|-------|--------|---|---|---|
| 5 | 1 | 1 | 1 | 1 |
| | 1 | 2 | 3 | 4 |
| | 1 | 4 | 4 | 1 |
| | 1 | 3 | 2 | 4 |
| | 1 | 1 | 1 | 1 |

Задача В. (Пересечение множеств) Напишите программу, которая находит пересечение двух введённых множеств натуральных чисел. Для хранения чисел используйте динамически выделенную память. Используйте, по возможности, указатели.

Формат ввода. Две строки, содержащие описание двух множеств. В начале каждой строки указан размер множества — число меньше 10000. Затем в каждой строке приведён список чисел множества (через пробел) в порядке возрастания. Все числа меньше 2^{31} . В каждой строке возможны повторения.

Формат вывода. Одна строка, содержащая числа (без повторений), которые содержатся и в первой, и во второй строке, перечисленные в порядке возрастания. Если пересечение пусто, то выведите слово `Empty`.

Задача С. (Сумма трёх квадратов) Напишите программу, которая определяет, можно ли данное число N представить в виде суммы трёх квадратов целых чисел.

Формат ввода. Натуральное число N , $0 < N \leq 20\,000\,000$.

Формат вывода. Выведите в одной строке три полных квадрата через пробел, если N можно разложить на сумму квадратов. Иначе, выведите число -1.

| stdin | stdout |
|-------|---------|
| 125 | 0 4 121 |
| 7 | -1 |

Задача D. (Генерация хороших слов) Напишите программу, выводит бинарные слова длины N , в которых ровно K единиц и нет двух подряд идущих единиц. Объявите глобальный массив `char word[30]` и реализуйте рекурсивную функцию $g(n, k, i)$, которая в “подмассиве” `word[i], word[i+1], ..., word[i+n]` перебирает различные варианты хороших слов длины n с k единицами.

Формат ввода. Два натуральных числа N и K : $0 \leq N$, $K < 25$, $K \leq (N+1)/2$.

Формат вывода. Все хорошие слова, упорядоченные по убыванию. Каждое слово выведите в отдельной строке.

| stdin | stdout |
|-------|--------|
| 4 2 | 1010 |
| | 1001 |
| | 0101 |