

**ЗАДАЧИ ДЛЯ ОТБОРОЧНОГО ЭТАПА ШКОЛЬНОЙ ОЛИМПИАДЫ
ПО ИНФОРМАТИКЕ ГАЗПРОМ**

Название: Сейфы

Сложность: 5%

Область знаний: логика

Темы:

- умение написать короткую программу на предложенном языке программирования
- знание основных конструкций языка программирования

Описание: Объем параллелепипеда вычисляется по формуле $V=abc$.

Где a -длина, b -ширина, c - высота.

Задача: У банкира Джона есть два сейфа, Джон не знает объемы своих хранилищ. Банкир имеет мешок денег, Джон не знает, влезут ли все его деньги в сейфы. Но проблема в том, что он не знает объемов сейфов, однако имеет их параметры a b c . Помогите Джону определить влезут ли все его деньги в сейфы.

Во входной строке дано целое число v – объем мешка.

И шесть параметров $a1, b1, c1, a2, b2, c2$.

Варианты ответа: 1 – «YES» 2 – «NO»

Решение: Поскольку мы имеем объем мешка и все параметры сейфов, то мы можем вычислить объемы сейфов и сложить их. Далее сравниваем суммарный объем сейфов с объемом мешка. Если объем мешка меньше либо равно суммарного объема сейфов, то выводим YES, в противном случае NO.

Название: Сортировка времени

Область знаний: логика

Сложность: 10%

Темы:

- знание основных понятий и законов математической логики
- знание основных конструкций языка программирования
- умение строить логические схемы и таблицы истинности

Описание: Требуется сортировку времени, заданного в часах, минутах и секундах. $X=3600*H + 60*M+S$, где x - общее время, h - количество часов, m - количество минут, s - количество секунд.

Задача: Джош очень плохо ориентируется в пространстве и времени и его секретарь всегда составляет ему четкое расписания. Но в этот день она заболела, а на его столе оказалось три листочка со временем проведения мероприятий. Помогите ему определить какое время будет раньше, а какое позже. Приведите время в порядке возрастания. Самое ближайшее время приведите к виду часы, минуты, секунды.

Во входной строке дано девять целых чисел $a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3$.

Соответственно 1- часы, 2 - минуты, 3 - секунды.

Выходная строка должна содержать три целых числа в порядке возрастания времени и три числа обозначающих время: часы, минуты, секунды.

Решение: Поскольку мы имеем часы, минуты и секунды, необходимо все перевести в секунды, а дальше сравнить их величины. Наибольшие с меньшими рассортировать в порядке возрастания времени. Далее необходимо взять наименьшее и совершить обратное преобразование в часы минуты и секунды посредством целочисленного деления и вычислением остатка.

Название: Листок в клетку

Область знаний: логика

Сложность: 15%

Темы:

- умение представлять и считывать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы)
- знание основных конструкций языка программирования

Описание: Дан листок в клетку, некоторые клетки закрашены. Вычислить по координатам чистая или закрашенная клетка.

Задача: У Алека есть вырванный тетрадный листок, он раскрасил квадрат 8×8 клеток так, что первая в первом ряду белая, вторая черная, и так они чередуются слева направо, и при передвижении вверх на следующий ряд предыдущая первая и настоящая первая клетка отличаются по цвету. Друг Алека говорит координаты клетки, и мальчик должен сказать белая она или черная. Помогите Алеку, напишите программу, которая при получении координат в виде чисел **A** и **B** будет говорить, какая это клетка.

Во входной строке дано целые числа **A** и **B** – координаты клетки.

Выходная строка должна состоять из одного слова «**WHITE**» или «**BLACK**»:
Варианты ответа: 1 – «**WHITE**» 2 – «**BLACK**»

Решение: Поскольку данный рисунок - это таблица с чередующимися белыми и черными клетками, нам достаточно определить массив, который будет заполнен значениями 0 - белая и 1 - черная. Необходимо сравнить значение по адресу ячейки с 0 или 1, и вывести на экран **WHITE** если значение равно 0 и **BLACK** если равно 1.

Название: НЛО

Сложность: 8%

Область знаний: базовое понимание алгоритмов и логических схем

Темы:

- Умение анализировать структуру и результат исполнения алгоритма
- Знание основных конструкций языка программирования
- Умение анализировать программу, содержащую ветвление

Описание: Необходимо проверить возможность уместения внутри объекта двух других без возможности их пересечения. Диаметр круга равен $2 * r$.

Задача: Многие утверждают, что круги на полях дело инопланетян, которые хотят что-то подсказать нашей расе. Так по словам одного уфолога на одном из полей в штате Огайо радиусом r_1 были обнаружены следы НЛО в виде кругов радиусом r_2 и r_3 . Он утверждает, что они находились внутри поля, не пересекаясь (возможно соприкасаясь по контуру). Необходимо проверить, действительно ли уфолог говорил правду, или пытался оправдать свои исследования.

Во входной строке даны три целых числа r_1 , r_2 и r_3 – радиусы полей.

Выходная строка должна состоять из одного слова «TRUE» или «FALSE».

Решение: Поскольку самым худшим вариантом исполнения условий будет возможность размещения кругов по диаметру, нам достаточно посчитать диаметр исходного поля, и сравнить с диаметром суммы радиусов двух кругов. Если диаметр поля больше диаметра суммы, то строка должна содержать слово «TRUE», в противном случае «FALSE»:

1 – «TRUE»

2 – «FALSE»

Название: Двоичное число.

Сложность: 8%

Область знаний: базовое понимание алгоритмов и логических схем

Темы:

- знание основных понятий и законов математической логики
- знание основных конструкций языка программирования
- умение строить логические схемы и таблицы истинности

Описание: Логическая операция "XOR" – “исключающее ИЛИ”. Например:
при 0 XOR 0 - будет 0,
при 0 XOR 1 - будет 1,
при 1 XOR 0 – будет 1,
при 1 XOR 1 - будет 0.

Необходимо применить операцию относительно трех чисел в двоичной системе счисления.

Задача: Юный программист Алан пытается решить задачу: ему даны три числа **a b c**, записанные в двоичной системе счисления и необходимо применить к ним логическую операцию «исключающее или». К сожалению Алан не знает как работает эта операция, помогите ему решить задание.

Во входной строке даны три целых двоичных числа **a,b,c**.
Выходная строка должна состоять из одного двоичного числа.

Решение: Поскольку мы имеем три двоичных числа, необходимо применить к ним данную операцию попарно, сначала первое со вторым, потом итоговое с третьим. При этом необходимо разделить данные числа поразрядно и сверять разряды попарно. Таким образом в итоге мы должны получить одно двоичное число.

Название: Кузя на прогулке

Область знаний: базовое понимание алгоритмов и логических схем

Сложность: 18%

Темы:

- Умение анализировать структуру и результат исполнения алгоритма.
- Знание основных конструкций языка программирования
- Умение анализировать программу, содержащую циклы и ветвление
- Умение представить и считать данные в разных типах информационных моделей.

Описание: Подается количество дорог и таблица, составленная по принципу таблицы умножения, где строки и столбца пронумерованы номерами пунктов, на пересечении строки и столбца стоит либо 1, либо 0. Единица стоит на пересечении номеров пунктов, между которыми есть дорога.

Задача: Кузя очень долго сидел дома и решил прогуляться, но пока он выбирал куда пойти он понял, что не может сосчитать количество дорог, по которым он может направляться из пункта в пункт, но не все пункты соединены между собой. Требуется написать программу, помогающую Кузе сосчитать количество дорог, по которым он может прогуляться.

Выходная строка должна содержать количество дорог.

Решение: Чтобы сосчитать количество дорог, необходимо сосчитать количество единиц в таблице, передаваемой построчно и посимвольно. После того как единицы посчитаны нужно поделить их количество пополам и получим ответ.

Название: Ошибка

Сложность: 4%

Область знаний: циклические алгоритмы

Темы:

- Умение обнаружить ошибки во фрагменте программного кода
- Умение создать алгоритм с ограниченным набором команд
- Умение анализировать программу, содержащую циклы

Задача:

Выведите n знаков после десятичной точки числа π . В данной задаче будем считать, что число π в точности равно 3,1415926535897932384626433832795

Входная строка содержит целое число n ($0 \leq n \leq 25$).

Выходная строка содержит n знаков после десятичной точки числа π .

Решение:

Создаем символьный массив, содержащий в себе точное число π , указанное в задаче. Запускаем цикл до n и выводим на экран посимвольно.

Название: Умножение

Сложность: 7%

Область знаний: циклические алгоритмы

Темы:

- анализ алгоритма, содержащего цикл и ветвление
- знание основных конструкций языка программирования

Описание: Привычное для нас умножение – это всего лишь многократное сложение.

Задача: Племя Чевилуар содержит 3 человека, а должно содержать M человек для совершения темного ритуала, но у них недостаточно людей для этого. Тогда глава племени Чевилуар решил объединиться с главами других N племен по R человек. Глава Чевилуар не умеет умножать. Напишите программу, которая вычислит возможен ли ритуал после объединения племен, **не** используя операции умножения.

Входная строка содержит M, N, R .

Варианты ответа: 1 – «YES» 2 – «NO»

Решение: Поскольку умножение - это многократное сложение, то нам необходимо сложить B с самим собой C раз в цикле. Нельзя забывать об оставшихся трех членах племени Чевилуар. Далее сравнить сумму людей с условием N , таким образом проверим условие возможности совершения ритуала.

Название: Двойные агенты

Сложность: 17%

Область знаний: выборка из базы данных

Темы:

- Умение строить логические схемы и таблицы истинности
- Знания о системах счисления
- Умение строить логические выражения
- Анализ алгоритма, содержащего цикл
- Поиск информации в базах данных

Описание: ID агента имеет 8 цифр, например, 14794128

Задача: Анонимный информатор сообщил, что в секретном отделе “Z” присутствует внутренняя, вражеская агентурная сеть. Информатор не смог запомнить все базу номеров двойных агентов, однако он вычислил определенные закономерности для идентификации двойных агентов:

*Первая и последняя цифры – четные.

*Сумма всех цифр больше 50.

*Вторая или третья цифры это 9

*Двоичная запись суммы цифр имеет равное количество нулей и единиц.

Напишите программу, которая поможет вычислить всех двойных агентов в секретном отделе “Z”.

Входная строка содержит число N, кол-во сотрудников, далее массив идентификаторов агентов.

Выходная строка содержит только список предполагаемых двойных агентов по закономерностям информатора.

Решение:

Запускаем цикл N раз и считываем число. Создаем массив на 8 символов.

Заполняем его текущим ID. Используйте операцию деление нацело для получения одной цифры нужной позиции. Вычислить сумму массива (sum).

Вычисляем двоичную запись и считаем количество нулей и единиц, добавим переменную F-флаг, присвоить F=истина, если кол-во нулей и единиц равны.

Далее создаем сложное условие:

Если $((A[1] \bmod 2) \neq 0)$ И $(A[7] \bmod 2) \neq 0)$ и $(sum > 50)$ И $(A[2]=9)$ ИЛИ $(A[3]=9)$ и $(F=истина)$, то выводим весь номер, переходим к следующему.

Название: Шифр-перевертыш

Сложность 2%:

Область знаний: Область знаний: простейшие программы, кодирование информации

Темы:

- Знание основные конструкций языка программирования
- Умение написать короткую программу на предложенном языке программирования
- Анализ программы, содержащей цикл.

Задача:

Агент «Большой Джо» решил передать агенту «Темный Ястреб» тайное послание он решил передавать свои сообщения в обратном порядке, т.е «АВС» будет передано как «ВСА».

Помогите «Большому Джо» написать программу для шифрования своих сообщений.

Входная строка содержит исходное сообщение.

Выходная – зашифрованное сообщение

Пример:

Big Joe —> eoJ giB

Решение:

Для решения задачи достаточно запустить цикл «с конца» и выводить последовательное все символы от N до нуля, при $N > 0$.

Название: Шарик

Сложность: 12%

Область знаний: Область знаний: простейшие программы, кодирование информации

Темы:

- Умение представлять и считывать данные в разных типах данных информационных моделей
- Умение кодировать и декодировать информацию
- Умение анализировать программу, использующую функции

Задача:

Шулер показывает следующий трюк. Он имеет три одинаковых наперстка. Под первый (левый) он кладет маленький шарик. Затем он очень быстро выполняет ряд перемещений наперстков, каждое из которых – это одно из трех перемещений - А, В, С:

А - обменять местами левый и центральный наперстки,

В - обменять местами правый и центральный наперстки,

С - обменять местами левый и правый наперстки.

Напишите программу, которая определит под какими из наперстков НЕ окажется шарика после всех перемещений.

Входная строка: записана строка длиной не более 50 символов из множества {А, В, С}

Выходная строка: номера наперстков, через запятую под которыми нет шарика.

Решение:

Создадим функции обмена А, В, С. Добавим массив из трех элементов и «положим шарик» в левый элемент, получим {1,0,0}.

Посимвольно считаем строку через цикл и установим серию условий:

Если символ равен «А», то осуществим функцию замены «левый и центральный наперстки»

Если символ равен «В», то осуществим функцию замены «правый и центральный наперстки»

Если символ равен «С», то осуществим функцию замены «левый и правый наперстки»

Вычисляем и выводим индексы элементов, равных нулю.

Название: Шифровка

Сложность: 18%

Область знаний: простейшие программы, кодирование информации

Темы:

- Умение кодировать и декодировать информацию
- Умение анализировать структуру и результат исполнения алгоритма.
- Поиск информации

Описание: Шифр простой замены заключается в том, что каждая буква текста заменяется на соответствующую букву шифротекста по индексу.

Задача: агент Блонд получил шифровку S, алфавит состоит всего из 6 символов необходимо дешифровать текст по ключевой таблице:

A=E

B=C

D=F

E=A

C=B

F=D

Помогите агенту Блонду написать программу для дешифровки текста.

Гарантируется, что все буквы в шифровке состоят из табличного алфавита.

Пример:

Шифротекст: CEFEC

Дешифровка: BADAB

Входная строка содержит сообщение – шифротекст длиной N символов

Выходная строка содержит дешифровку

Решение:

Создадим три массива: первый массив заполняем символами шифротекста.

Второй заполняем алфавитом "ABDEC F"

Третий заполняем шифроалфавитом "ECFABD"

Запустим цикл с условием: Если буква шифротекста совпадает с буквой шифроалфавита ($\text{шифротекст}[i]=\text{шифроалфавит}[i]$), то выводим букву исходного алфавита по текущему индексу i ($\text{алфавит}[i]$).