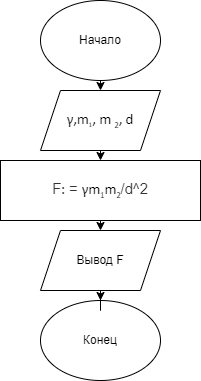
Лабораторная работа №1. Составление алгоритмов

Вариант 2

Задача 1.

1.1-2 Даны два тела с массами m1 и m2 , расположенные на расстоянии d друг от друга. Найти силу их взаимного притяжения. (Гравитационная постоянная γ ≈ 6,67 × 10−11 м 3 кг·с 2 .)

**Алг** Нахождение силы взаимного притяжения

**дано** γ ≈ 6,67 × 10−11

**надо** F

**нач**

**цел** m1, m2, d

**вещ** γ, F

**ввод** γ, m1, m2, d

**нц**

F: = γm1m2/d^2

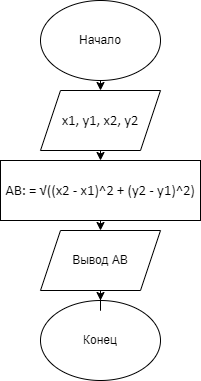
**кц**

**вывод** F

**кон**

Задача 2.

1.2-2 Написать программу, вычисляющую расстояние между точками (x1 , y1 ) и (x2 , y2 ).

**Алг** Нахождение расстояния между точками

**дано**

**надо** AB

**нач**

**цел**

**вещ** x1, y1, x2, y2, AB

**ввод** x1, y1, x2, y2

**нц**

AB: = √((x2 - x1)^2 + (y2 - y1)^2)

**кц**

**вывод** AB

**кон**

Задача 3.

1.4. Операции ввода и вывода

1.4-2 Дан угол в градусах. Написать программу, переводящую его в деления угломера (единица измерения углов, принятая в артиллерии). 6000 делений угломера составляют 360°. Последние две цифры обычно отделяются дефисом. Например, 705 делений угломера записываются как «7-05». Ответ вывести в виде: «Угол d° равен a-b.». Вместо буквенных обозначений должны стоять конкретные целые числа. Перед запросом ввода с клавиатуры выводить подсказку. (Код символа градуса в Unicode — 00B0.)

**Алг**

**дано**

**надо**

**нач**

**цел**

**вещ**

**ввод**

**нц для**

**кц**

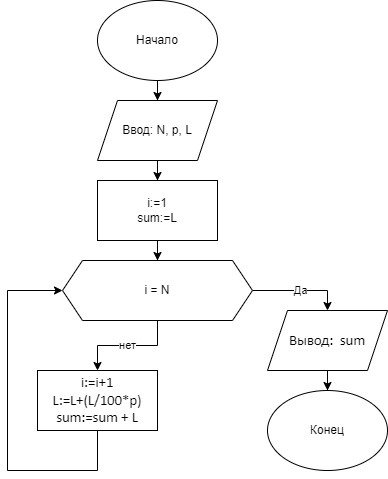
**вывод**

**кон**

Задача 4.

3.1. Цикл с параметром

3.1-2 Начав тренировки спортсмен в первый день пробежал L км. Каждый следующий день он увеличивал пробег на p % от пробега предыдущего дня. Написать программу, определяющую суммарный пробег за N дней.

**Алг** Суммарный пробег за N дней

**дано** i:=1

sum:=L

**надо** sum

**нач**

**цел**

**вещ** N, p, L

**ввод**  N, p, L

**нц**

i:=i+1

L:=L+(L/100\*p)

sum:=sum + L

**кц**

**вывод** sum

**кон**

Задача 5.

3.2. Цикл с условием

3.2-2 В 50-х годах XX века был предложен (к сожалению, неэффективный) метод Фибоначчи получения последовательности псевдослучайных чисел. Последовательность задается следующими формулами: X0 = 1; X1 = 1; Xk ≡ Xk−1 + Xk−2 (mod m), где m — некоторое натуральное число. Эта последовательность повторяется с определенным периодом. Например, для m = 3 последовательность имеет вид: 1, 1, 2, 0, 2, 2, 1, 0, 1, 1, . . . , то есть период равен 8. Написать программу, определяющую для заданного m длину периода последовательности.

X0 = 1;

X1 = 1;

Xk ≡ (Xk−1 + Xk−2) % m

**Алг**

**дано**

**надо**

**нач**

**цел**

**вещ**

**ввод**

**нц для**

**кц**

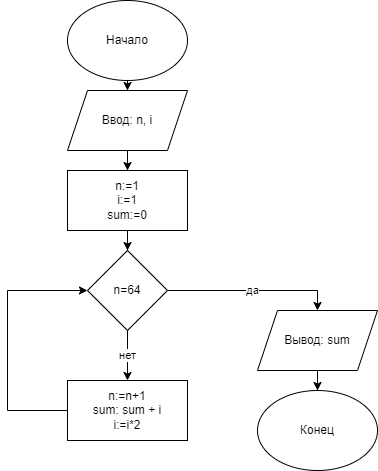
**вывод**

**кон**

Задача 6.

На алгоритмическом языке составьте алгоритм решения данной задачи и нарисуйте блок-схему алгоритма ее решения: Когда создатель шахмат, древнеиндийский мудрец и математик Сисса бен Дахир, показал своё изобретение Правителю страны, тому так понравилась игра, что он позволил изобретателю право самому выбрать награду. Мудрец попросил у Повелителя за первую клетку шахматной доски заплатить ему одно зерно пшеницы (по другой версии — риса), за второе — два, за третье —четыре и т. д., удваивая количество зерен на каждой следующей клетке. Правитель, не разбирающийся в математике, быстро согласился, даже несколько обидевшись на столь невысокую оценку изобретения, и приказал казначею подсчитать и выдать изобретателю нужное количество зерна. Однако, когда неделю спустя казначей всё ещё не смог подсчитать, сколько нужно зёрен, правитель спросил, в чём причина такой задержки. Казначей показал ему расчёты и сказал, что расплатиться невозможно, поскольку количество зерна превышает весь урожай пшеницы, собранный за всю историю человечества.

**Алг** Зерна на шахматной доске

**дано** n:=1

i:=1

sum:=0

**надо** sum

**нач**

**цел** n, i

**вещ**

**ввод** n, i

**нц** делать до n=64

n:=n+1

sum: sum + i

i:=i\*2

**кц**

**вывод** sum

**кон**

Для выполнения блок-схем воспользоваться веб-сайтом app.diagrams.net

или программным обеспечением Dia. Ссылка тут.

Внимание! Все ваши работы должны быть размещены в вашем

удаленном git-репозитории. Клиент Github. Для всех задач, выполненных в

ходе лабораторной работы рассчитать временную сложность составленных

алгоритмов (О-нотация).

