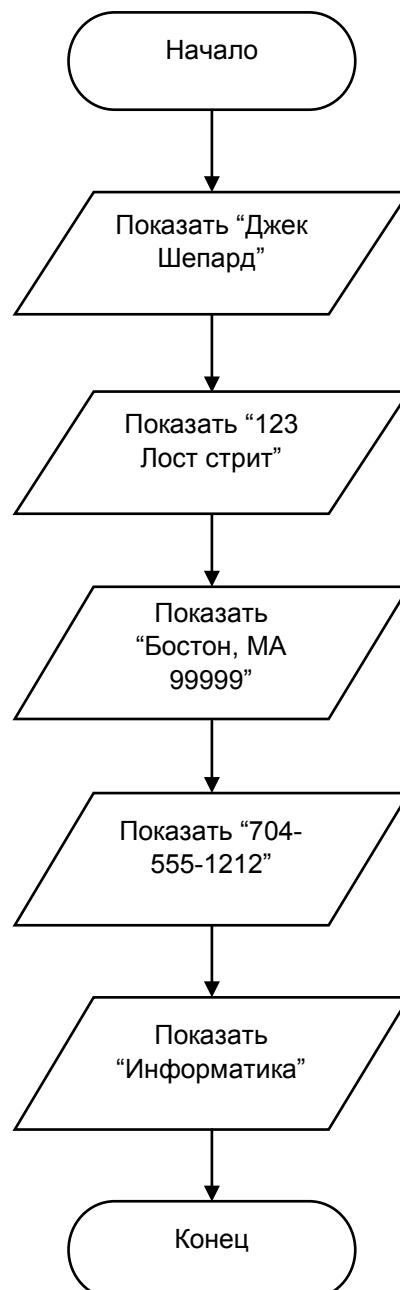


Упражнения по программированию главы 2

coding: utf-8

Упражнение 2.1. Персональные данные

```
print("Джек Шепард")  
print("123 Лост стрит")  
print("Бостон, МА 99999")  
print("704-555-1212")  
print("Информатика")
```



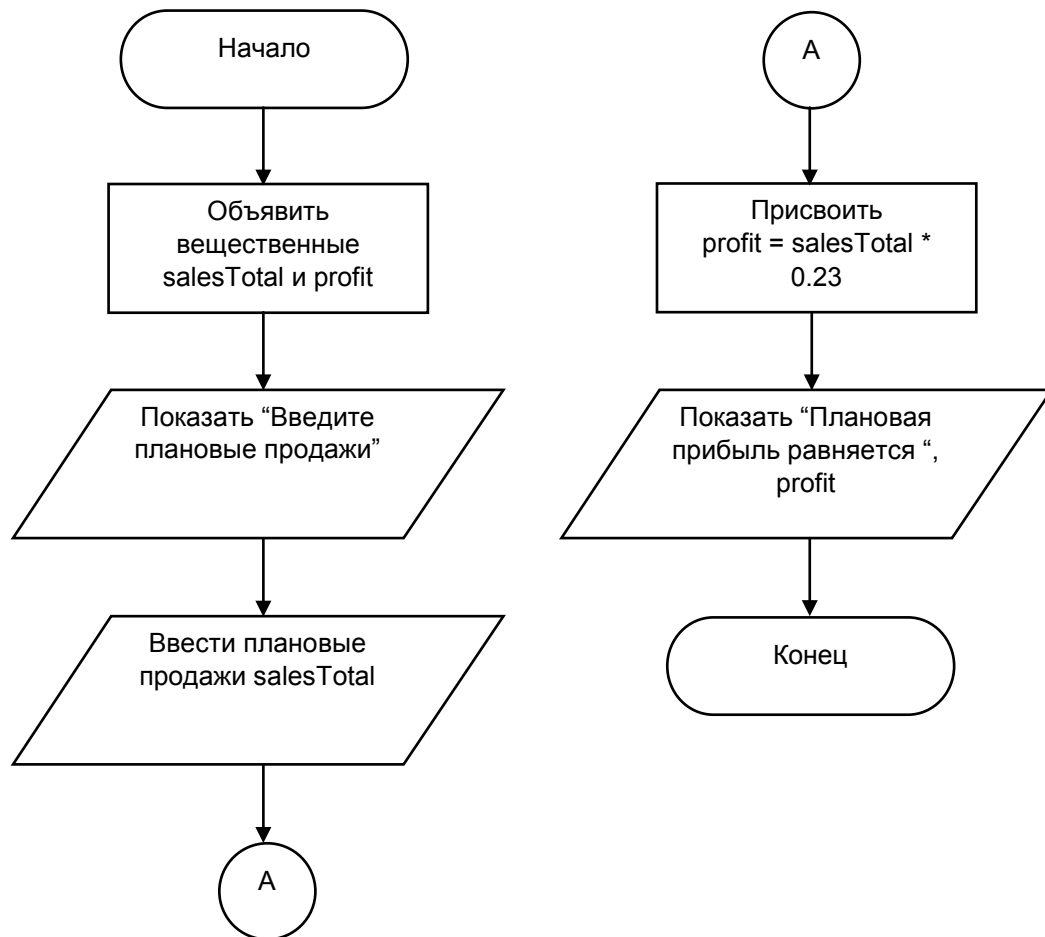
Упражнение 2.2. Прогноз продаж

```
# Переменные, содержащие общую сумму продаж и прибыль
salesTotal = 0.0
profit = 0.0

# Получить сумму плановой прибыли.
salesTotal = float(input("Введите плановые продажи: "))

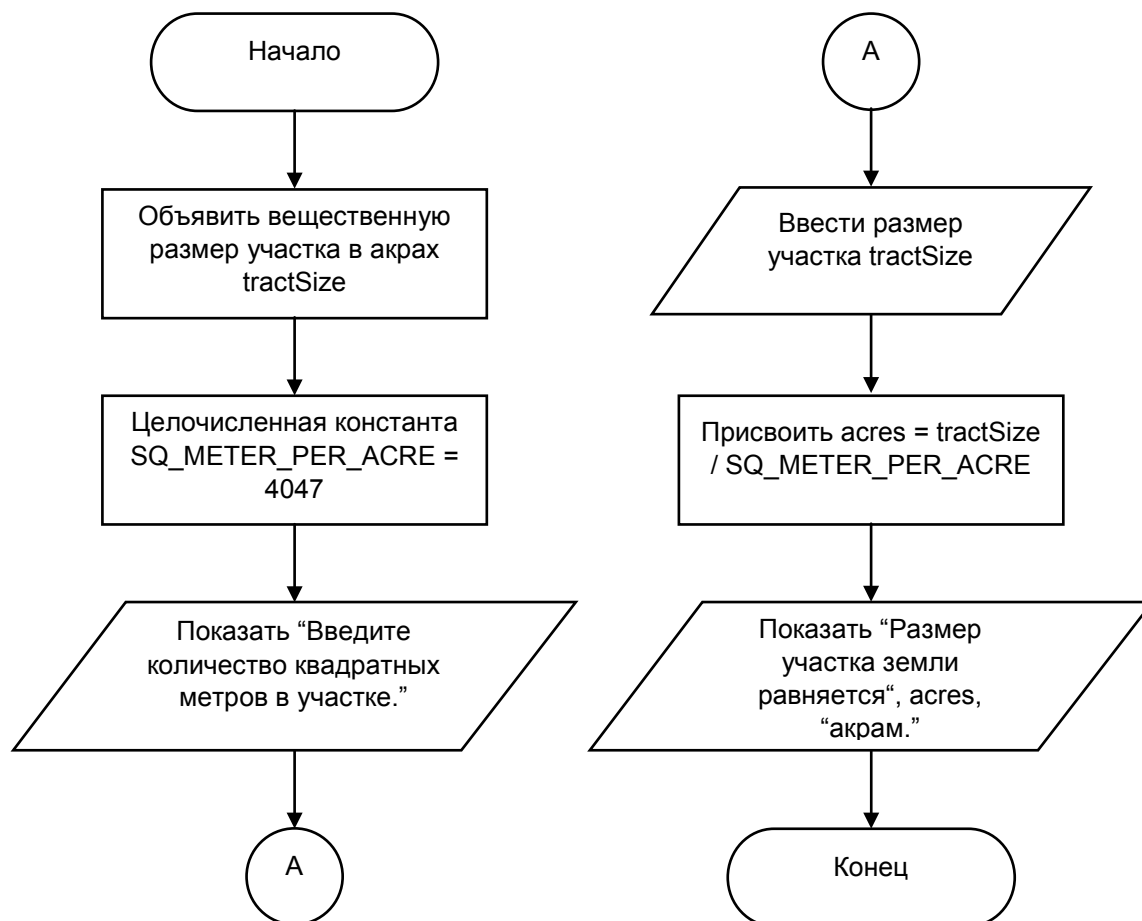
# Вычислить плановую прибыль.
profit = salesTotal * 0.23

# Напечатать плановую прибыль.
print ("Плановая прибыль составит ", format(profit, '.2f'))
```



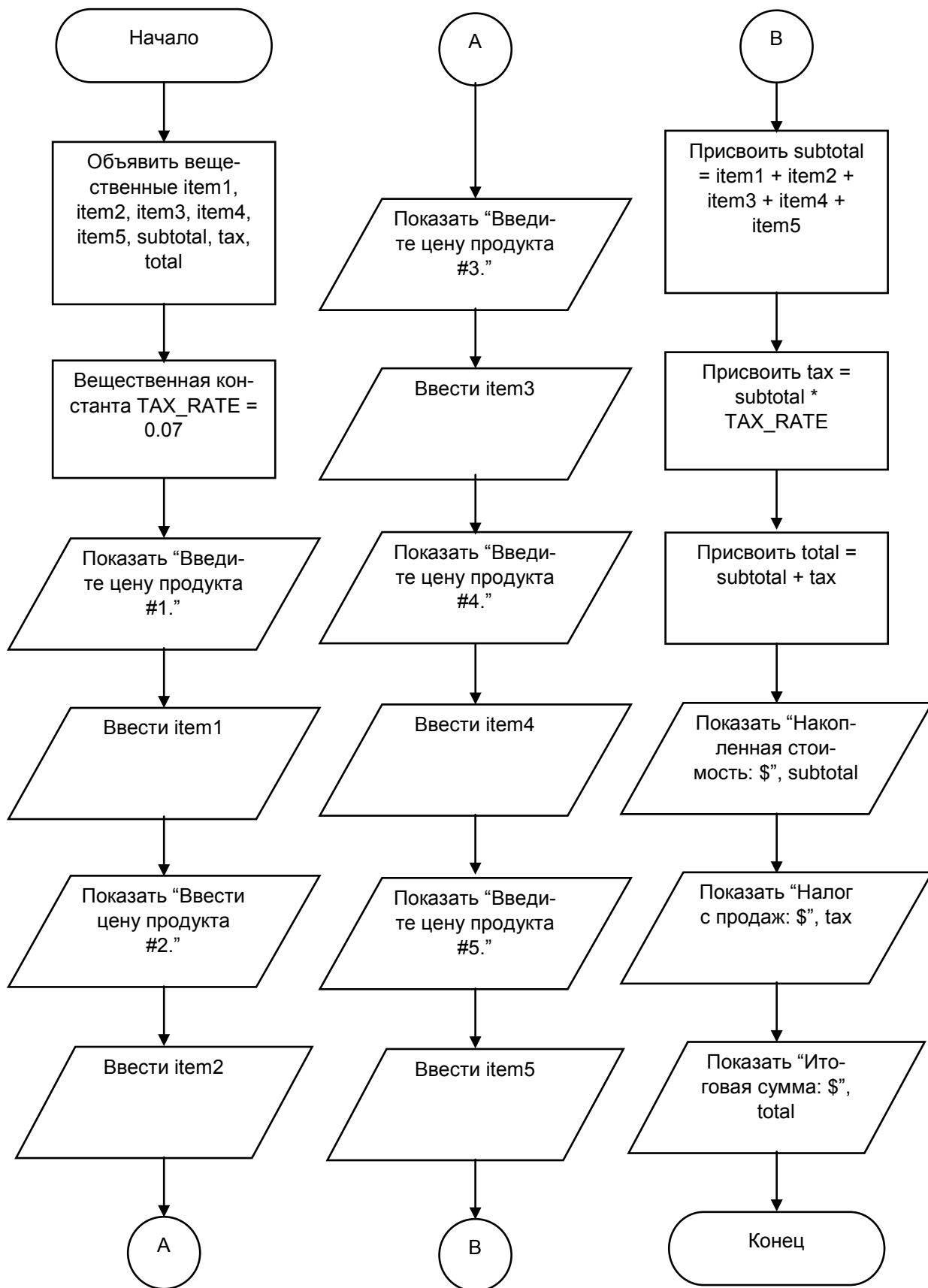
Упражнение 2.3. Расчет площади земельного участка

```
# Переменные, содержащие размер земельного участка и количество акров.  
tractSize = 0.0  
acres = 0.0  
  
# Константа количества квадратных метров в земельном участке в акрах.  
SQ_FEET_PER_ACRE = 4047  
  
# Получить квадратные метры земельного участка.  
tractSize = input("Ввести количество квадратных метров в земельном участке: ")  
  
# Вычислить площадь в акрах.  
acres = float(tractSize) / SQ_FEET_PER_ACRE  
  
# Напечатать количество акров.  
print ("Размер земельного участка составит", format(acres, '.2f'), "акров.")
```



Упражнение 2.4. Общий объем продаж

```
# Переменные, содержащие цену каждого товара, накопленную стоимость,  
# и итоговую сумму.  
item1 = 0.0  
item2 = 0.0  
item3 = 0.0  
item4 = 0.0  
item5 = 0.0  
subtotal = 0.0  
tax = 0.0  
total = 0.0  
  
# Константа для ставки налога с продаж.  
TAX_RATE = 0.07  
  
# Получить цену каждой товарной позиции.  
item1 = float(input("Введите цену позиции #1: "))  
item2 = float(input("Введите цену позиции #2: "))  
item3 = float(input("Введите цену позиции #3: "))  
item4 = float(input("Введите цену позиции #4: "))  
item5 = float(input("Введите цену позиции #5: "))  
  
# Вычислить накопленную стоимость.  
subtotal = item1 + item2 + item3 + item4 + item5  
  
# Вычислить налог с продаж.  
tax = subtotal * TAX_RATE  
  
# Вычислить итоговую сумму.  
total = subtotal + tax  
  
# Напечатать значения.  
print("Накопленная стоимость: ", format(subtotal, '.2f'))  
print("Налог с продаж: ", format(tax, '.2f'))  
print("Итоговая сумма: ", format(total, '.2f'))
```



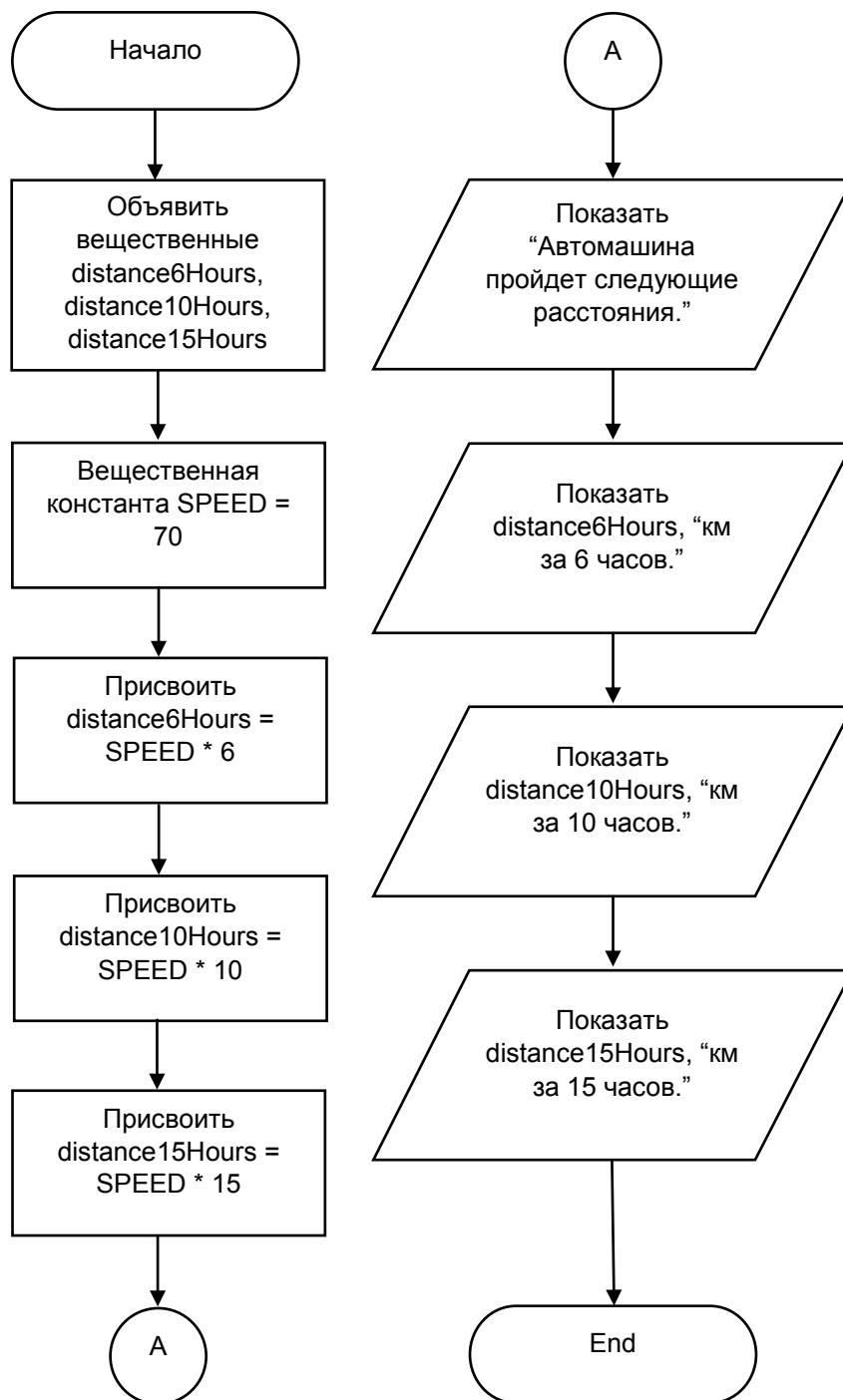
Упражнение 2.5. Пройденное расстояние

```
# Переменные, содержащие расстояния.
distance6Hours = 0.0
distance10Hours = 0.0
distance15Hours = 0.0

# Константа для скорости.
SPEED = 70

# Вычислить расстояние, которое автомобиль пройдет за
# 6, 10 и 15 часов.
distance6Hours = SPEED * 6
distance10Hours = SPEED * 10
distance15Hours = SPEED * 15

# Напечатать результаты.
print("Автомобиль пройдет следующие расстояния:")
print(distance6Hours, "километров за 6 часов.")
print(distance10Hours, "километров за 10 часов.")
print(distance15Hours, "километров за 15 часов.")
```



Упражнение 2.6. Налог с продаж

```
# Объявления переменных
purchase = 0.0
stateTax = 0.0
countyTax = 0.0
totalTax = 0.0
totalSale = 0.0

# Константы для ставок федерального и регионального налога с продаж
STATE_TAX_RATE = 0.05
COUNTY_TAX_RATE = 0.025

# Получить сумму покупки.
purchase = float(input("Введите сумму покупки: "))

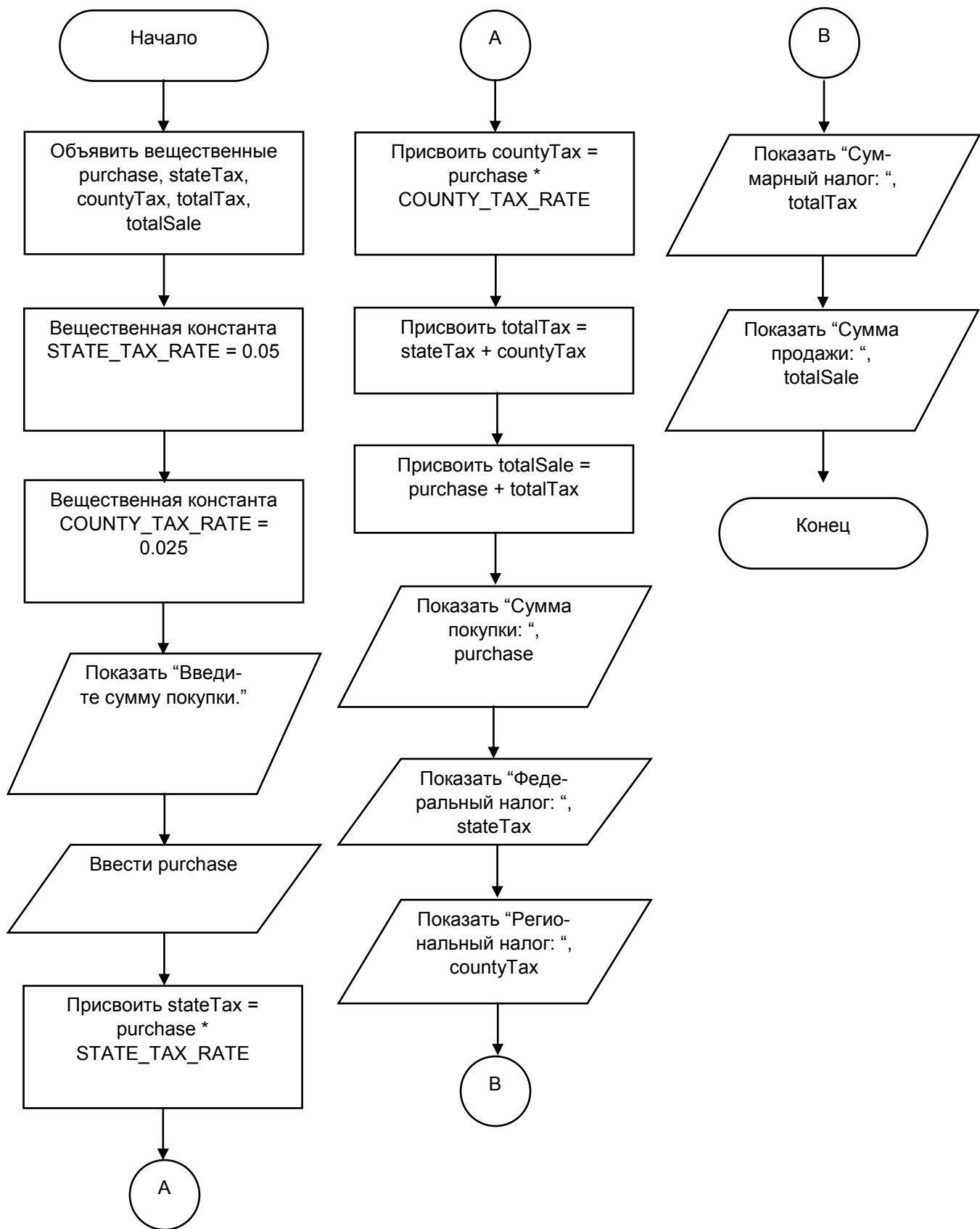
# Вычислить федеральный налог с продаж.
stateTax = purchase * STATE_TAX_RATE

# Вычислить региональный налог с продаж.
countyTax = purchase * COUNTY_TAX_RATE

# Вычислить суммарный налог.
totalTax = stateTax + countyTax

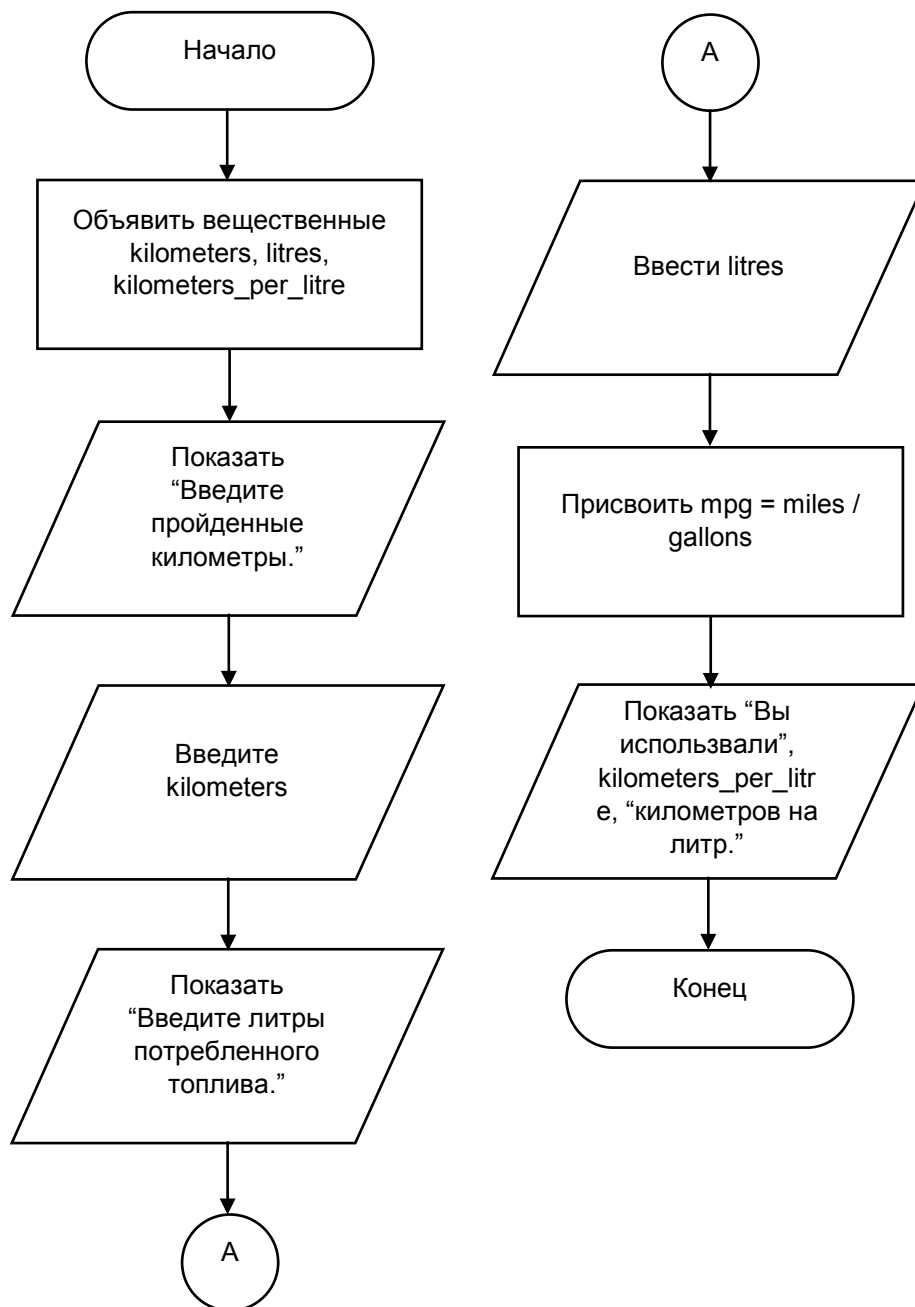
# Вычислить итоговую сумму продажи.
totalSale = purchase + totalTax

# Напечатать информацию о продаже.
print("Сумма покупки:", format(purchase, '.2f'))
print("Федеральный налог:", format(stateTax, '.2f'))
print("Региональный налог:", format(countyTax, '.2f'))
print("Суммарный налог:", format(totalTax, '.2f'))
print("Сумма покупки:", format(totalSale, '.2f'))
```

Упражнение 2.7. Расход бензина в расчете на километры пройденного пути

```
# Объявить переменные, содержащие пройденные километры,  
# расход топлива в литрах и километр в расчете на литр.  
kilometers = 0.0  
litres = 0.0  
kilometers_per_litre = 0.0 # miles-per-gallon  
  
# Получить пройденное расстояние.  
kilometers = float(input("Введите пройденные километры: "))  
  
# Получить объем потребленного топлива.  
litres = float(input("Введите литры потребленного топлива: "))  
  
# Вычислить километры в расчете на литр.  
kilometers_per_litre = kilometers / litres  
  
# Напечатать результаты.  
print("Вы использовали", format(kilometers_per_litre, '.2f'), "километров на литр.")
```



Упражнение 2.8. Чаевые, налог и общая сумма

```
# Объявить переменные для стоимости еды, размера чаевых, налога и итоговой суммы.
```

```
food = 0.0
```

```
tip = 0.0
```

```
tax = 0.0
```

```
total = 0.0
```

```
# Константы для ставки налога и ставки чаевых.
```

```
TAX_RATE = 0.07
```

```
TIP_RATE = 0.18
```

```
# Получить стоимость еды.
```

```
food = float(input("Введите стоимость еды: "))
```

```
# Вычислить чаевые.
```

```
tip = food * TIP_RATE
```

```
# Вычислить налог.
```

```
tax = food * TAX_RATE
```

```
# Вычислить итоговую сумму.
```

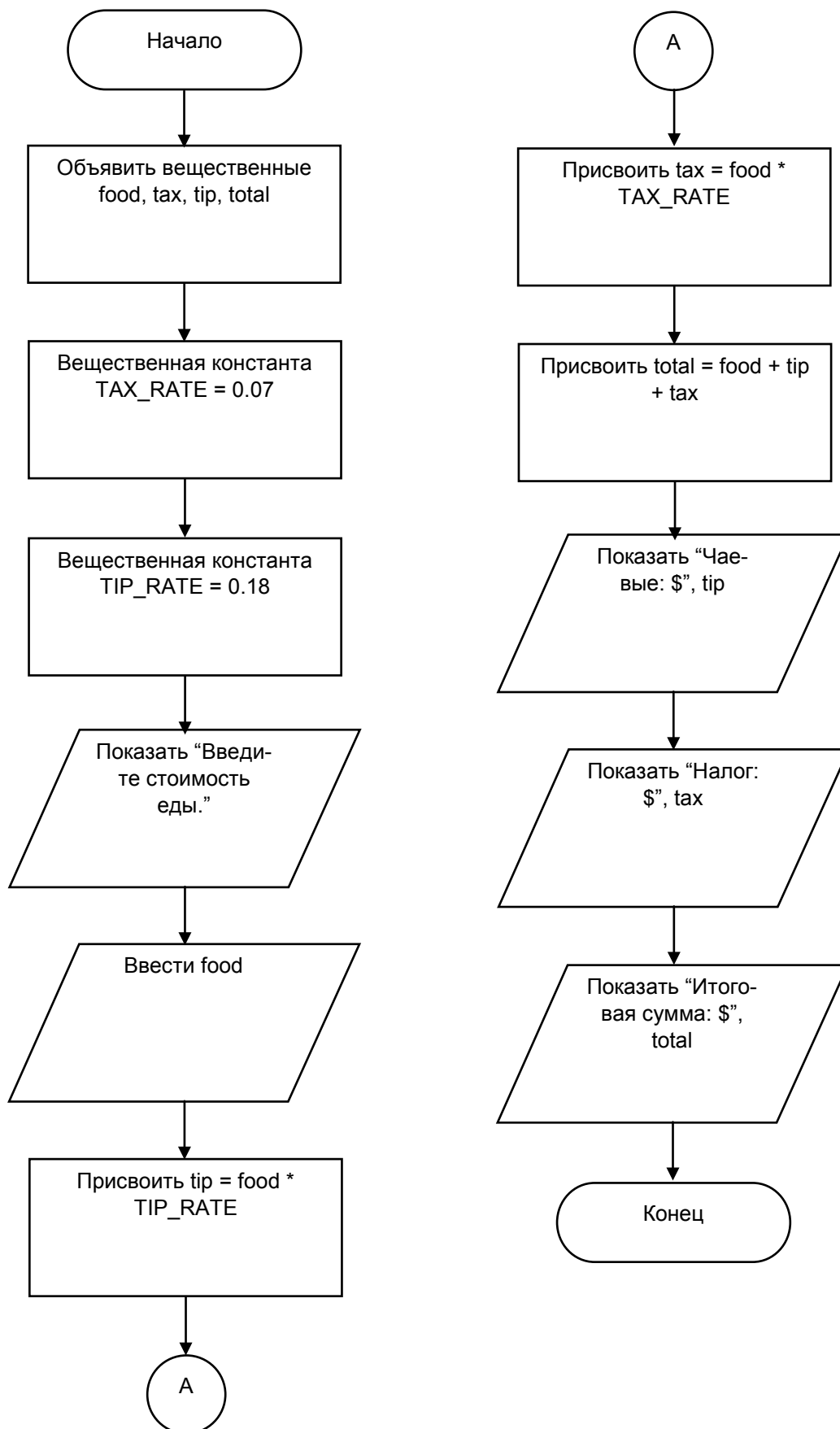
```
total = food + tip + tax
```

```
# Напечатать чаевые, налог и итоговую сумму.
```

```
print("Чаевые: $", format(tip, '.2f'))
```

```
print("Налог: $", format(tax, '.2f'))
```

```
print("Итоговая сумма: $", format(total, '.2f'))
```



Упражнение 2.09. Преобразователь температуры по шкале Цельсия в температуру по шкале Фаренгейта

```
# Объявить переменные, содержащие температуры.
```

```
celsius = 0.0
```

```
fahrenheit = 0.0
```

```
# Получить температуру по шкале Цельсия.
```

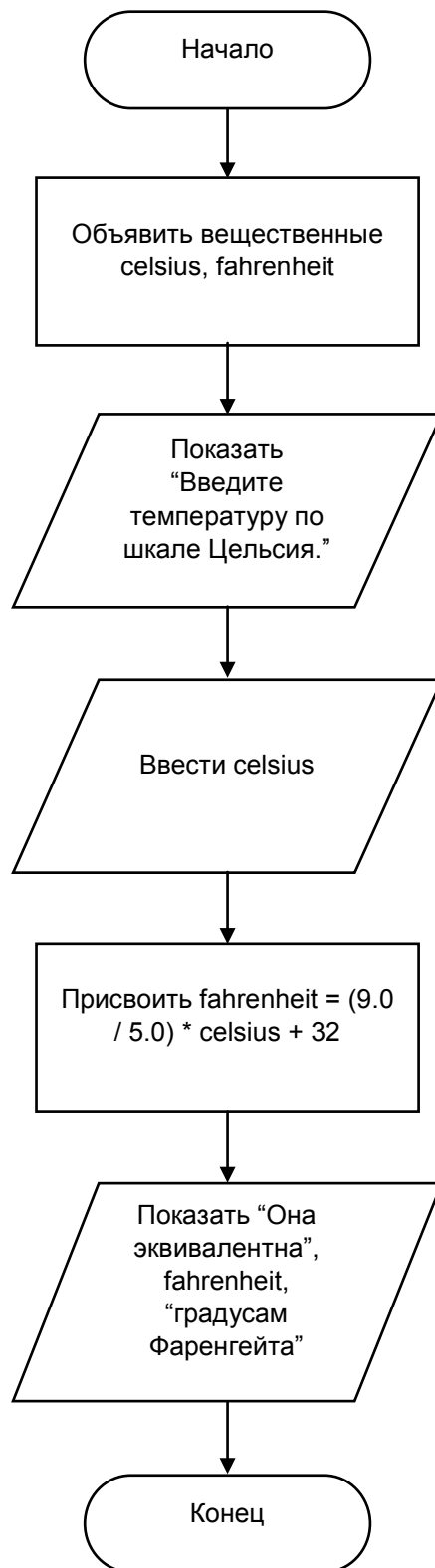
```
celsius = float(input("Введите температуру по шкале Цельсия: "))
```

```
# Вычислить эквивалент по шкале Фаренгейта.
```

```
fahrenheit = (9.0 / 5.0) * celsius + 32
```

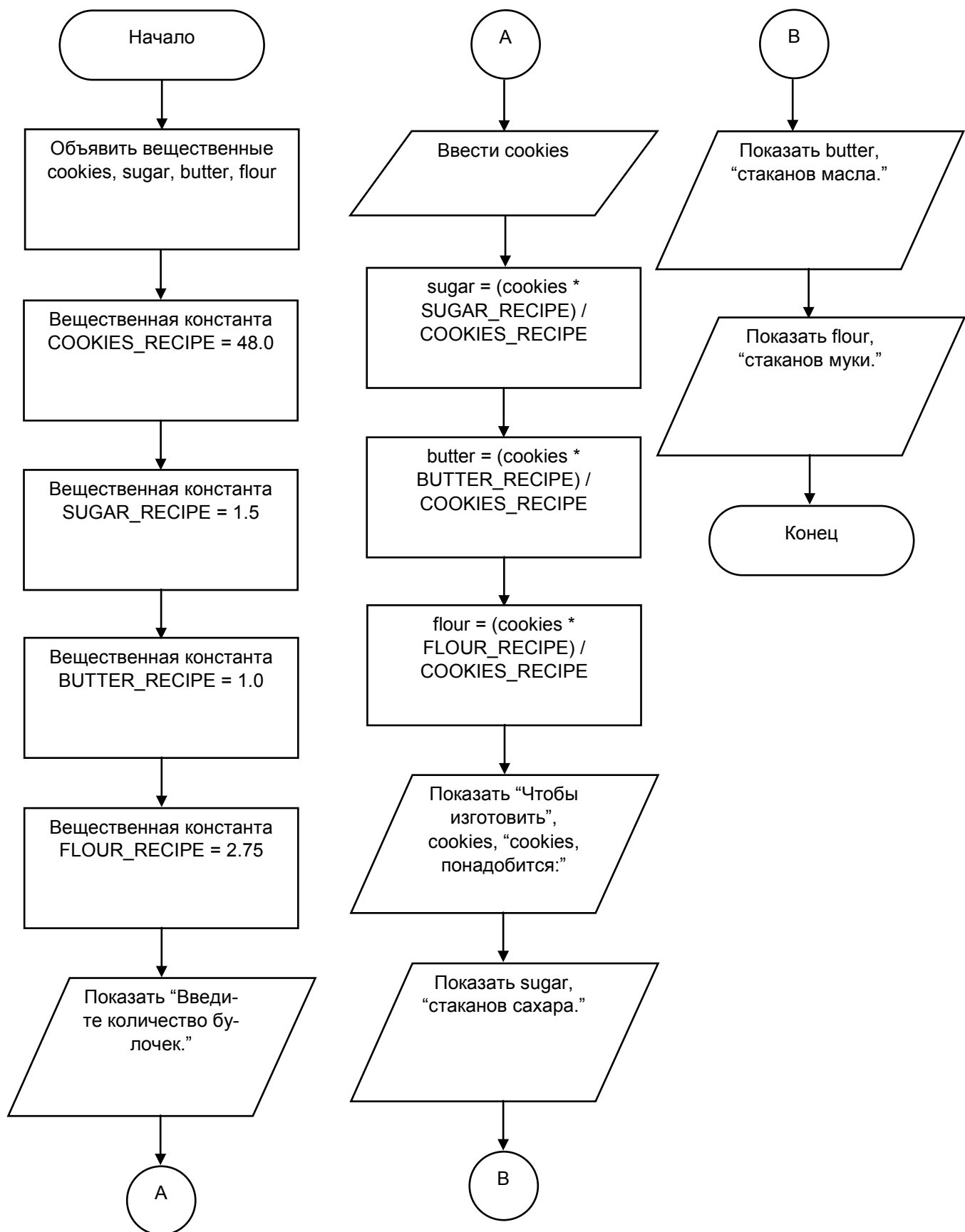
```
# Показать температуру по шкале Фаренгейта.
```

```
print("Она эквивалентна", format(fahrenheit, '.2f'), "градусам Фаренгейта.")
```



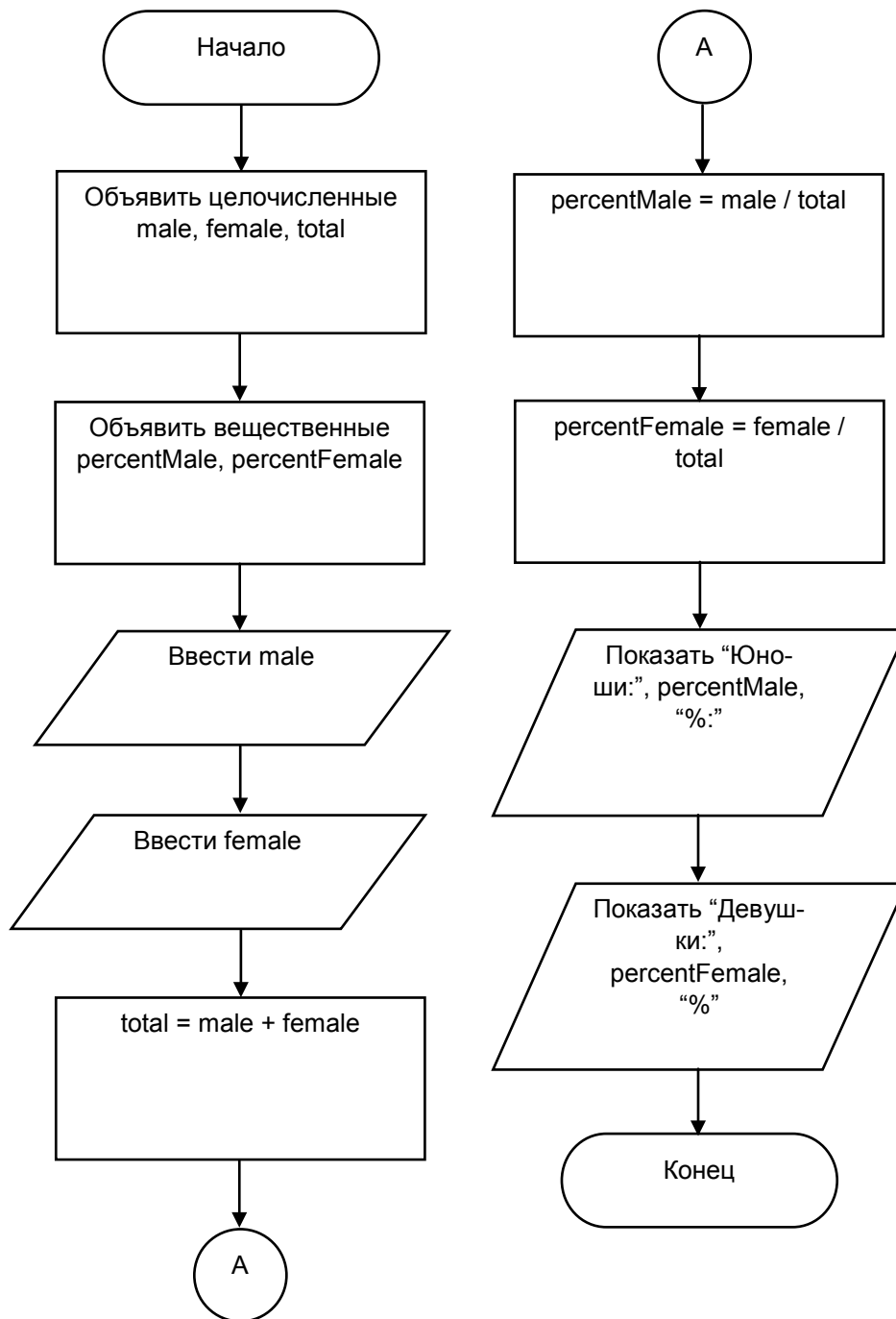
Упражнение 2.10. Регулятор ингредиентов

```
# Переменные, содержащие количество булочек,  
# стаканов сахара, масла и муки.  
cookies = 0.0  
sugar = 0.0  
butter = 0.0  
flour = 0.0  
  
# Константы для количества булочек,  
# стаканов сахара, масла и муки в исходном рецепте.  
COOKIES_RECIPE = 48.0  
SUGAR_RECIPE = 1.5  
BUTTER_RECIPE = 1.0  
FLOUR_RECIPE = 2.75  
  
# Получить количество булочек.  
cookies = float(input("Введите количество булочек: "))  
  
# Вычислить стаканы сахара, необходимые для изготовления булочек.  
sugar = (cookies * SUGAR_RECIPE) / COOKIES_RECIPE  
  
# Вычислить стаканы масла, необходимые для изготовления булочек.  
butter = (cookies * BUTTER_RECIPE) / COOKIES_RECIPE  
  
# Вычислить стаканы муки, необходимые для изготовления булочек.  
flour = (cookies * FLOUR_RECIPE) / COOKIES_RECIPE  
  
# Напечатать объем масла, сахара и муки, необходимых  
# для изготовления указанного количества булочек.  
print("Чтобы приготовить", cookies, "булочек, вам понадобятся:")  
print(format(sugar, '.2f'), "стаканов сахара")  
print(format(butter, '.2f'), "стаканов масла")  
print(format(flour, '.2f'), "стаканов муки")
```

Упражнение 2.11. Процент учащихся обоего пола

```
# Переменные для количества учащихся женского и мужского пола,  
# общего количества учащихся и процента учащихся  
# женского и мужского пола.  
male = 0  
female = 0  
total = 0  
percentMale = 0.0  
percentFemale = 0.0  
  
# Получить количество учащихся мужского пола.  
male = int(input("Введите количество учащихся мужского пола: "))  
  
# Получить количество учащихся женского пола.  
female = int(input("Введите количество учащихся женского пола: "))  
  
# Вычислить общее количество учащихся.  
total = male + female  
  
# Вычислить процент учащихся мужского пола.  
percentMale = male / total  
  
# Вычислить процент учащихся женского пола.  
percentFemale = female / total  
  
# Напечатать процент учащихся мужского пола.  
print("Юноши:", format(percentMale, '.2f'), "%")  
  
# Напечатать процент учащихся женского пола.  
print("Девушки:", format(percentFemale, '.2f'), "%")
```



Упражнение 2.12. Программа расчета купли-продажи акций

```
# Именованные константы
COMMISSION_RATE = 0.03
NUM_SHARES = 2000
PURCHASE_PRICE = 40.0
SELLING_PRICE = 42.75

# Переменные
amountPaidForStock = 0.0 # Сумма, выплаченная за акции
purchaseCommission = 0.0 # Комиссия, уплаченная за покупку акций
totalPaid = 0.0          # Итоговая уплаченная сумма
stockSoldFor = 0.0       # Сумма, за которую акции были проданы
sellingCommission = 0.0  # Комиссия, уплаченная за продажу акций
totalReceived = 0.0      # Итоговая полученная сумма
profitOrLoss = 0.0       # Сумма дохода или убытка

# Вычислить сумму, которую Джо уплатил за акции,
# не включая комиссию.
amountPaidForStock = NUM_SHARES * PURCHASE_PRICE

# Вычислить сумму комиссии, которую Джо уплатил брокеру
# при покупке акций.
purchaseCommission = COMMISSION_RATE * amountPaidForStock

# Вычислить общую сумму, которую Джо уплатил, т.е. сумму,
# которую он уплатил за акции плюс комиссию, уплаченную брокеру.
totalPaid = amountPaidForStock + purchaseCommission

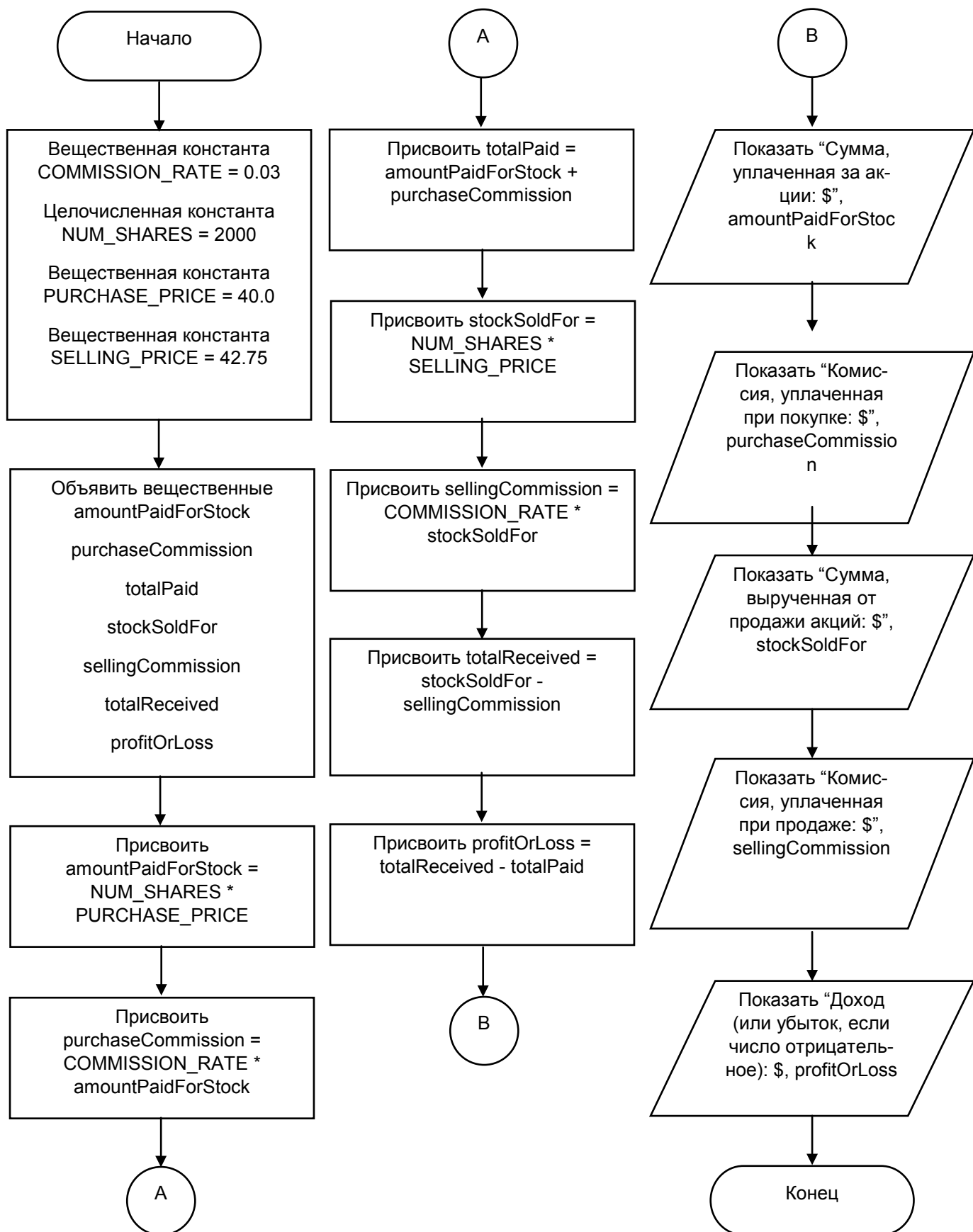
# Вычислить сумму, за которую Джо продал акции.
stockSoldFor = NUM_SHARES * SELLING_PRICE

# Вычислить сумму комиссии, которую Джо уплатил брокеру
# при продаже акций.
sellingCommission = COMMISSION_RATE * stockSoldFor

# Вычислить сумму денег, оставшихся после того, как Джо
# заплатил брокеру.
totalReceived = stockSoldFor - sellingCommission

# Вычислить сумму дохода или убытка. Если эта сумма является
# положительным числом, то это доход. Если она является
# отрицательным числом, то это убыток.
profitOrLoss = totalReceived - totalPaid

# Напечатать требующиеся данные.
print("Сумма, уплаченная за акции: $", format(amountPaidForStock, '.2f'))
print("Комиссия, уплаченная при покупке: $", format(purchaseCommission, '.2f'))
print("Сумма, вырученная от продажи акций: $", format(stockSoldFor, '.2f'))
print("Комиссия, уплаченная при продаже: $", format(sellingCommission, '.2f'))
print("Доход (или убыток, если число отрицательное): $", format(profitOrLoss, '.2f'))
```



Упражнение 2.13. Выращивание винограда

```
# Эта программа вычисляет количество виноградных лоз, которые
# могут быть высажены на виноградной гряде.

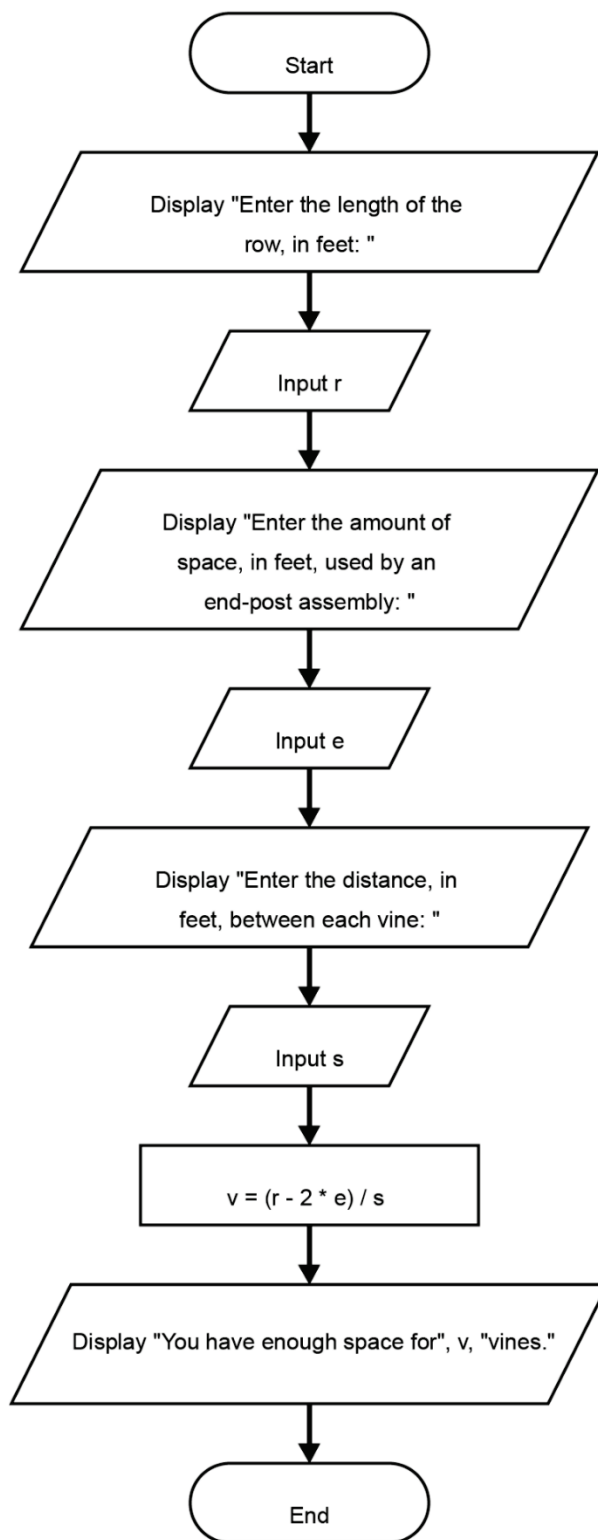
print('Введите длину гряды в метрах: ', end='')
r = float(input())

print('Введите объем пространства в метрах, используемого под трельяжные концевые опоры: ',
end='')
e = float(input())

print('Введите расстояние в метрах между каждой лозой: ', end='')
s = float(input())

# Вычислить количество виноградных лоз.
v = (r - 2 * e) / s

print('У вас достаточно места для', v, 'виноградных лоз.')
```



Надписи: начало; показать 'Введите длину гряды в метрах'; ввести r; показать 'Введите объем пространства в метрах, используемого под трельяжные концевые опоры'; ввести e; показать 'Введите расстояние в метрах между каждой лозой'; ввести s; показать 'У вас достаточно места для', v, 'виноградных лоз.'; конец

Упражнение 2.14. Сложный процент

```
# Эта программа вычисляет остаток основного счета в банке
# после начисления сложного процентного дохода.

print('Введите начальный остаток основного счета: ', end='')
p = float(input())

print('Введите годовую ставку процента: ', end='')
r = float(input())

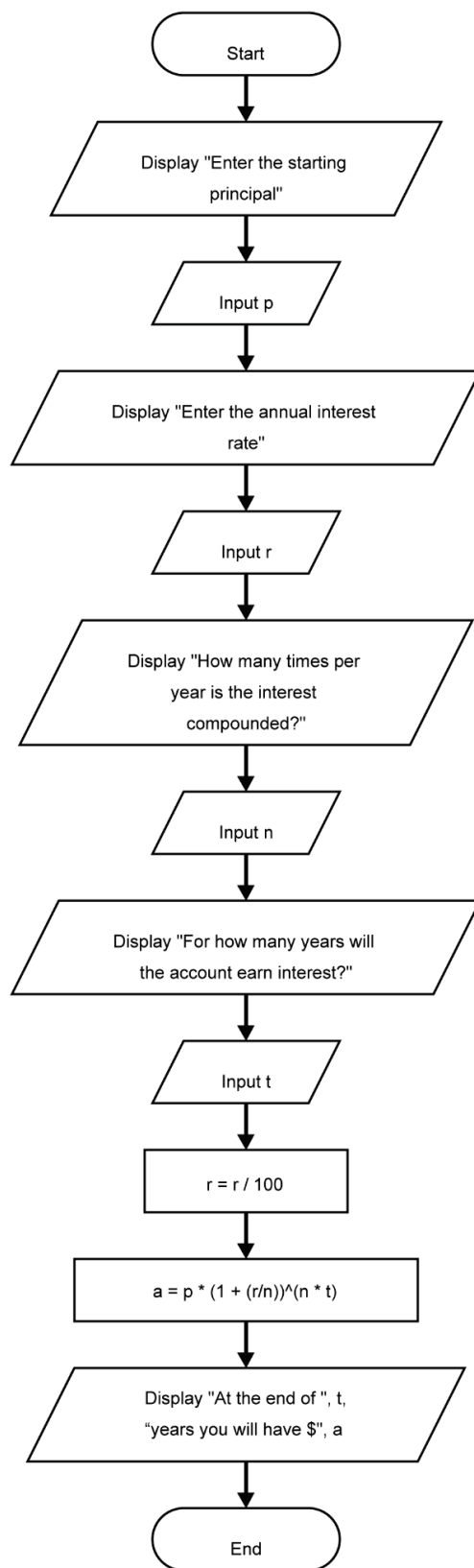
print('Сколько раз в году начисляется сложный процентный доход? ', end='')
n = int(input())

print('Сколько лет этот счет будет получать процентные начисления? ', end='')
t = int(input())

# Скорректировать процентную ставку.
r = r / 100

# Вычислить конечный остаток счета.
a = p * (1 + float(r) / n) ** (n * t)

# Показать конечный остаток счета.
print('В конце периода из', t, 'лет у вас будет $', format(a, ',.2f'))
```

Надписи: начало; показать 'Введите начальный остаток основного счета'; ввести p; показать 'Введите годовую ставку процента'; ввести r; показать 'Сколько раз в году начисляется сложный процентный доход?'; ввести n; показать 'Сколько лет этот счет будет получать процентные начисления?'; ввести t; показать 'В конце периода из', t, 'лет у вас будет \$', a; конец

Упражнение 2.15. Эскизы

Эскиз 1

```
import turtle

# Спрятать черепаху.
turtle.hideturtle()

# Сделать цвет заполнения синим.
turtle.fillcolor('blue')

# Начертить первый ромб.
turtle.begin_fill()
turtle.left(135)
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.end_fill()

# Начертить второй ромб.
turtle.begin_fill()
turtle.forward(100)
turtle.right(90)
turtle.forward(100)
turtle.right(90)
turtle.forward(100)
turtle.right(90)
turtle.forward(100)
turtle.end_fill()
```

Эскиз 2

```
import turtle

# Именованные константы
OUTER_TOP_X = 0
OUTER_TOP_Y = 200
INNER_TOP_X = OUTER_TOP_X
INNER_TOP_Y = OUTER_TOP_Y / 2
BASE_LEFT_X = -100
BASE_LEFT_Y = 0
BASE_RIGHT_X = 100
BASE_RIGHT_Y = 0

# Спрятать черепаху и поднять перо.
turtle.hideturtle()
turtle.penup()
```

```

# Переместить перо в правый нижний угол.
turtle.goto(BASE_RIGHT_X, BASE_RIGHT_Y)

# Сделать цвет заполнения синим и опустить перо.
turtle.fillcolor('blue')
turtle.pendown()

# Начертить внешний треугольник.
turtle.goto(OUTER_TOP_X, OUTER_TOP_Y)
turtle.goto(BASE_LEFT_X, BASE_LEFT_Y)
turtle.goto(BASE_RIGHT_X, BASE_RIGHT_Y)

# Начертить внутренний треугольник.
turtle.begin_fill()
turtle.goto(INNER_TOP_X, INNER_TOP_Y)
turtle.goto(BASE_LEFT_X, BASE_LEFT_Y)
turtle.goto(BASE_RIGHT_X, BASE_RIGHT_Y)
turtle.end_fill()

```

Эскиз 3

```

import turtle

# Именованные константы
TOP_SQUARE_TOP_LEFT_X = -100
TOP_SQUARE_TOP_LEFT_Y = 100
TOP_SQUARE_TOP_RIGHT_X = 0
TOP_SQUARE_TOP_RIGHT_Y = 100
TOP_SQUARE_BOTTOM_LEFT_X = -100
TOP_SQUARE_BOTTOM_LEFT_Y = 0
TOP_SQUARE_BOTTOM_RIGHT_X = 0
TOP_SQUARE_BOTTOM_RIGHT_Y = 0

BOTTOM_SQUARE_TOP_LEFT_X = 0
BOTTOM_SQUARE_TOP_LEFT_Y = 0
BOTTOM_SQUARE_TOP_RIGHT_X = 100
BOTTOM_SQUARE_TOP_RIGHT_Y = 0
BOTTOM_SQUARE_BOTTOM_RIGHT_X = 100
BOTTOM_SQUARE_BOTTOM_RIGHT_Y = -100
BOTTOM_SQUARE_BOTTOM_LEFT_X = 0
BOTTOM_SQUARE_BOTTOM_LEFT_Y = -100

# Спрятать черепаху и поднять перо.
turtle.hideturtle()
turtle.penup()

# Начертить самый верхний квадрат
turtle.goto(TOP_SQUARE_BOTTOM_RIGHT_X, TOP_SQUARE_BOTTOM_RIGHT_Y)
turtle.pendown()

turtle.goto(TOP_SQUARE_TOP_RIGHT_X, TOP_SQUARE_TOP_RIGHT_Y)
turtle.goto(TOP_SQUARE_TOP_LEFT_X, TOP_SQUARE_TOP_LEFT_Y)
turtle.goto(TOP_SQUARE_BOTTOM_LEFT_X, TOP_SQUARE_BOTTOM_LEFT_Y)

```

```

turtle.goto(TOP_SQUARE_BOTTOM_RIGHT_X, TOP_SQUARE_BOTTOM_RIGHT_Y)
turtle.penup()

# Начертить нижний квадрат
turtle.goto(BOTTOM_SQUARE_BOTTOM_RIGHT_X, BOTTOM_SQUARE_BOTTOM_RIGHT_Y)
turtle.pendown()
turtle.goto(BOTTOM_SQUARE_TOP_RIGHT_X, BOTTOM_SQUARE_TOP_RIGHT_Y)
turtle.goto(BOTTOM_SQUARE_TOP_LEFT_X, BOTTOM_SQUARE_TOP_LEFT_Y)
turtle.goto(BOTTOM_SQUARE_BOTTOM_LEFT_X, BOTTOM_SQUARE_BOTTOM_LEFT_Y)
turtle.goto(BOTTOM_SQUARE_BOTTOM_RIGHT_X, BOTTOM_SQUARE_BOTTOM_RIGHT_Y)
turtle.penup()

# Соединить углы
turtle.goto(BOTTOM_SQUARE_BOTTOM_RIGHT_X, BOTTOM_SQUARE_BOTTOM_RIGHT_Y)
turtle.pendown()
turtle.goto(TOP_SQUARE_TOP_LEFT_X, TOP_SQUARE_TOP_LEFT_Y)
turtle.penup()
turtle.goto(BOTTOM_SQUARE_TOP_RIGHT_X, BOTTOM_SQUARE_TOP_RIGHT_Y)
turtle.pendown()
turtle.goto(TOP_SQUARE_TOP_RIGHT_X, TOP_SQUARE_TOP_RIGHT_Y)
turtle.penup()
turtle.goto(BOTTOM_SQUARE_BOTTOM_LEFT_X, BOTTOM_SQUARE_BOTTOM_LEFT_Y)
turtle.pendown()
turtle.goto(TOP_SQUARE_BOTTOM_LEFT_X, TOP_SQUARE_BOTTOM_LEFT_Y)

```

Эскиз 4

```

import turtle

# Именованные константы
RADIUS = 100
STARTING_POINT_X = -250
STARTING_POINT_Y = 0
HSHIFT = 125
VSHIFT = 100

# Начертить круг #1
x = STARTING_POINT_X
y = STARTING_POINT_Y
turtle.penup()
turtle.goto(x, y)
turtle.pendown()
turtle.circle(RADIUS)

# Начертить круг #2
x += HSHIFT
y -= VSHIFT
turtle.penup()
turtle.goto(x, y)
turtle.pendown()
turtle.circle(RADIUS)

```

```
# Начертить круг #3
x += HSHIFT
y = 0
turtle.penup()
turtle.goto(x, y)
turtle.pendown()
turtle.circle(RADIUS)
```

```
# Начертить круг #4
x += HSHIFT
y -= VSHIFT
turtle.penup()
turtle.goto(x, y)
turtle.pendown()
turtle.circle(RADIUS)
```

```
# Начертить круг #5
x += HSHIFT
y = 0
turtle.penup()
turtle.goto(x, y)
turtle.pendown()
turtle.circle(RADIUS)
```

Эскиз 5

```
import turtle

# Именованные константы
CENTER_X = 0
CENTER_Y = 0
X_AXIS_LENGTH = 200
Y_AXIS_LENGTH = 200
RADIUS = 25
SOUTH = 270
EAST = 0

# Спрятать черепаху и задать скорость анимации.
turtle.hideturtle()
turtle.speed(0)

# Начертить ось X
x = CENTER_X - (X_AXIS_LENGTH / 2)
y = CENTER_Y
turtle.penup()
turtle.goto(x, y)
turtle.pendown()
turtle.forward(X_AXIS_LENGTH)

# Начертить ось Y
x = CENTER_X
y = CENTER_Y + (Y_AXIS_LENGTH / 2)
```

```

turtle.penup()
turtle.goto(x, y)
turtle.pendown()
turtle.setheading(SOUTH)
turtle.forward(X_AXIS_LENGTH)

# Начертить центральный круг
x = CENTER_X
y = CENTER_Y - RADIUS
turtle.penup()
turtle.setheading(EAST)
turtle.goto(x, y)
turtle.pendown()
turtle.circle(RADIUS)

# Написать "Север"
x = CENTER_X - 10
y = CENTER_Y + (Y_AXIS_LENGTH / 2)
turtle.penup()
turtle.goto(x, y)
turtle.pendown()
turtle.write("Север")

# Написать "Юг"
x = CENTER_X - 10
y = CENTER_Y - (Y_AXIS_LENGTH / 2) - 10
turtle.penup()
turtle.goto(x, y)
turtle.pendown()
turtle.write("Юг")

# Написать "Запад"
x = CENTER_X - (X_AXIS_LENGTH / 2) - 25
y = CENTER_Y - 7
turtle.penup()
turtle.goto(x, y)
turtle.pendown()
turtle.write("Запад")

# Написать "Восток"
x = CENTER_X + (X_AXIS_LENGTH / 2) + 2
y = CENTER_Y - 7
turtle.penup()
turtle.goto(x, y)
turtle.pendown()
turtle.write("Восток")

```

Эскиз 6

```

import turtle

# Именованные константы

```

```

UPPER_LEFT_X = -100
UPPER_LEFT_Y = 100
UPPER_RIGHT_X = 100
UPPER_RIGHT_Y = 100
LOWER_LEFT_X = -100
LOWER_LEFT_Y = -100
LOWER_RIGHT_X = 100
LOWER_RIGHT_Y = -100
CENTER_X = 0
CENTER_Y = 0
GAP = 20

# Спрятать черепаху и задать скорость анимации.
turtle.hideturtle()
turtle.speed(0)

# Начертить сплошные линии.
turtle.penup()
turtle.goto(UPPER_LEFT_X, UPPER_LEFT_Y)
turtle.pendown()
turtle.goto(LOWER_RIGHT_X, LOWER_RIGHT_Y)
turtle.penup()
turtle.goto(UPPER_RIGHT_X, UPPER_RIGHT_Y)
turtle.pendown()
turtle.goto(LOWER_LEFT_X, LOWER_LEFT_Y)
turtle.penup()
turtle.goto(UPPER_LEFT_X, UPPER_LEFT_Y)
turtle.pendown()
turtle.goto(LOWER_LEFT_X, LOWER_LEFT_Y)
turtle.penup()
turtle.goto(UPPER_RIGHT_X, UPPER_RIGHT_Y)
turtle.pendown()
turtle.goto(LOWER_RIGHT_X, LOWER_RIGHT_Y)

# Начертить верхнюю пунктирную линию.
turtle.penup()
turtle.goto(UPPER_LEFT_X, UPPER_LEFT_Y)
turtle.pendown()
turtle.forward(GAP)
turtle.penup()
turtle.forward(GAP)
turtle.pendown()
turtle.forward(GAP)
turtle.penup()
turtle.forward(GAP)
turtle.pendown()
turtle.forward(GAP)
turtle.penup()
turtle.forward(GAP)
turtle.pendown()

```

```
turtle.forward(GAP)
turtle.penup()
turtle.forward(GAP)
turtle.pendown()
turtle.forward(GAP)
turtle.penup()
turtle.forward(GAP)

# Начертить нижнюю пунктирную линию.
turtle.penup()
turtle.goto(LOWER_LEFT_X, LOWER_LEFT_Y)
turtle.pendown()
turtle.forward(GAP)
turtle.penup()
turtle.forward(GAP)
turtle.pendown()
turtle.forward(GAP)
turtle.penup()
turtle.forward(GAP)
turtle.pendown()
turtle.forward(GAP)
turtle.penup()
turtle.forward(GAP)
turtle.pendown()
turtle.forward(GAP)
turtle.penup()
turtle.forward(GAP)
turtle.pendown()
turtle.forward(GAP)
turtle.penup()
turtle.forward(GAP)

# Начертить точки.
turtle.penup()
turtle.goto(UPPER_LEFT_X, UPPER_LEFT_Y)
turtle.dot()
turtle.goto(UPPER_RIGHT_X, UPPER_RIGHT_Y)
turtle.dot()
turtle.goto(LOWER_LEFT_X, LOWER_LEFT_Y)
turtle.dot()
turtle.goto(LOWER_RIGHT_X, LOWER_RIGHT_Y)
turtle.dot()
turtle.goto(CENTER_X, CENTER_Y)
turtle.dot()
```