

Положение к проведению учебно-диагностического входного контроля в целях набора 8-х классов математического профиля

Оглавление

Общие положения	2
Тематическое содержание	2
Основные алгоритмы	3
Дополнительно проверяемые элементы	4
Контрольные вопросы для самопроверки	5
Структура экзаменационного билета	6
Первое задание	6
Второе задание	7
Третье задание	11
Четвёртое задание	15
Пятое задание	15
Модель экзаменационного билета.....	17
Список рекомендуемой литературы.....	19

Общие положения

Отбор в учебные профили, связанные с углубленным изучением информатики и программирования, предусматривает успешное прохождение учебно-диагностических контрольных мероприятий, включая предмет «Информатика и ИКТ» (в дальнейшем – «Информатика»).

Для предмета информатики предусмотрено проведение экзамена продолжительностью **120** минут.

Экзамен состоит из двух диагностических частей: теоретической и практической. На экзамене учащийся демонстрирует как фактическое знание содержательного теоретического материала, так и практические умения создания, применения и использования алгоритмов решения задач, понимание принципов построения и исполнения основных алгоритмических структур, а также знание отдельно выносимых основных алгоритмов, широко применяющихся в практике программирования.

В качестве базового языка программирования для сдачи экзамена выступает язык Python версии не ниже 3-ей, однако для учащихся, поступающих из сторонних по отношению к комплексу «Школа №1514» учебных заведений, допускается использование иных распространённых языков программирования, например, языка C++, Pascal или Java. Тем не менее, знания языка Python потребуются для дальнейшего обучения, поэтому настоятельно рекомендуется именно его использование при сдаче экзамена.

Тематическое содержание

Тематическое содержание охватывает все основополагающие аспекты темы «Алгоритмизация и программирование», соответствующие кодификаторам ФГОС:

- **1.2 «Алгоритмизация и программирование»;**
- **1.3 «Основы логики», п. 1.3.1 «Алгебра логики»**
- **2.8 «Технологии программирования»**

Основными блоками тематического содержания являются следующие:

1. Алгоритмы и их свойства
2. Алгоритмические структуры:
 - a. Линейная структура
 - b. Структура ветвления

с. Циклическая структура

Полная версия тематического содержания, проверяемого на экзамене, представлена ниже:

1. Понятие алгоритма и его свойства
2. Способы записи алгоритмов:
 - a. Словесная форма
 - b. Блок-схемная форма
 - c. Программный код
3. Алгоритмические структуры:
 - a. Линейная структура
 - b. Структура ветвления
 - c. Циклическая структура
4. Линейные алгоритмы. Операторы ввода/вывода и диалоговые программы
5. Операторы целочисленного деления:
 - a. Оператор «//»
 - b. Оператор «%»
6. Условный оператор (оператор «IF»):
 - a. Формы условного оператора
 - b. Логические условия и логические операции
 - c. Логический тип данных
 - d. Составление сложных логических выражений для решения логических задач
7. Циклический алгоритм. Общие принципы построения и структурные составляющие любого цикла
8. Оператор циклов языка Python «for» и «while»
9. Вложенные циклы

Основные алгоритмы

При подготовке к экзамену необходимо изучить, понимать и уметь применять на практике некоторые распространённые и широко используемые алгоритмы, список которых перечислен ниже:

1. Алгоритм нахождения любой цифры заданного целого n-значного числа
2. Алгоритм нахождения факториала любого целого положительного числа

3. Алгоритм возведения любого целого положительного числа в любую целую положительную степень, не используя стандартные функции языка программирования
4. Алгоритм нахождения суммы, количества и среднего арифметического чисел последовательности, отвечающих заданному пользователем условию
5. Алгоритм нахождения минимального (максимального) числа произвольной последовательности чисел
6. Алгоритм поиска всех делителей произвольного целого положительного числа
7. Алгоритм определения, является ли заданное пользователем целое положительное число простым
8. Алгоритм подсчёта суммы, количества и произведения цифр произвольного целого положительного числа
9. Алгоритм поиска НОД и НОК

Дополнительно проверяемые элементы

Помимо основных универсальных алгоритмов на экзамене проверяется умение составлять и применять различные алгоритмы обработки последовательностей или заданного ряда чисел, обработки цифр числа, форматированный вывод, запись и вычисление сложных логических выражений, использование логического типа данных, алгоритмы вывода данных на экран в заданном виде и прочие элементы, тематическое содержание которых соответствует описанному выше тематическому содержанию.

В частности, проверяются следующие умения по составлению алгоритмов и выполнению необходимой программной обработки:

1. Алгоритм определения минимального (максимального) из трёх чисел без использования циклов
2. Составление логического выражения, отвечающего заданной области на двумерной плоскости
3. Вывод в табличном виде (с использованием форматного вывода) с заданным шагом (значений функции, стоимости товара по граммам и так далее)
4. Вывод по заданному шаблону
Пример: вводится натуральное N . Вывести треугольник чисел.
Для $N=3$ на экран выводится:

- 1
22
333
5. Вывод последовательности чисел по заданному условию.
Пример: вывести квадраты 100 нечётных чисел
 6. Нахождение и вывод заданного количества первых чисел по заданному условию.
Пример: найти первые 100 чисел, делящихся на 4, 5 и 6 одновременно
 7. Обработка последовательности с заданным или заданным размером (количеством элементов последовательности).
 8. Обработка ряда чисел с заданным количеством элементов (нахождение суммы ряда, вывод квадратов и так далее)
 9. Выделение любой цифры N-значного числа, если заранее известно значение N (без использования циклов). Нахождение суммы всех цифр
 10. Подбор параметров по условию.
Пример: найти все пары чисел, произведение которых даёт заданное целое число
 11. Вывод всех N-значных чисел, отвечающих заданному условию.
Пример условия: число содержит ровно один «0»

Контрольные вопросы для самопроверки

Для самоконтроля изученного тематического содержания предлагаются следующие контрольные вопросы для самопроверки:

1. Что такое алгоритм? Какие бывают виды алгоритмических структур?
2. Какие способы записи алгоритма вы знаете?
3. Что такое диалоговая форма программы и как она организуется в языке Python?
4. Какие параметры есть у команды «print()»?
5. Какие типы данных вы знаете и для чего они используются?
Назовите отдельные операции, применимые лишь к определённым типам данных
6. Что такое операторы целочисленного деления и как они используются в решении различных задач? Привести конкретные примеры
7. Что такое операторы целочисленного деления? Чем целочисленное деление отличается от простого арифметического?
Как использовать операторы целочисленного деления для определения чётности числа и выделения его цифр?

8. Для чего используется условная конструкция? Какие бывают формы условных конструкций языка Python? Запишите общий вид различных форм условных конструкций
9. Что такое логическое высказывание? Что такое логические операции? Какие основные логические операции вы знаете?
Продемонстрируйте применение этих операций на конкретных примерах, приведите составное логическое выражение
10. Как задаётся и для чего применяется логический тип данных? Какие значения принимают переменные этого типа данных?
Приведите примеры применения логического типа данных
11. Назначение циклической алгоритмической структуры. Общие принципы построения и структурные составляющие любого цикла.
12. Оператор «FOR». Назначение, структура, способы задания и особенности использования в решении различных задач
13. Оператор «WHILE». Назначение, структура, способы задания и особенности использования в решении различных задач
14. Отличия построения и особенности использования операторов «WHILE» и «FOR» языка Python
15. Как перейти от одного циклического оператора к другому?
16. Для чего используются и как задаются вложенные циклы?
Особенности построения и выполнения вложенных циклов.
17. Приведите пример задания вложенного цикла

Структура экзаменационного билета

Экзаменационный билет состоит из 5-ти заданий, некоторые из которых могут состоять из нескольких частей.

Первое задание

Первое задание проверяет основные базовые теоретико-практические знания и навыки учащегося и представлено несколькими небольшими тестовыми вопросами и тестовыми заданиями на умение строить и читать различные циклические структуры, использовать операторы целочисленного деления, переходить от одного циклического оператора к другому, понимать количество выполнений циклического алгоритма и понимание других базовых вопросов программирования или особенностей использования языка Python.

Пример задания

1. Что выведет следующий фрагмент программы:

```
a = 50
a = int(input())
a = a + 25
print('a')
```

2. Напишите, что выведет на экран следующий алгоритм:

```
a = 4 + 2 // 2
if a > 5:
    print(a)
else:
    print(-a)
```

3. Чему будет равно значение переменной «k» после выполнения следующего выражения:

```
k = 16 % 5
```

4. Чему будет равно значение переменной «k» после выполнения следующего выражения:

```
k = 5 % 16
```

5. Сколько раз выполнится цикл:

```
n = 15
while n >= 5:
    n = n - 3
```

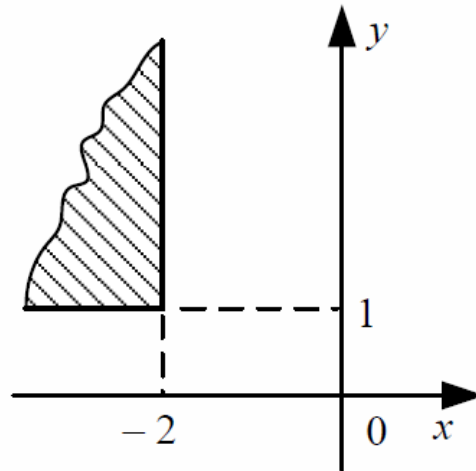
Второе задание

Второе задание проверяет знание и умение использовать элементарные логические операции «ИЛИ», «И», «НЕ», записывая на их основе составные логические выражения для определения принадлежности заданной пользователем точки той или иной геометрической области (областям), определённой (определённых) в задании. Области могут быть ограничены, представляя собой очерченную фигуру, например, фигуру прямоугольника, могут иметь бесконечную протяжённость в одном или нескольких направлениях. В других видах заданий учащимся, наоборот, необходимо правильно прочитать и получить результат выполнения сложного логического выражения.

Примеры задания

Пример 1

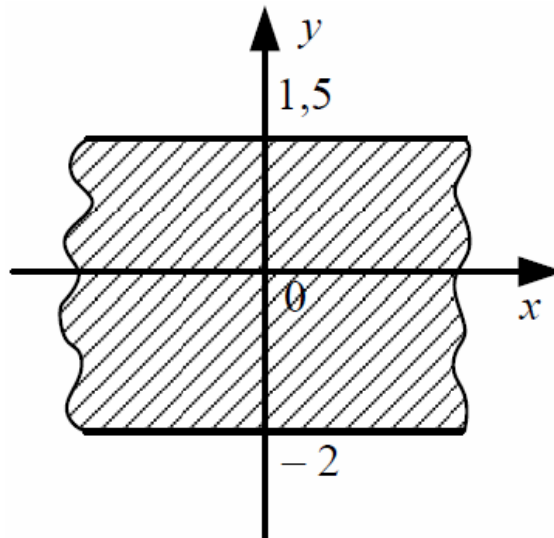
Напишите программу, определяющую, попадает ли указанная пользователем точка с координатами $(x;y)$ в заштрихованную область:



При задании логического условия использование двойных неравенств недопустимо.

Пример 2

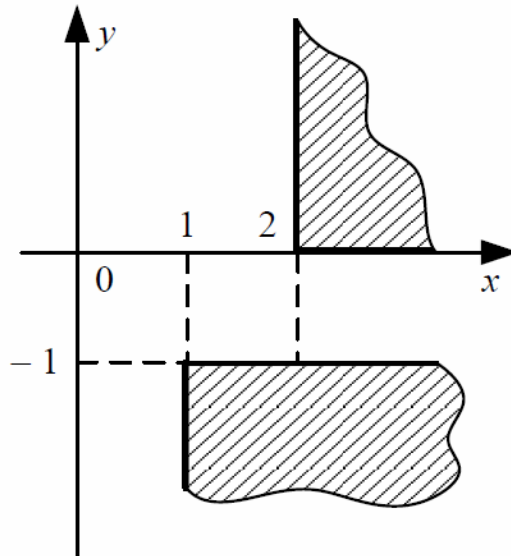
Напишите программу, определяющую, попадает ли указанная пользователем точка с координатами $(x;y)$ в заштрихованную область:



При задании логического условия использование двойных неравенств недопустимо.

Пример 3

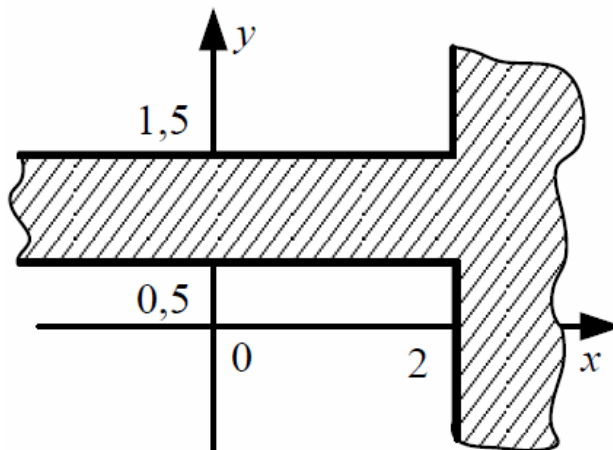
Напишите программу, определяющую, попадает ли указанная пользователем точка с координатами $(x;y)$ в какую-либо заштрихованную область:



При задании логического условия использование двойных неравенств недопустимо.

Пример 4

Напишите программу, определяющую, попадает ли указанная пользователем точка с координатами $(x;y)$ в заштрихованную область:



При задании логического условия использование двойных неравенств недопустимо.

Пример 5

Что будет выведено в результате выполнения следующего фрагмента алгоритма:

```
k = 16
t = False
if k >= 18 or (k < 16 and t == False):
    print('YES')
else:
    print('NO')
```

Пример 6

Что будет выведено в результате выполнения следующего алгоритма:

```
t1 = False
t2 = False
t3 = True
if (t1 == t2) or t1 and t2 and t3 or (t1 != t3):
    print(1)
elif (t1 != t2) and t1 or t2 and t3 or (t1 == t3):
    print(2)
else:
    print(3)
```

Третье задание

Третье задание проверяет умение читать, составлять и применять элементарные математические алгоритмы для решения односложных задач с использованием циклических конструкций.

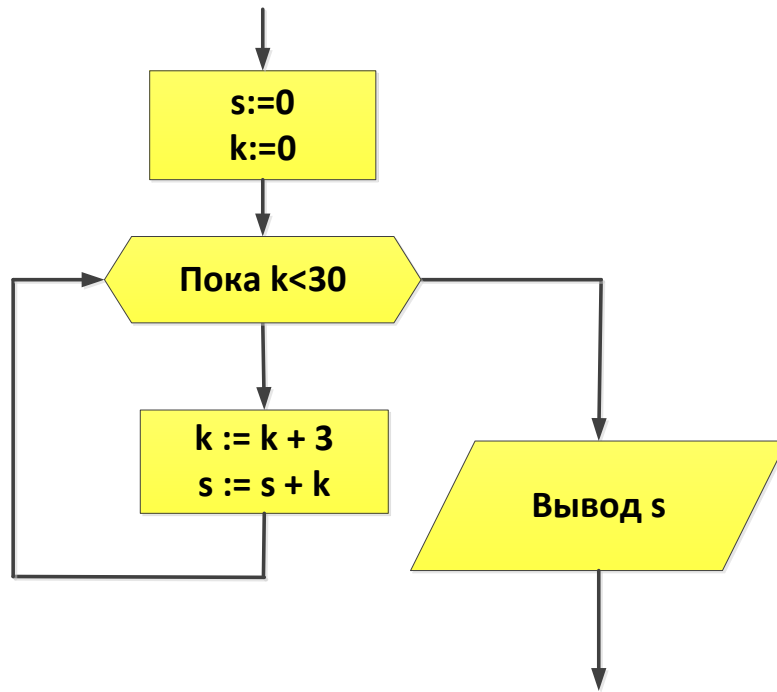
Третье задание состоит из двух частей:

1. Задача на умение читать и исполнять алгоритмы, заданные с помощью блок-схемного способа записи
2. Создание итерационной таблицы пошагового выполнения циклического алгоритма с записью значений участвующих в цикле переменных и счётчиков цикла. Ввиду автоматизированной проверки решений учащихся экзаменуемому может быть предложена упрощённая форма записи, включающая одну или несколько переменных, в формате, указанном в пояснении к данному заданию. Однако для учебной подготовки к данному типу заданий и самопроверки рекомендуется заполнять полные таблицы для всех участвующих в расчётах переменных, поэтому такие таблицы приводятся для примера ниже.

Примеры задач первой части

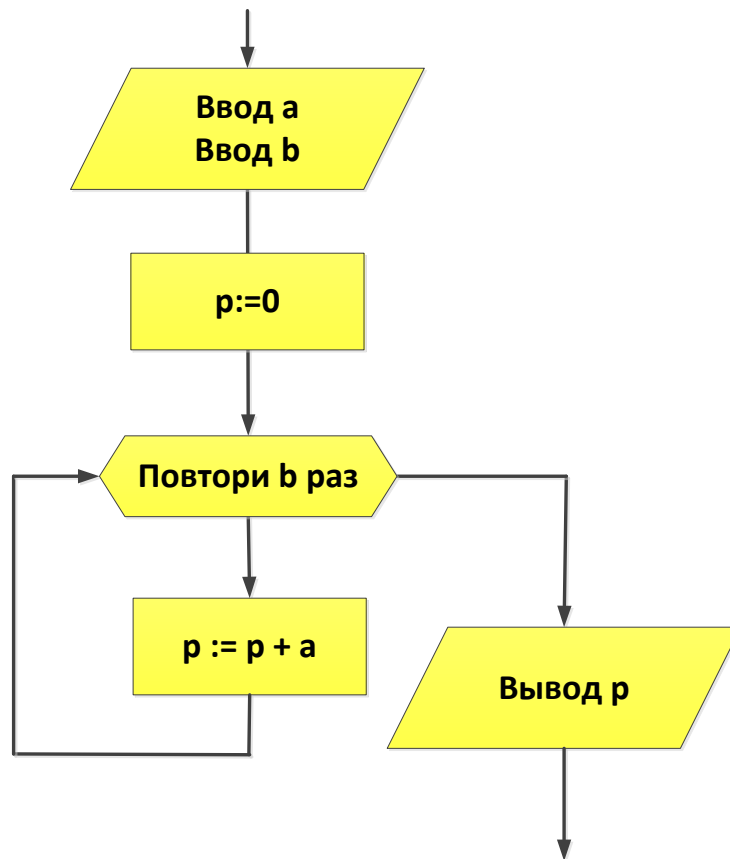
Пример 1

Напишите, что будет выведено на экран как итог выполнения следующего циклического алгоритма:



Пример 2

Что в общем случае для любого целого числа n находит записанный ниже алгоритм? В ответе необходимо дать общую формулу с использованием переменных, участвующих в алгоритме.



Примеры задач второй части

Заполните итерационную таблицу пошагового выполнения фрагмента циклического алгоритма, представленного ниже:

Пример 1

```

x = 7
while x > -4:
    k = x * x
    x = x - 2
  
```

№ итерации	Переменная «k»	Переменная «x»
0 – начальная инициализация	-	7
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

Пример 2

x = 3

k = 1

i = 0

while i < 5:

 x = x + 1

 k = k * x

 i = i + 1

№ итерации	Переменная «i» – счётчик цикла	Переменная «x»	Переменная «k»
0 – начальная инициализация	0	3	1
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

Четвёртое задание

Четвёртое задание проверяет элементарные навыки программирования, умение составлять алгоритм решения задачи, записать и реализовать его на языке программирования, отладить программу и в конечном счёте получить полностью работоспособный программный код, правильно работающий на всех входных тестовых данных, то есть выдающих верные выходные данные при любых входных данных, включая все ограничения из условия задачи.

Примеры заданий

Пример 1

Непустая последовательность целых чисел, вводимых с клавиатуры, заканчивается нулём. Найдите произведение нечётных чисел.

Пример 2

Найдите значение выражения, если количество слагаемых определяет пользователь программы:

$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{N}$, где N – номер соответствующего члена последовательности

Пример 3

Пользователь задаёт размер и саму последовательность.

Определите, сколько раз в данной последовательности встретилось максимальное и минимальное число.

Пример 4

Определите, является ли заданная пользователем последовательность убывающей?

Пользователь вводит последовательность до значения «0», которое является признаком конца ввода и не входит в саму последовательность.

Пятое задание

Пятое задание представлено задачами, в решении которых обычно используются вложенные циклы. Подобный тип заданий не обладает большой математической и алгоритмической сложностью, позволяя экзаменаторам оценить основные навыки программирования учащегося.

Для решения этого задания необходимо хорошо понимать принципы композиции алгоритма, особенности выполнения всех видов алгоритмических структур, в частности, циклических структур разного вида.

Примеры заданий

Пример 1

Пользователь задаёт два целых числа N и K , определяющих диапазон целых чисел $[N, K]$.

Найдите в этом диапазоне число с максимальной суммой делителей.

Пример 2

Пользователь задаёт два целых числа N и K , определяющих диапазон целых чисел $[N, K]$.

Определите общее количество простых делителей у всех чисел из этого диапазона. Если у одного числа два и более простых одинаковых делителя (как, например, у числа «25» это два делителя «5»), считать только один из них.

Пример 3

Пользователь программы вводит два целых положительных числа « a » и « b », задающих диапазон целых чисел $[a; b]$.

Найдите в пределах этого диапазона и выведите на экран все числа, начинающиеся и заканчивающиеся на цифру «8».

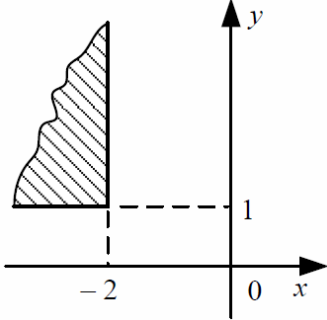
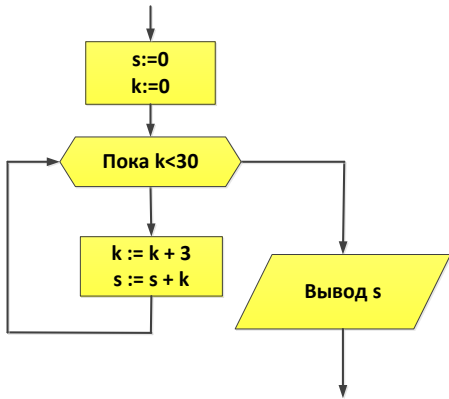
Модель экзаменационного билета

Ввиду полностью автоматизированной формы проведения экзамена с получением заданий в электронном виде и их автоматической проверкой прилагается модель экзаменационного билета для того, чтобы учащийся понимал приблизительный содержательный вид электронного экзаменационного билета.

Билет № ____

Фамилия и имя, класс:

№ задания	Задание / Ссылка на задание	Ответ	№ задачи в задании
1	Что выведет следующий фрагмент программы: a = 50 a = a + 25 print('a')		
	Напишите, что выведет на экран следующий алгоритм: a = 4 + 2 / 2 if a > 5: print(a) else: print(-a)		
	Чему будет равно значение переменной «k» после выполнения следующего выражения: k = 12 % 5		
	Чему будет равно значение переменной «k» после выполнения следующего выражения: k = 8 % 10		
	Сколько раз выполнится цикл: n = 19 while n >= 6: n = n - 3		

2	<p>Напишите программу, определяющую, попадает ли указанная пользователем точка с координатами (x;y) в заштрихованную область:</p>  <p>При задании логического условия использование двойных неравенств недопустимо.</p>	
3.1	<p>Напишите, что будет выведено на экран как итог выполнения следующего циклического алгоритма:</p>  <p>Ответ:</p>	
3.2	<p>Заполните итерационную таблицу пошагового выполнения фрагмента циклического алгоритма, представленного ниже:</p> <pre> x = 7 while x > -4: k = x * x x = x - 2 </pre> <p><i>Для выполнения данного задания необходимо заполнить уже подготовленную таблицу, которую выдаёт экзаменатор вместе с экзаменационным билетом</i></p>	
4	https://official.contest.yandex.ru/contest/27336/enter	Задача А
5	https://official.contest.yandex.ru/contest/27337/enter	Задача А

Список рекомендуемой литературы

1. Д.М. Златопольский «Основы программирования на языке Python»
2. Зед Шоу «Лёгкий способ выучить Python 3»
3. Майк Макграт «Python. Программирование для начинающих»
4. Эл Свейгарт «Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих»
5. Задачники:
 - a. Павловская Т.А. "Программирование на языке высокого уровня"
 - b. Т.Ю. Грацианова «Программирование в примерах и задачах»
 - c. Д.М. Златопольский «Сборник задач по программированию»
 - d. В.А. Дагене, Г.К. Григас, К.Ф. Аугутис «100 задач по программированию»

Для подготовки подойдут любые номера изданий и сами издательства.