

Подготовка учащихся к ГИА по Информатике по теме «Алгоритмизации и программирование»



Лектор: Апольских Евгения Ивановна,
старший преподаватель кафедры
теоретических основ информатики



Содержательная линия «Алгоритмизация и программирование» представлена в ОГЭ следующими заданиями:

- ◆ **Задание №5. Анализ алгоритмов для Калькулятора**
- ◆ **Задание №6. Анализ программ с ветвлениями**
- ◆ **Задание №15_1. Составление программы для исполнителя Робот**
Задание №15_2. Составление программы на обработку потока данных

Основная цель заданий:

- ◆ На уровне воспроизведения знаний: понятие алгоритма, его свойства, способы записи, основные алгоритмические конструкции
- ◆ На уровне сформированности умений применять свои знания в стандартной ситуации, это: использовать стандартные алгоритмические конструкции для построения алгоритмов для формальных исполнителей и формально исполнять алгоритмы, записанные на естественном и алгоритмическом языках
- ◆ На уровне сформированности умений применять свои знания в новой ситуации: разрабатывать алгоритм для формального исполнителя или на языке программирования с использованием условных инструкций и циклов, а также логических связей при задании условий

Задание №5. Анализ алгоритмов для Калькулятора. Ученики должны уметь анализировать простые алгоритмы для конкретного исполнителя с фиксированным набором команд. Уровень сложности данного задания – базовый.

У исполнителя Бета две команды, которым присвоены номера:

1. прибавь b
2. умножь на 3

(b – неизвестное натуральное число).

Выполняя первую из них, Бета увеличивает число на экране на b , а выполняя вторую, умножает это число на 3. Программа для исполнителя Бета – это последовательность номеров команд. Известно, что программа **21212** переводит число 6 в число **282**. Определите значение b .

Для решения этой задачи необходимо последовательно выполнить заданную последовательность команд к числу 6 , составив математическое выражение, из которого потом и найти b .

$$((6 * 3 + b)*3 + b)*3 = 282$$

$$4b = 94 - 54$$

$$b = 10$$

Задание №6. Анализ программ с ветвлениями. Формально исполнять алгоритмы, записанные на языке программирования. Уровень сложности данного задания – базовый.

```
x = int(input())
y = int(input())
if y <= 100 or x > 90:
    print("ДА")
else:
    print("НЕТ")
```

Было проведено 10 запусков этой программы, при которых в качестве значений переменных x и y вводились следующие пары чисел:

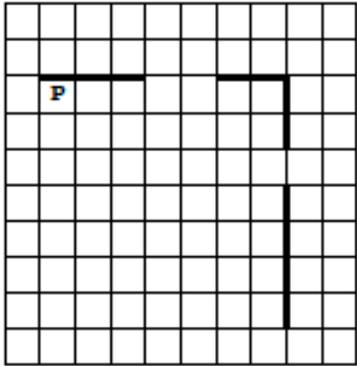
(105, 100); (95, 110); (100, 95);
(95, 90); (105, 90); (85, 110); (100, 110); (85, 105); (85, 95); (90, 100)

Сколько было запусков, при которых программа напечатала «НЕТ»?

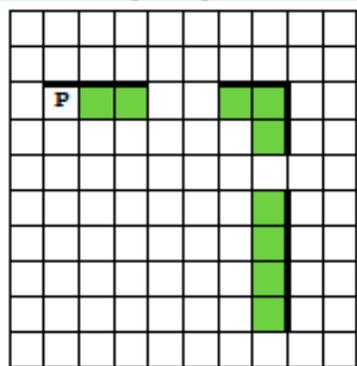
Необходимо провести анализ программы, выделив условие печати нужного ответа и произвести подсчет пар.

- важно не перепутать логические операции в условии
- правильно определить, какая ветвь условного оператора должна выполняться
- правильно выполнить строгое / нестрогое сравнение
- контроль, за вводом переменных, чтобы не ошибиться при проверке условий

Задание 15.1 проверяет умение записать формальный алгоритм с использованием инструкций ветвления и цикла для формального исполнителя «Робот».



Робот должен закрасить все клетки, расположенные непосредственно ниже горизонтальной стены и левее вертикальной стены, кроме клетки, в которой находится Робот перед выполнением программы.



```
вправо
нц пока сверху занято
закрасить
вправо
кц
нц пока сверху свободно
вправо
кц
нц пока справа свободно
закрасить
вправо
кц
нц пока не справа свободно
закрасить
вниз
кц
нц пока справа свободно
вниз
кц
нц пока справа занято
закрасить
вниз
кц
```

Задание 15.2. Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 4, но не кратных 7. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 4 и не кратное 7. Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число: количество чисел, кратных 4, но не кратных 7.

Входные данные	Выходные данные
4	2
16	
28	
26	
24	

Правильная программа должна содержать следующие части:

- 1. Ввод данных.**
- 2. Нахождение ответа.**
- 3. Вывод ответа.**

Для проверки правильности алгоритма решения задачи, необходимо чтобы программа прошла представленные тесты.