Алгебра логики: Задание 15 (ДЕЛ)

❗ Важный момент:
Нужно понимать, что выражение должно быть тождественно истинно, т. е. истинно при любых допустимых значениях переменных x и y, а не только при некоторых наборах значений.

📎 Базовые логические операции в Python:

Логическая операция	Операция в Python
❗ A ∧ B (и)	❗ A and B
❗ A ∨ B (или)	❗ A or B
❗ A → B (импликация)	❗ A ⩽ B
❗ ¬A (не)	❗ not(A)

Пример задания

Обозначим через ДЕЛ(n,m) утверждение «натуральное число n делится без остатка на натуральное число m». Для какого наибольшего натурального числа A формула
¬ДЕЛ(x,A) → (ДЕЛ(x,6) → ¬ДЕЛ(x,9))
тождественно истинна (то есть принимает значение 1 при любом натуральном значении переменной x)?

Решение

❗ Решение руками:

1) Введём обозначения:
A = ДЕЛ(x,A); Д6 = ДЕЛ(x,6); Д9 = ДЕЛ(x,9).

Алгебра логики: Задание 15 (ДЕЛ)

Решение

! Решение руками:

2) Перепишем данную формулу с новыми обозначениями:
¬А → (Д6 → ¬Д9)

3) Рассмотрим x, при котором выражение (Д6 → ¬Д9) будет ложным.

4) В данном случае, это будут такие x, которые делятся и на 6, и на 9 без остатка. Наименьшее общее кратное этих чисел — 18.

5) Следовательно, для x = 18 выражение ¬А должно быть ложным, значит, число 18 должно делиться на А без остатка.

6) Следовательно, А может принимать значения:

А = 1, 2, 3, 6, 9, 18.

7) Наибольшим натуральным числом А, при условии x % А = 0, является 18.

Ответ: 18

! Решение программой:

1) Создаём функцию ДЕЛ, в которой будем проверять утверждение «натуральное число n делится без остатка на натуральное число m»:

```
def Del(n,m):
```

2) Внутри функции проверяем, делится ли n на m без остатка. Если да, то возвращаем с помощью return значение True, если не делится — False:

```
return n % m == 0
```

Алгебра логики: Задание 15 (ДЕЛ)

Решение

3) Перебираем с помощью цикла for значения переменной A в диапазоне от 1 до 999 включительно: for A in range (1,1000):

4) Объявляем переменную flag и присваиваем ей значение True: flag = True
С помощью данной переменной будем смотреть, подходит ли нам значение A.

5) Перебираем значения переменной x с помощью цикла for в диапазоне от 1 до 999: for x in range (1, 1000):

6) С помощью условного оператора if смотрим, если формула ложна при каком-либо x, то переменной flag присваиваем значение False и выходим из цикла перебора x:

```
if ((not(Del(x,A))) ≤ (Del(x,6) ≤ (not(Del(x,9))))) == False:
        flag = False
        break
```

7) После перебора всех значений x смотрим, была ли формула истинна при любом x. Если да, то сохраняем значение A в переменную max_A:

```
if flag == True:
    max_A = A
```

8) Выводим максимальное подходящее значение A: print(max_A)

Алгебра логики: Задание 15 (ДЕЛ)

Решение

Полный код Python 3:

```
def Del(n,m): # создаём функцию ДЕЛ
    return n % m == 0 # возвращаем значение True, если n делится на m без остатка, в противном случае — False

for A in range (1,1000): # перебираем значения A в интервале [1;999], A != 0, т. к. делить на 0 нельзя!
    flag = True # объявляем переменную flag и присваиваем ей значение True
    for x in range (1, 1000): # перебираем значения x в интервале [1;999]
        if ((not(Del(x,A))) ⩽ (Del(x,6) ⩽ (not(Del(x,9))))) == False: # если при значении x формула ложна
            flag = False # то присваиваем значение False переменной flag, отмечая, что данный A не подходит
            break # выходим из цикла (перебора) x
    if flag == True: # если переменная flag равна True (т. е. смотрим, при переборе всех x для данного A формула была всегда истинной?)
        max_A = A # если да, то присваиваем переменной max_A значение A
print(max_A) # выводим на экран последнее значение A
```

Место для заметок

Логические выражения. Задание 15

Логические операторы

A ∧ B	логическое умножение, конъюнкция
A ∨ B	логическое сложение, дизъюнкция
A → B	импликация (следование)
A ≡ B	эквивалентность (равносильность)

Логическаие операторы. Python

Для решения данного номера с помощью программы нам понадобятся представления логических операторов в языке программирования (Python)

“≠” — !=
“∧” — and
“∨” — or
“→” — <=
“≡” — =

Переменная «флаг»

Такая переменная нам нужна, когда требуется проверить элементы/функцию на наличие искомых значений или на определённый результат работы

```
flag = False
for i in [[0, 0, 0], [0, 1, 0], [0, 0, 0]]:
    for j in i:
        if j == 1:
            flag = True
print(flag)   # Вывод: True
```

Если нужно, например, проверить, есть ли в списках единица, можно использовать переменную «флаг».

Логические выражения. Задание 15

Пример задания

Нам дано выражение: (x & A ≠ 0) → (((x & 23 = 0) ∧ (x & 8 = 0)) → (x & 7 ≠ 0)). В качестве ответа определите наибольшее натуральное значение A, при котором данное выражение истинно для любого натурального значения x.

Алгоритм решения

Для удобства создадим функцию с переменными x, A и сделаем так, чтобы она возвращала нам выражение из условия.

```
def func(x, A):
    return (x & A != 0) <= (((x & 23 == 0) and (x & 8 == 0)) <= (x & 7 != 0))
```

Нам понадобится переменная flag = True. Если при некотором x наше выражение не истинно, переменная изменится на flag = False и остановит вложенный цикл с помощью break.

```
if func(x, A) == False:
    flag = False
    break
```

Для вывода подходящих значений будем использовать проверку нашего «флага» на истинность.

```
if flag == True:
    print(A)
```

```
def func(x, A):
    return (x & A != 0) <= (((x & 23 == 0) and
            (x & 8 == 0)) <= (x & 7 != 0))
for A in range(1, 1000):
    flag = True
    for x in range(1, 1000):
        if func(x, A) == False:
            flag = False
            break
    if flag == True:
        print(A)
```

Ответ: 31