# Множення двійкових чисел

# Піддубняк Д. 6.1219-2

# Умножение двоичных чисел  
# Поддубняк Д. 6.1219-2  
# Функции: \_\_init\_\_ = конструктор класса,обьявляет переменные, binarconverter = конвертор в двоичный вид,  
# multiply = метод умножения двоичных чисел, minichecker = метод формулы чисел для рассчетов,  
# bigger = метод определения большего числа(по длинне)  
  
  
class Digits:  
 def \_\_init\_\_(self, first\_digit, second\_digit):  
 *"""Конструктор класса,объявляет переменные"""* self.first\_digit = str(first\_digit)  
 self.second\_digit = str(second\_digit)  
 self.straight\_first = self.binarconverter(self.first\_digit) # Перевод в двоичную  
 self.straight\_second = self.binarconverter(self.second\_digit) # Перевод в двоичную  
 self.bigger\_digit = self.bigger(self.straight\_first, self.straight\_second) # Узнаем какое число больше  
 self.multiply\_result = self.multiply(self.straight\_first, self.straight\_second) # Вызов метода умножения  
  
 def \_\_repr\_\_(self): # Коррекция вывода на экран:  
 return f"""Первое число в двоичном виде: {self.straight\_first}\n  
Второе число в двоичном виде: {self.straight\_second}\n  
Результат умножения: {self.multiply\_result}"""  
  
 def binarconverter(self, x, res=''): # перевод десятичного числа в двоичную СС.  
 x = str(x)  
 if '-' in x: # Если отрицательное  
 x1 = '-'  
 x = int(x.replace('-', ''))  
 if x == 0:  
 return '0'  
 else:  
 while x > 0:  
 res = ('0' if x % 2 == 0 else '1') + res # Формула перевода в бинарный вид  
 x //= 2  
 return str(res)  
 else: # Если положительное  
 x = int(x)  
 if x == 0:  
 return '0'  
 else:  
 while x > 0:  
 res = ('0' if x % 2 == 0 else '1') + res # Формула перевода в бинарный вид  
 x //= 2  
 return str(res)  
  
 def multiply(self, a, b):  
 all\_digits = [] # Словарь для записи вычислений  
 b1 = list(map(int, b)) # Словарь из цифр второго числа  
 counter = 0 # Счётчик  
 for i in range(-1, ~(self.bigger\_digit + self.minichecker(self.bigger\_digit)), -1): # Цикл в обратном порядке  
 if b1[i] == 0: # Если 0  
 all\_digits.append(0) # В словарь записываем 0  
 counter += 1 # Счётчик растет  
 elif b1[i] == 1: # Если 1  
 a2 = int(a) \* (10 \*\* counter) # Умножаем множимое на 10 в степени сноса влево  
 all\_digits.append(a2) # Заносим итог в словарь  
 counter += 1 # Счётчик растет  
 for i in range(len(all\_digits) - 1): # Удаляем из словаря все лишние 0  
 if all\_digits[i] == 0:  
 del all\_digits[i]  
 all\_digits1 = list(map(str, all\_digits)) # Делаем еще один словарь из предыдущего  
 answer = 0  
 for i in all\_digits1:  
 answer += int(i, 2) # Складываем элементы словаря  
 answer = self.binarconverter(answer) # Возвращаем итог  
 return answer  
  
 def minichecker(self, a): # Счётчик последовательности,для определения длинны и формулы множимого и множителя  
 b = 3  
  
 for i in range(1, 9):  
 b -= 1  
 if i == a:  
 return b  
  
 def bigger(self, a, b): # Какое число больше  
 return len(a) if len(a) > len(b) else len(b)  
  
  
first = input("Введите множимое:\n")  
second = input("Введите множитель:\n")  
a = Digits(first, second) # Объявляю объект класса  
print(a) # Вывод класса на экран(выведет метод для коррекции вывода класса)

# Скріншоти:

