**Отчет**

Выполнил студент Шакирзянов Руслан 4214

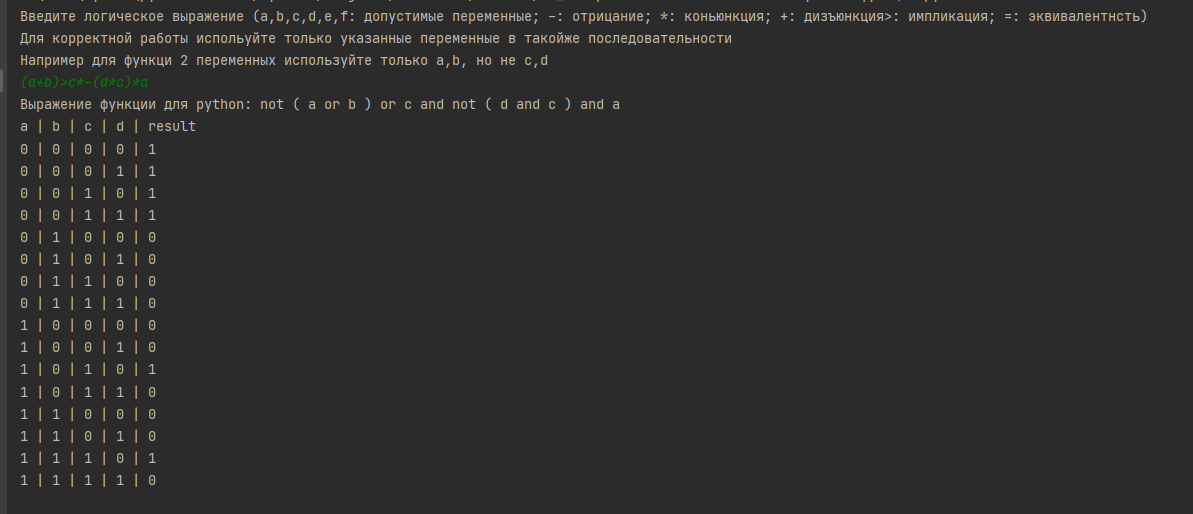
**Задача**

Построение таблицы истинности для произвольной функции заданной как строка.

**Процедурное решение**

На вход дается произвольная функция в указанном формате. Переводим эту функцию в формат логического выражения в python заменяя условные символы на команды python. Находим количество использованных в функции переменных. Используя стандартную функцию eval исполняем полученную строку и получаем значения True или False, полученное значение записываем в список результатов. Полученный список передаем в функцию из прошлой программы и получаем табличное представление данной функции.

**Пример работы программы**



**Код программы**

from math import log2  
  
  
def ToPython(txt\_func): # функция преобразования фулевой функции в читаемую питоном строку длф функции eval  
 py\_txt\_func = []  
 for i in txt\_func:  
 if i == '-':  
 py\_txt\_func.append('not') # замена - на not  
 elif i == '\*':  
 py\_txt\_func.append('and') # замена \* на and  
 elif i == '+':  
 py\_txt\_func.append('or') # замена + на or  
 elif i == '>':  
 if py\_txt\_func[-1] != ')':  
 py\_txt\_func.insert(-1, 'not') # замена > на not x or, с поддержкой скобок  
 else:  
 py\_txt\_func.insert(len(py\_txt\_func) - py\_txt\_func[::-1].index('(') - 1, 'not')  
 py\_txt\_func.append('or')  
 elif i == '=':  
 py\_txt\_func.append('==') # замена = на ==  
 else:  
 py\_txt\_func.append(i) # не заменяет скобки и переменные  
 return py\_txt\_func  
  
  
def TableOut(res\_list):  
 n\_val = round(log2(len(res\_list)))  
 header = 'abcdef'[:n\_val]  
 print(' | '.join(i for i in header) + ' | result')  
 for i in range(len(res\_list)):  
 print(' | '.join(str(i) for i in bin(i)[2:].rjust(n\_val, '0')) + f' | {res\_list[i]}')  
  
  
def main():  
 txt\_func = input("Введите логическое выражение "  
 "(a,b,c,d,e,f: допустимые переменные; -: отрицание; \*: коньюнкция; +: дизъюнкция"  
 ">: импликация; =: эквивалентнсть)\n"  
 "Для корректной работы испольуйте только указанные переменные в такойже последовательности\n"  
 "Например для функци 2 переменных используйте только a,b, но не c,d\n") # ввод функции с пояснениями  
 py\_txt\_func = ' '.join(ToPython(txt\_func)) # преобразование списка в строку  
 n\_val = sum([1 if txt\_func.count(i) else 0 for i in 'abcdef']) # подсчет количества использованных переменных  
 res\_list = list() # результирующий список  
 val = [False, True] # список заначений кахждой переменной  
 for a in val:  
 if n\_val == 1:  
 res = eval(py\_txt\_func) # исполнение строки как кода python функцией eval  
 res\_list.append(int(res))  
 continue  
 for b in val:  
 if n\_val == 2:  
 res = eval(py\_txt\_func)  
 res\_list.append(int(res))  
 continue  
 for c in val:  
 if n\_val == 3:  
 res = eval(py\_txt\_func)  
 res\_list.append(int(res))  
 continue  
 for d in val: # куча циклов для всех возможных 6 переменных  
 if n\_val == 4:  
 res = eval(py\_txt\_func)  
 res\_list.append(int(res))  
 continue  
 for e in val:  
 if n\_val == 5:  
 res = eval(py\_txt\_func)  
 res\_list.append(int(res))  
 continue  
 for f in val:  
 if n\_val == 6:  
 res = eval(py\_txt\_func)  
 res\_list.append(int(res))  
  
 print('Выражение функции для python:', py\_txt\_func)  
 TableOut(res\_list)  
  
  
main()  
"""  
пример  
(a+b)>c\*-(d\*c)\*a  
"""