**+Дисциплина «Информатика»**

**Рабочая тетрадь № 6**

|  |
| --- |
| Релейно-контактная схема представляет собой схематическое изображение некоторого устройства, состоящего из переключателей, соединяющих их проводников, входов в схему и выходов из нее. В релейно-контактной схеме любой переключатель имеет только два состояния: замкнутое и разомкнутое. |

|  |  |
| --- | --- |
| **1. Теоретический материал** | |
| Простейшая схема состоит из одного переключателя X, одного входа и одного выхода. Переключателю X ставится в соответствие высказывание x – «Переключатель X замкнут». Если x истинно, то схема проводит электрический ток, в противном случае – не проводит. Если принимать во внимание только значение высказывания, то любому высказыванию x можно поставить в соответствие следующую двухполюсную релейно-контактную схему    Отрицание высказывания x будем изображать двухполюсной схемой    Конъюнкцию двух высказываний x и y можно представить двухполюсной схемой с последовательным соединением переключателей X и Y    а дизъюнкцию – двухполюсной схемой с их параллельным соединением    Любая формула алгебры логики может быть представлена в нормальной форме, следовательно, любой формуле алгебры логики можно поставить в соответствие некоторую релейно-контактную схему, и наоборот, каждой релейно-контактной схеме можно поставить в соответствие некоторую формулу алгебры логики. | |
| **2. Пример** | |
| ***Задача:*** | |
|  | Составить релейно-контактную схему, соответствующую формуле. |
| ***Решение:*** | |
|  | Упростим формулу:    Релейно-контактная схема, соответствующая данной формуле, будет иметь следующий вид: |
| ***Задача:*** | |
|  | Упростить релейно-контактную схему: |
| ***Решение:*** | |
|  | Составим формулу, соответствующую данной схеме, и упростим ее:    Упрощенная схема выглядит следующим образом: |
| ***Задача:*** | |
|  | Составить релейно-контактную схему, соответствующую импликации. |
| ***Решение:*** | |
|  |  |
| ***Задача:*** | |
|  | Составить релейно-контактную схему, соответствующую следующей формуле: |
| ***Решение:*** | |
|  | Упростим формулу с помощью равносильных преобразований, затем составим релейно-контактную схему |
| ***Задача:*** | |
|  | Построить релейно-контактную схему, соответствующую *f(x, y, z)*, если известно, что:*f(1, 0, 0) = f(0, 1, 1) = f(0, 1, 0) = f(0, 0, 0) = 1.*  Остальные значения функции равны нулю. |
| ***Решение:*** | |
|  | Запишем аналитическое выражение, упростим его, после чего составим релейно-контактную схему. |
| ***Задача:*** | |
|  | Упростить релейно-контактную схему: |
| ***Решение:*** | |
|  | Запишем формулу алгебры логики, соответствующую приведенной схеме, упростим с помощью равносильных преобразований, после этого составим схему, соответствующую упрощенной формуле. |
| ***Задача:*** | |
|  | Доказать равносильность следующих релейно-контактных схем: |
| ***Решение:*** | |
|  | Запишем формулы алгебры логики, соответствующие каждой паре релейно-контактных схем, докажем их равносильность |

|  |  |  |
| --- | --- | --- |
| **3. Задания** | | |
| 1. | ***Задача:*** | |
|  | Составить релейно-контактную схему, соответствующую эквивалентности. |
| ***Решение:*** | |
|  |  |
| 2. | ***Задача:*** | |
|  | Составить релейно-контактные схемы, соответствующие следующим формулам: |
| ***Решение:*** | |
|  | 1)  2) |
| 3. | ***Задача:*** | |
|  | Построить релейно-контактные схемы, соответствующие *fi*(*x*, *y*,*z*), если известно, что:    Остальные значения функций *fi*(*x*, *y*,*z*) равны нулю. |
| ***Решение:*** | |
|  | |  |  |  |  | | --- | --- | --- | --- | | x | y | z | F | | 0 | 0 | 0 | 0 | | 0 | 0 | 1 | 0 | | 0 | 1 | 0 | 0 | | 1 | 0 | 0 | 0 | | 0 | 1 | 1 | 0 | | ***1*** | ***1*** | ***0*** | ***1*** | | ***1*** | ***0*** | ***1*** | ***1*** | | 1 | 1 | 1 | 0 |   1)  2)   |  |  |  |  | | --- | --- | --- | --- | | x | y | z | F | | 0 | 0 | 0 | 0 | | 0 | 0 | 1 | 0 | | ***0*** | ***1*** | ***0*** | ***1*** | | 1 | 0 | 0 | 0 | | 0 | 1 | 1 | 0 | | 1 | 1 | 0 | 0 | | ***1*** | ***0*** | ***1*** | ***1*** | | ***1*** | ***1*** | ***1*** | ***1*** | |

|  |  |  |
| --- | --- | --- |
| 4. | ***Задача:*** | |
|  | Упростить релейно-контактную схему: |
| ***Решение:*** | |
|  |  |
| 5. | ***Задача:*** | |
|  | Доказать равносильность следующих релейно-контактных схем: |
| ***Решение:*** | |
|  |  |

|  |
| --- |
| **1. Теоретический материал** |
| Любое вычислительное устройство состоит из набора так называемых «вентилей» – минимальных логических устройств, способных преобразовывать входные сигналы в выходные. Ниже в таблице 1 представлены основные обозначения логических вентилей в различных стандартах.  Таблица 1 – Логические вентили и их условные графические обозначение в стандартах «ГОСТ 2.743-91», «IEC 60617-12: 1997» и «US ANSI 91-1984»   |  |  |  |  | | --- | --- | --- | --- | | **Название вентиля** | **Условные графические обозначения (УГО)** | | | | **ГОСТ 2.743-91** | **IEC 60617-12: 1997** | **US ANSI 91-1984** | | **НЕ** | NOT gate RU.svg | IEC NOT label.svg | Not-gate-en.svg | | **И** | AND gate RU.svg | IEC AND label.svg | Logic-gate-and-us.svg | | **ИЛИ** | OR gate RU.svg | IEC OR label.svg | Or-gate-en.svg | | **НЕ И** | NAND gate RU.svg | IEC NAND label.svg | Nand-gate-en.svg | | **НЕ ИЛИ** | NOR gate RU.svg | IEC NOR label.svg | Nor-gate-en.svg | | **исключающее ИЛИ** | XOR gate RU.svg | IEC XOR label.svg | Xor-gate-en.svg |   Маленькие кружочки в УГО обозначают инверсию сигнала (т.е. отрицание НЕ). Они могут стоять и на входе вентиля, обозначая инверсию входного сигнала.  Рассмотрим первое простое устройство, которое можно собрать на вентилях: **полусумматор**. Его таблица истинности и схема представлена на рисунке 1.    Рисунок 1 – Таблица истинности для сложения одноразрядных чисел  и схема полусумматора  Полусумматор предназначен для нахождения суммыдвух бит. При необходимости сложения чисел, состоящих из двух и более разрядов, эта схема уже не работает (нельзя просто взять необходимое число полусумматоров, т.к. в этой схеме не предусмотрен учёт переноса из младшего разряда в старший). Для этого нужно уже соединять между собой так называемые **полные сумматоры**. Таблица истинности и схема полного сумматора представлена на рисунке 2.    Рисунок 2 – Таблица истинности для полного сумматора (а);  схема для полного сумматора (б)  В вычислительной технике также можно встретить применение **мультиплексоров**. Мультиплексор иметь один выход и две группы входов: информационные и адресные. Код, подаваемый на адресные входы, определяет, какой из информационных входов в данный момент подключен к выходному выводу. Поскольку n-разрядный двоичный код может принимать 2n значений, то если число адресных входов мультиплексора равно n, число его информационных входов должно равняться 2n. УГО мультиплексора с 8 информационными входными линиями , 4 адресными входными линиями и 1 выходной линией приведено на рисунке 3.    Рисунок 3 – Мультиплексор 8 в 1  Используя мультиплексор можно реализовывать логические функции (пример ниже). |

|  |  |  |
| --- | --- | --- |
| **2. Пример** | | |
| 1. | ***Задача:*** | |
|  | Нарисовать схему (от руки или в любом графическом редакторе), реализующую следующую логическую функцию: |
| ***Решение:*** | |
|  | C:\Users\Qverty\g3488.png |
| 2. | ***Задача:*** | |
|  | Записать логическую формулу (формулы) и таблицу истинности для следующей схемы:  C:\Users\Qverty\g3488-6.png |
| ***Решение:*** | |
|  | **,**     |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **A** | **B** |  |  |  |  |  |  | | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | |

|  |  |  |
| --- | --- | --- |
| 3. | ***Задача:*** | |
|  | Нарисовать схему (от руки или в любом графическом редакторе), позволяющую суммировать трёхбитовые двоичные числа (, ,и C). |
| ***Решение:*** | |
|  | C:\Users\Qverty\rect2970-2.png |
| 4. | ***Задача:*** | |
|  | Используя мультиплексор, нарисовать схему (от руки или в любом графическом редакторе), реализующую функцию большинства от трёх логических переменных. |
| ***Решение:*** | |
|  | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Таблица истинности для функции большинства** | | | | **Схема, реализующая необходимую логическую функцию** | | **A** | **B** | **C** | **F** |  | | 0 | 0 | 0 | 0 | | 0 | 0 | 1 | 0 | | 0 | 1 | 0 | 0 | | 0 | 1 | 1 | 1 | | 1 | 0 | 0 | 0 | | 1 | 0 | 1 | 1 | | 1 | 1 | 0 | 1 | | 1 | 1 | 1 | 1 | |

|  |  |  |
| --- | --- | --- |
| **3. Задания** | | |
| 1. | ***Задача:*** | |
|  | Нарисовать схему (от руки или в любом графическом редакторе), реализующую следующую логическую функцию: |
| ***Решение:*** | |
|  |  |
| 2. | ***Задача:*** | |
|  | Записать логическую формулу (формулы) и таблицу истинности для следующей схемы:  C:\Users\Qverty\g3907.png |
| ***Решение:*** | |
|  | 1. ¬ ( (¬A B) & (A⋁B) ) ⋁ A = F0 2. ¬ (B ⋁ A) = ¬A &¬B = F1  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | A | B | ¬A | ¬A⊕B | A∨B | (¬A⊕B)∧(A∨B) | ¬ (¬A⊕B)∧(A∨B) | F0 | | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |  |  |  |  |  | | --- | --- | --- | --- | | A | B | B∨A | F1 | | 0 | 0 | 0 | 1 | | 0 | 1 | 1 | 0 | | 1 | 0 | 1 | 0 | | 1 | 1 | 1 | 0 | |
| 3. | ***Задача:*** | |
|  | Используя мультиплексор, нарисовать схему (от руки или в любом графическом редакторе), реализующую функцию импликации. |
| ***Решение:*** | |
|  | |  |  |  |  | | --- | --- | --- | --- | | A | B | F |  | | 0 | 0 | **1** | D0 | | 0 | 1 | **1** | D1 | | 1 | 0 | **0** | D2 | | 1 | 1 | **1** | D3 | |
| 4. | ***Задача:*** | |
|  | Нарисовать схему (от руки или в любом графическом редакторе), позволяющую умножать двухбитовые двоичные числа. |
| ***Решение:*** | |
|  | , где , ,  https://cdn.discordapp.com/attachments/768376607722569740/780405262505869332/unknown.png |

|  |  |  |
| --- | --- | --- |
| **Тест** | | |
| 1. | Для сложения многоразрядных двоичных чисел используется последовательность соединённых между собой: | |
|  | а) мультиплексоров; б) триггеров;  в) полусумматоров; г) полных сумматоров. |
| *Ответ:* | |
|  | **г** |
| 2. | Устройство, выполняющее базовые логические операции, называется: | |
|  | а) реестр; б) ячейка; в) вентиль; г) триггер |
| *Ответ:* | |
|  | **в** |
| 3. | Графическое изображение логического выражения называется: | |
|  | а) схема б) рисунок в) чертеж г) график |
| *Ответ:* | |
|  | **а** |
| 4. | Электронная схема, применяемая в регистрах компьютера для запоминания одного разряда двоичного кода это … | |
|  | а) вентиль б) логическая схема  в) триггер г) электронная схема |
| *Ответ:* | |
|  | **в** |
| 5. | Мультиплексор – это устройство для ... | |
|  | а) умножения чисел  б) коммутации одного из информационных входов с единственным выходом на основе кода, подаваемого на адресные входы  в) произведения операции сдвига на основе данных на информационных входах и кода на адресных входах  г) хранения одного бита информации |
| *Ответ:* | |
|  | **б** |
| 6. | Какому логическому выражению соответствует схема на рисунке? | |
|  | а) F = A б) F = A& Ā  в) F = AV Ā г) F = Ā |
|  | *Ответ:* | |
|  | **г** |
| 7. | Какой минимально необходимо взять мультиплексор, чтобы реализовать заданную таблицей истинности логическую функцию от 4 логических переменных? | |
|  | а) с 4 информационными входными линиями, 2 адресными входными линиями и 1 выходной линией  б) с 16 информационными входными линиями, 4 адресными входными линиями и 1 выходной линией  в) с 4 информационными входными линиями, 4 адресными входными линиями и 2 выходными линиями  г) с 4 информационными входными линиями, 16 адресными входными линиями и 2 выходными линиями |
| *Ответ:* | |
|  | **б** |
| 8. | Такой структурной схемой обозначается  AND gate RU.svg | |
|  | а) конъюнктор б) дизъюнктор  в) бинарное НЕ г) инвертор |
| *Ответ:* | |
|  | **а** |
| 9. | Равносильны или нет следующие релейно-контактные схемы? | |
|  | https://fsd.videouroki.net/html/2011/03/31/98657307/98657307_14.png |
| *Ответ:* | |
|  | **нет** |
| 10. | Какому логическому выражению соответствует схема на рисунке? | |
|  | а) F=не(A & B) б) F=не(A & неB)  в) F=не(неA & неB) г) F=не(неA & B) |
| *Ответ:* | |
|  | **г** |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| г | в | а | в | б | г | б | а | нет | г |

**Реализация задач на языке программирования Python**

|  |
| --- |
| Массивы в языке Python реализованы в виде следующих конструкций: строки, списки, кортежи, множества и словари. Эти конструкции достаточно мощные и выразительные, что обеспечивает эффективность использования языка Python. |

|  |
| --- |
| **1. Теоретический материал** |
| Строки - это последовательности символов. Создать строки можно следующим образом:  s = "Hello, World!"  p = 'Миру - мир!'  Доступ к символу с индексомn(начиная с нуля!) осуществляется с помощью выраженияs[n], подстроку можно выделить, указав интервалs[n:m]вот пример:  s = "Мама мыла раму"  print(s[6])  p = s[5:9]  print(p)  Если индекс отрицательный, то отсчет идет от конца строки. При этом индекс -1 соответствует последнему символу. Хотя к символам строки можно обращаться через индекс, но при этом нельзя изменять символы строки.  Поскольку строка задается с помощью кавычек (двойных или одинарных), то для того, чтобы в самой строке использовать символы кавычек, их необходимо экранировать следующим образом:  s = "А Балда приговаривал с укоризной: \"Не гонялся бы ты, поп, за дешевизной\""  print(s)  В строках можно использовать специальные символы, которые задается следующим образом:   |  |  |  | | --- | --- | --- | | \n |  | новая строка | | \t |  | табуляция |   Поэтому более красивой будет следующий вывод:  s = "\n\n\t\tА Балда приговаривал с укоризной:\n \  \t\t\"Не гонялся бы ты, поп, за дешевизной\"\n\n"  print(s)  Со строками можно выполнять операции конкатенации с помощью оператора "+", который мы уже проходили. Строку можно умножить на целое число**n**, в результате чего строка будет повторена**n**раз:  s = "Мама"  s = s \* 3  print(s)  Узнать длину строки можно с помощью функции**len(s)**:  s = "Жили у Бабуси два веселых гуся"  n = s.find("два")  print(n)  Часто возникает задача поиска подстроки в строке. Для этого используем функцию**s.find(str)**:  s = "Мама|Папа|Я"  S = s.split('|')  print(S[1])  Полный список функций для строк:   * <https://pythonworld.ru/tipy-dannyx-v-python/stroki-funkcii-i-metody-strok.html> |

|  |  |
| --- | --- |
| 2. Пример | |
| ***Задача:*** | |
|  | Дана строка. Удалить из нее все пробелы |
| ***Решение (код программы):*** | |
|  | txt = "I like bananas"  x = txt.replace(" ", "")  print(x) |

|  |  |  |
| --- | --- | --- |
| **3. Задания** | | |
| 1. | ***Задача:*** | |
|  | Проверить является ли строка палиндромом. **Палиндромом** называется слово, которое одинаково читается как слева направо, так и справа налево, например, в английском языке такими словами являются "radar" и "racecar". |
| ***Решение (код программы):*** | |
|  | s = input()  ans = s[ : : -1]  if s == ans:  print('true')  else:  print('no') |
| 2. | ***Задача:*** | |
|  | Преобразовать строку к нижнему регистру |
| ***Решение (код программы):*** | |
|  | s = input()  ans = s.lower()  print(ans) |
| 3. | ***Задача:*** | |
|  | Преобразовать строку к верхнему регистру |
| ***Решение (код программы):*** | |
|  | s = input()  ans = s.upper()  **print(ans)** |
| 4. | ***Задача:*** | |
|  | Проверить состоит ли строка из цифр |
| ***Решение (код программы):*** | |
|  | s = input()  if s.isdigit():  print('true')  else:  print('no') |
| 5. | ***Задача:*** | |
|  | Дана непустая строка S. Вывести строку, содержащую символы строки S, между которыми вставлено по одному пробелу. |
| ***Решение (код программы):*** | |
|  | s = input()  **ans = ''**  **for i in range(len(s)):**  **if s[i] != C ':**  **ans += s[i] + ' '**  **print(ans)** |
| 6. | ***Задача:*** | |
|  | Дан символ C. Вывести его код (то есть номер в кодовой таблице). |
| ***Решение (код программы):*** | |
|  | s = input()  if len(s) == 1:  print(ord(s))  else:  print('wrong symbol') |

|  |
| --- |
| **1. Теоретический материал** |
| По сути списки представляют собой массивы различных объектов, с которыми можно работать разнообразными функциями. Создать список можно следующими способами  A = [] # пустой список  B = [1, 1, 2, 3, 5, 8, 13, 21] # числа Фибоначчи  C = ["мама", 120, 'z', "список"] # смешанный список  D = [[1, 2], ['a', 'b', 'c'], [[3.14, "abc"], [120, 140]]] # список списков  К элементам списка можно обращаться через индексы, которые начинаются с нуля. В отличии от строк, элементы списков можно заменять:  A = [1, 2, 3]  A[1] = "новое значение"  print(A)  Для работы со списками существует большое количество методов. Вот основные:   |  |  |  | | --- | --- | --- | | list.append(x) |  | добавляет элемент в конец списка | | list.insert(i, x) |  | вставляет на i-ый элемент значение x | | list.count(x) |  | возвращает количество элементов со значением x | | list.reverse() |  | разворачивает список | | list.clear() |  | очищает список |   Рассмотрим понятие кортежа в Python. Кортеж очень напоминает список, но в отличии от списков кортежи нельзя изменить после создания, за счет чего достигается более эффективная реализация кортежей. Создать кортеж можно следующим образом:  A = (“Ivanov", “Ivan", 1988)  Surname, Name, Date = A  print(Surname + " " + Name + " " + Date.\_\_str\_\_()) |

|  |  |
| --- | --- |
| 2. Пример | |
| ***Задача:*** | |
|  | Заполните список случайными числами. Используйте в коде цикл for, функции range() и randint(). |
| ***Решение (код программы):*** | |
|  | import random  a = []  for i in range(1,10):  a.append(random.randint(5,10))  print(a) |
| ***Задача:*** | |
|  | Напишите программу, которая запрашивает с ввода восемь чисел, добавляет их в список. На экран выводит их сумму, максимальное и минимальное из них. Для нахождения суммы, максимума и минимума воспользуйтесь встроенными в Python функциями sum(), max() и min(). |
| ***Решение (код программы):*** | |
|  | l = []  for i in range(0,8):  a = int(input())  l.append(a)  print(l)  print(sum(l))  print(min(l))  print(max(l)) |

|  |  |  |
| --- | --- | --- |
| **3. Задания** | | |
| 1. | ***Задача:*** | |
|  | Если объект range (диапазон) передать встроенной в Python функции list(), то она преобразует его к списку. Создайте таким образом список с элементами от 0 до 100 и шагом 17. |
| ***Решение (код программы):*** | |
|  | **a = [i for i in range(0, 100, 17)]**  print(a) |
| 2. | ***Задача:*** | |
|  | Напишите программу, которая заполняет список пятью словами, введенными с клавиатуры, измеряет длину каждого слова и добавляет полученное значение в другой список. Например, список слов – ['yes', 'no', 'maybe', 'ok', 'what'], список длин – [3, 2, 5, 2, 4]. Оба списка должны выводиться на экран. |
| ***Решение (код программы):*** | |
|  | **s = input()**  **s = s.split(' ')**  **c = []**  **for i in range(len(s)):**  **c.append(len(s[i]))**  **print(s)**  print(c) |

|  |
| --- |
| **1. Теоретический материал** |
| Теперь рассмотрим множества в Python. Множества отличаются от списков и кортежей тем, что хранящиеся в них объекты являются неупорядоченными. Кроме того, множества содержат только уникальные элементы. Зато с множествами можно производить стандартные операции объединения, пресечения и разности  Кроме того, множества содержат только уникальные элементы. Зато с множествами можно производить стандартные операции объединения, пресечения и разности   |  |  |  | | --- | --- | --- | | | |  | объединение | | & |  | пересечение | | - |  | разность | | ^ |  | симметричная разность |   создается множество с помощью оператора **set()**:  A = set(['a', "apple", 120])  Добавить один элемент в множество можно с помощью метода**A.add(x)**, чтобы добавить другое множество используется операция**A.update([])**, для удаления элементов метод**A.remove(x)**. Приведем пример работы с множествами:  A = set(["apple", "orange"])  A.add("pear")  B = set(["apple", "pear", "banana"])  C = A | B  D = A&B  print(C)  print(D)  Элементы списков и кортежей индексируются натуральными числами от 0 до N-1, где N - количество элементов в массиве, но часто к элементам массивам удобнее обращаться по номерам. Для этого в Python есть конструкция, которая называется словарем. Покажем словарь на примере:  people = {  "Name": “Ivan”,  "Surname": “Ivanov",  "Date": 1988  }  print(people)  people["Name"] = “Alex"  people["Date"] = 2008  print(people) |

|  |  |  |
| --- | --- | --- |
| **2. Задания** | | |
| 1. | ***Задача:*** | |
|  | Даны два множества X и Y. Найти:  1) их объединение  2) их пересечение  3) элементы множества A, которых нет во множестве B |
| ***Решение (код программы):*** | |
|  | **x = input()**  **y = input()**  **x = x.split(' ')**  **y = y.split(' ')**  **a = set(x)**  **b = set(y)**  **print('1 ', a|b)**  **print('2 ', a&b)**  print('3 ', a-b) |