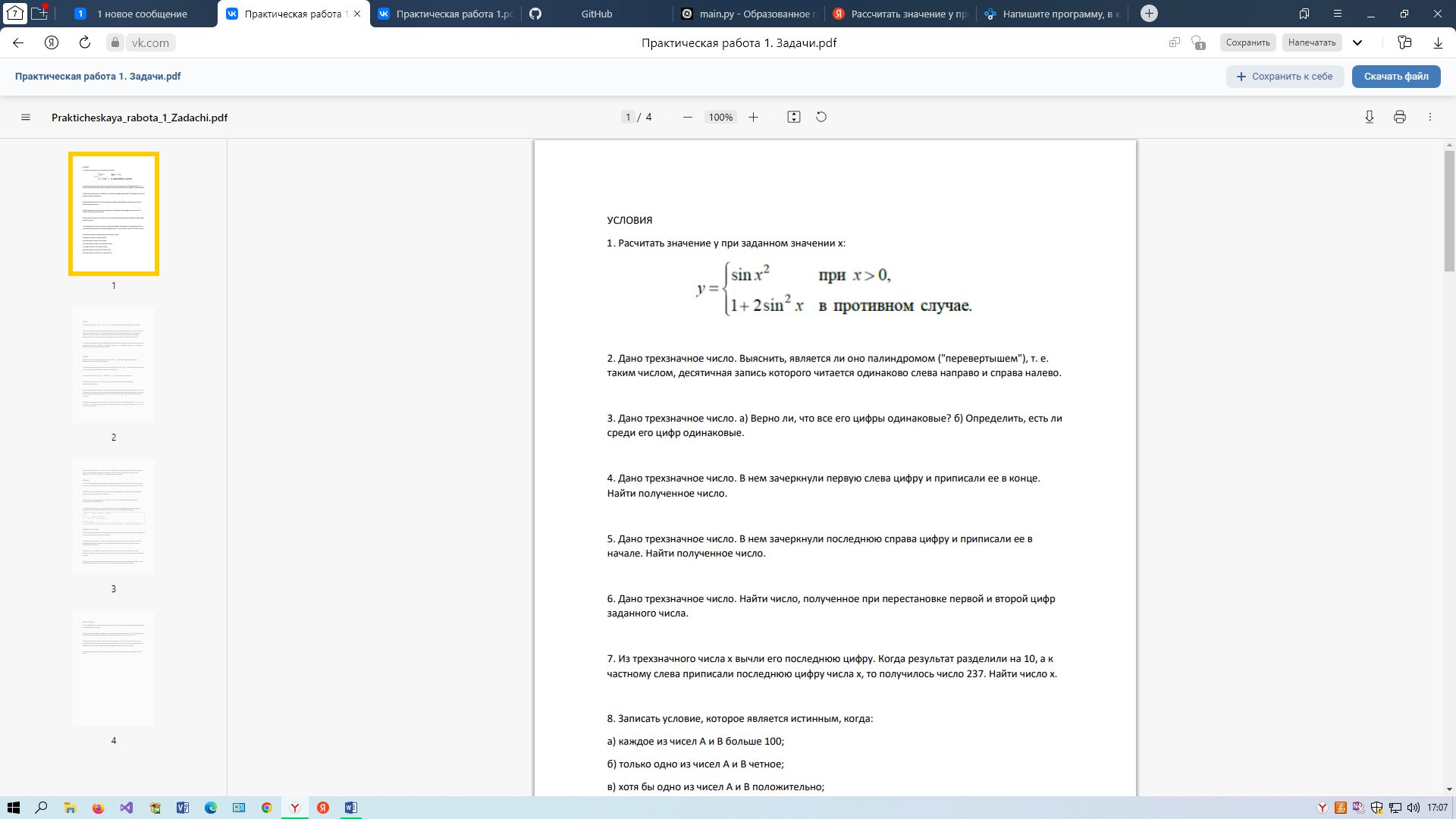
**Практическая работа №1**

**ИЗУЧЕНИЕ И НАСТРОЙКА СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ**

**Цель работы:** научиться работать и настраивать систему контроля версий.

Листинг 1.

|  |
| --- |
| from math import sin      x = float(input())    if x > 0:  y = sin(x)\*\*2  else:  y = 1+2\*sin(x)\*\*2    print(y) |

2. Дано трехзначное число. Выяснить, является ли оно палиндромом ("перевертышем"), т. е. таким числом, десятичная запись которого читается одинаково слева направо и справа налево.

Листинг 2.

|  |
| --- |
| n=input("Введите целое трехзначное число: ")  if n==n[2]+n[1]+n[0]:  print("Палиндром")  else:  print("Не палиндром") |

3. Дано трехзначное число. а) Верно ли, что все его цифры одинаковые? б) Определить, есть ли среди его цифр одинаковые.

Листинг 3.

|  |
| --- |
| n=int(input('Введите целое трехзначное число: '))  n1 = n // 100  n2 = n // 10 % 10  n3 = n % 10  a) if n1==n2==n3:  print('Да')  else:  print('Нет')  б) if n1 == n2 or n1 == n3 or n2 == n3:  print('Да')  else:  print('Нет') |

4. Дано трехзначное число. В нем зачеркнули первую слева цифру и приписали ее в конце. Найти полученное число.

Листинг 4.

|  |
| --- |
| n = int(input('Введите целое трехзначное число:'))  print(n // 100 + n % 100 \* 10) |

5. Дано трехзначное число. В нем зачеркнули последнюю справа цифру и приписали ее в начале. Найти полученное число.

Листинг 5.

|  |
| --- |
| n = int(input('Введите целое трехзначное число:'))  x = n // 10 + n % 10 \* 100  print(x) |

6. Дано трехзначное число. Найти число, полученное при перестановке первой и второй цифр заданного числа.

Листинг 6.

|  |
| --- |
| n = (i for i in input('Введите целое трехзначное число:'))  try :  a = (next(n) + next(n))[::-1]  while n :  a += next(n)  except StopIteration :  print(int(a)) |

7. Из трехзначного числа x вычли его последнюю цифру. Когда результат разделили на 10, а к частному слева приписали последнюю цифру числа x, то получилось число 237. Найти число x.

Листинг 7.

|  |
| --- |
| print ('Введите целое трехзначное число')  n=int(input())  c=n//100  n=(n%100)\*10+c  print (n, '<- исходное число Х') |

8. Записать условие, которое является истинным, когда:

а) каждое из чисел А и В больше 100;

б) только одно из чисел А и В четное;

в) хотя бы одно из чисел А и В положительно;

г) каждое из чисел А, В, С кратно трем;

д) только одно из чисел А, В и С меньше 50;

е) хотя бы одно из чисел А, В, С отрицательно

Листинг 8.

|  |
| --- |
| а) if (a>100) and (b>100)  б) if ((a mod 2 = 0) and (b mod 2 <>0)) or ((a mod 2 <> 0) and (b mod 2 =0))  в) if (a>0) or (b>0)  г) if (a mod 3 = 0) and (b mod 3 = 0) and (c mod 3 = 0)  д) if ((a<50) and (b>=50) and (c>=50)) or ((a>50) and (b<50) and (c>=50)) or ((a>=50) and (b>=50) and (c<50))  е) if (a<0) or (b<0) or (c<0) |

9. Вычислить сумму 1! 2! 3! ... n!, k! 1 2 3 ... k (значение n вводится с клавиатуры; 1<n≤10).

Листинг 9.

|  |
| --- |
| n = int(input('Введите число от 1 до 10:'))  N = 1  sum = 0  for a in range(1, n+1):  for b in range(1, a+1):  N = N \* b  sum += N  print(sum) |

10. В некоторой стране используются денежные купюры достоинством в 1, 2, 4, 8, 16, 32 и 64. Дано натуральное число n. Как наименьшим количеством таких денежных купюр можно выплатить сумму n (указать количество каждой из используемых для выплаты купюр)? Предполагается, что имеется достаточно большое количество купюр всех достоинств.

Листинг 10.

|  |
| --- |
| n = int(input())  money = []  money\_range = sorted([1, 2, 4, 8, 16, 32, 64], reverse=True)  for i in range(0, len(money\_range)):  if n // money\_range[i] != 0:  num = n // money\_range[i]  for j in range(0, num):  money.append(money\_range[i] )  n -= money\_range[i]  else:  continue  print(money) |

11. Старинная задача. Имеется 100 рублей. Сколько быков, коров и телят можно купить на все эти деньги, если плата за быка — 10 рублей, за корову — 5 рублей, за теленка — полтинник (0,5 рубля) и надо купить 100 голов скота?

Листинг 11.

|  |
| --- |
| for b in range (1, 100//10+1):  for k in range(1, 100//5+1):  t = 100-b-k  if b\*10+k\*5+t\*0.5 == 100:  print("быки:", b, "коровы:", k, "телята:", t); |

12. Дан список a. Определить сумму a1-a2+a3-a4+… Условный оператор и операцию возведения в степень не использовать.

Листинг 12.

|  |
| --- |
| a=[1,2,3,4,5,6,7,8,9]  try:  Sum\_even = 0  for i in range(0,len(a)+1,2):  Sum\_even+=a[i]  except IndexError:  pass  try:  Sum\_odd = 0  for j in range(1,len(a)+1,2):  Sum\_odd+=a[j]  except IndexError:  pass  print(Sum\_even - Sum\_odd) |

13. Переставьте соседние элементы списка (А[0] c A[1], A[2] с A[3], …) Если элементов нечетное число, то последний элемент останется на свое месте.

Листинг 13.

|  |
| --- |
| a = [int(i) for i in input().split()]  for i in range(1, len(a), 2):  a[i - 1], a[i] = a[i], a[i - 1]  print(' '.join([str(i) for i in a])) |

14. Создать список 11, 22, 33, …, 1010, 1111, … , nn. Использовать генератор.

Листинг 14.

|  |
| --- |
| def is\_palindrome(num):  # Skip single-digit inputs  if num // 10 == 0:  return False  temp = num  reversed\_num = 0  while temp != 0:  reversed\_num = (reversed\_num \* 10) + (temp % 10)  temp = temp // 10  if num == reversed\_num:  return num  else:  return False  for i in infinite\_sequence():  pal = is\_palindrome(i)  if pal:  print(pal) |

15. Дан список целых чисел. Напечатать: все четные элементы и все элементы, оканчивающиеся нулем

Листинг 15.

|  |
| --- |
| a = [int(s) for s in input().split()]  for element in a:  if element % 2 == 0:  print(element, end='')  def main(l):  print(\*(i for i in l if i[-1] == '0'))    input()  lst = input().split()    main(lst) |

16. Дан список имен гостей в том порядке, как они будут рассажены за праздничным столом. Посчитайте, сколько в нем пар гостей с одинаковыми именами будут сидеть рядом. Считается, что любые два подряд одинаковые имени гостя образуют одну пару, которую необходимо посчитать.

Листинг 16.

|  |
| --- |
| def counting(arr):  d=dict()  for element in arr:  if element in d:  d[element] += 1  else:  d[element] = 1  result = []  for key, value in d.items():  if value==1:  result.append(key)  return result  A=[1,2,3,3,4,2]  B=counting(A)  print(\*B) |

17. Дан список оценок мальчика Васи по русскому языку в числовом формате 5, 4, 5, 3, 2, 5, 4, 3, 5, 5, 4, 2, 2, 3. Посчитать средний балл ученика и вывести его оценки в формате «5; 4; 5; 3; 2; 5; 4; 3; 5; 5; 4; 2; 2; 3».

Листинг 17.

|  |
| --- |
| a = [5, 4, 5, 3, 2, 5, 4, 3, 5, 5, 4, 2, 2, 3]  print(f'{sum(a) / len(a)}')  print(\*a, sep=';') |

18. Дан список натуральных чисел от 1 до 20. Вывести список, генерирующийся по принципу: для четных элементов получить их квадраты, а для нечетных увеличить значение на 2. (Пример: [3, 4, 5, 16, 7, 36, 9, 64, …]). Использовать генератор.

Листинг 18.

|  |
| --- |
| i = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]  def gen(a):  for i in range(1,21):  while a < len[a+1]:  if a[i] % 2 == 0:  a[i] = a[i] \*\* 2  else:  a[i] += 2  print(gen()) |

19. Написать рекурсивную процедуру перевода натурального числа из десятичной системы счисления в N-ричную. Значение N в основной программе вводится с клавиатуры (2<=N<=16).

Листинг 19.

|  |
| --- |
| def conv(n, r):  if n < r:  return "0123456789ABCDEF"[n]  else:  k = n % r  return conv(n//r, r) + "0123456789ABCDEF"[k]  print(conv(12,2)) |

20. Дан список вещественных чисел. Используя lambda-функцию и функцию map выведите список остатков от деления этих чисел на 7.

Листинг20.

|  |
| --- |
| num = [1, 2, 3, 4, 5]  num2 = list(map(lambda x: x % 7, num))  print(num2) |

21. Дан список имен детей: ['катя', 'маша', 'таня', 'саша']. Сделайте так, чтобы имена начинались с заглавной буквы.

Листинг 21.

|  |
| --- |
| a = ['катя', 'маша', 'таня', 'саша']  print(a.title()) |

22. Попробуйте переписать следующий код через map и lambda-функцию. Он принимает список реальных имён и заменяет их прозвищами, используя надёжный метод.

Листинг 22.

|  |
| --- |
| names = ['Маша', 'Петя', 'Вася']  secret\_names = list(map(lambda x: hash(x), names))  print(secret\_names) |

23. Даны два множества чисел. Выведите третье, состоящее из элементов первого и элементов второго множества, которых нет в первом.

Листинг 23.

|  |
| --- |
| print(\*sorted(set(input().split()) & set(input().split()), key=int)) |

24. Дан текст, состоящий из строк. Определить количество слов в тексте. Словом считается последовательность символов, слова разделены пробелом или символом конца строки. Используйте множества.

Листинг 24.

|  |
| --- |
| words = set()  a = int(input())  for \_ in range(a):  words.update(input().split())  print(len(words)) |

25. Дан текст, состоящий из количества строк текста и самого текста. Вывести на экран построчно каждое слово и сколько раз оно встречается в тексте. Для реализации используйте словари.

Листинг 25.

|  |
| --- |
| from collections import defaultdict  words = "слово1 слово2 слово1 слово3 слово2 Вася слово1"  word\_list = words.split()  word\_count\_dict = defaultdict(int)  for word in word\_list:  print(word, word\_count\_dict[word])  word\_count\_dict[word] += 1 |

26. Дан текст, состоящий из количества строк текста и самого текста. Вывести на экран слово, которое встречается в тексте чаще всего. Для реализации используйте словари.

Листинг 26.

|  |
| --- |
| counter = {}  for i in range(int(input())):  line = input().split()  for w in line:  counter[w] = counter.get(w, 0) + 1  max\_count = max(counter.values())  most\_frequent = [k for k, v in counter.items() if v == max\_count]  print(min(most\_frequent)) |

27. Создайте файл, в котором записано два целых числа, каждое в отдельной строке. Выведите в выходной файл их сумму.

Листинг 27.

|  |
| --- |
| with open('') as f:  text = f.read()  print(sum(map(int, text.split(None, 2)[:2]))) |

28. Имеется входной файл и введенный с клавиатуры символ (например, “!”). Определить, есть ли данный символ в файле. Выведите с качестве результата слово “Yes” или ”No”.

Листинг 28.

|  |
| --- |
| sym = input()  with open('') as f:  if sym in f.read():  print('yes')  else:  print('no') |

29. Дан файл, каждая строка которого может содержать одно или несколько целых чисел, разделенных одним или несколькими пробелами. Вычислите сумму чисел в каждой строке и выведите эту сумму (для каждой строки выводится сумма чисел в этой строке).

Листинг 29.

|  |
| --- |
| with open('input.txt') as t:  print(\*(sum(map(int, line.split())) for line in t.readlines()), sep='\n') |

30. Создайте текстовый файл. Определите сколько в нем букв (латинского алфавита), слов, строк.

Листинг 30.

|  |
| --- |
| file = open('text.txt')  lines = 0  words = 0  symbols = 0  for line in file:  lines += 1  words += len(line.split())  symbols += len(line.strip('\n'))  print("Lines:", lines)  print("Words:", words)  print("Symbols:", symbols) |

**Контрольные вопросы**

1. Принцип работы системы контроля версий?

Система контроля версий — это система, записывающая изменения в файл или набор файлов в течение времени и позволяющая вернуться позже к определённой версии. Контроль версий можно использовать практически для любых типов файлов.

Программисты давным-давно разработали локальные СКВ с простой базой данных, которая хранит записи о всех изменениях в файлах, осуществляя тем самым контроль ревизий.

Были разработаны централизованные системы контроля версий (ЦСКВ). Такие системы, как CVS, Subversion и Perforce, используют единственный сервер, содержащий все версии файлов, и некоторое количество клиентов, которые получают файлы из этого централизованного хранилища. Применение ЦСКВ являлось стандартом на протяжении многих лет.

Распределённые системы контроля версий (РСКВ). В РСКВ (таких как Git, Mercurial, Bazaar или Darcs) клиенты не просто скачивают снимок всех файлов (состояние файлов на определённый момент времени) — они полностью копируют репозиторий. В этом случае, если один из серверов, через который разработчики обменивались данными, умрёт, любой клиентский репозиторий может быть скопирован на другой сервер для продолжения работы. Каждая копия репозитория является полным бэкапом всех данных.

Вывод: мы научились работать и настраивать систему контроля версий.