**Планы проведения ЛПЗ и методические рекомендации по выполнению заданий**

**ЛПЗ № 1. Ввод/вывод в программе Phyton.**

**План: Первая программа. Ввод и вывод информации на экран.**

**Задания.**

1. Определите (не оформляя программу), что будет выведено на экран в результате выполнения следующих инструкций:

print(31, 15, end = '')

print(77)

2. Определите (не оформляя программу), что будет выведено на экран в результате выполнения следующих инструкций:

print(51, 36)

print(77, 45, end = '')

3. Определите (не оформляя программу), что будет выведено на экран в результате выполнения следующих инструкций:

print(25, 86)

print()

print(27, 51, end = '')

4. Определите (не оформляя программу), что будет выведено на экран в результате выполнения следующих фрагментов программ:

a)

a = 5; b = 3

print('F(', b, ') = ',a, sep = '')

б)

a = 5; b = 3

print('f(a)=', '(b)', sep = '')

в)

a = 5; b = 3

print('F(', a, ')=(', b, ')', sep = '')

**ЛПЗ № 2. Типы данных в программировании. Определение переменной.**

**План: запись по правилам языка Python арифметические выражения. Инструкция присваивания, работа с переменными.**

**Задания.**

1. Запишите в одну строку по правилам языка Python следующие арифметические выражения:

а)

|  |  |
| --- | --- |
|  |  |
|  |  |

2. Запишите следующие выражения (представленные в линейной записи) в обычную форму:

а) a/b/c;

б) a + b/c;

в) (a + b)/c;

г) a\*b/(c + 2);

д) a/b\*(c – 3)/d;

е) (a/3 + b/2)/(b+ c/(2 + b)).

Задания 3–7, приведенные ниже, удобно выполнять с помощью таблицы, имитирующей «трассировку» программы. Например, если требуется определить значение переменной y после выполнения следующего фрагмента программы:

x = 15

x = x + 6

y = –x + 26

у = y – x

то такая таблица имеет вид:

|  |  |  |
| --- | --- | --- |
| Инструкция | x | y |
| x = 15 | 15 |  |
| x = x + 6 | 15 + 6 = 21 |  |
| y = –x + 26 | 21 | –21 + 26 = 5 |
| у = y – x | 21 | 5 – 21 = –16 |

Ответ: –16.

3. Определите значение переменной c после выполнения следующего фрагмента программы:

a = 5

a = a + 6

b = –a

c = a – 2 \* b

4. Определите значение переменной v после выполнения следующего фрагмента программы:

a = 40

b = 10

b = –а/2 \* b

v = b + a \* 2

5. Определите значения переменных s и k после выполнения следующего фрагмента программы:

s = 14

k = –3

d = s + 1

s = d

k = 2 \* s

6. Определите значения переменных p и q после выполнения следующего фрагмента программы:

p= 0

q = 30

d = q – 5

q = 2 \* d

p = q – 100

7. Вычислите значение переменной zпри x= 25 и y= 7:

а) z = x % y + y

б) z = x//y + x

в) y = x//y

z = x//y

г) y = x//y + y

z = x % y + x

д) y = x % y + 4

z = x % y + 1

е) y = x//y

z = x % (y + 1)

ж) y = x % y

z = x//(y + 1)

**ЛПЗ №3. Ветвление.**

**План: Составление алгоритмов в виде блок-схем. Инструкции альтернативы и варианта.**

1. Составьте алгоритм в виде блок-схемы вычислений значений функции:

***y = |x - 2| + |x + 5|***

2. Составьте алгоритм в виде блок-схемы вычислений значения

⎧ *n/3*, если  *n* кратно *3*;

*X =* ⎨ *n2*, если *n* при делении на *3* даётостаток *1*;

⎩ *n-2*, если  *n* при делении на *3* даёт остаток *2*.

3. Составьте алгоритм в виде блок-схемы. Даны действительные числа *x, y, z*. Выяснить, существует ли треугольник с длинами сторон *x, y, z*.

4. Заданы значения переменных *r, a*. Составьте алгоритм в виде блок-схемы, определяющую, поместится ли круг радиусом *r* в квадрате со стороной *a*. Если да, то присвоить переменной Yes – значение True, если нет – False.

5. Составьте алгоритм в виде блок-схемы. Для заданного числа *a* найти корень уравнения *f(x) = 0*, где

****

6. Составьте алгоритм в виде блок-схемы. Даны числа *a1*, *b1*, *c1*, *a2*, *b2*, *c2*. Напечатать координаты точки пересечения прямых, описываемых уравнениями

***a1 x + b1 y = c1* и *a2 x + b2 y = c2***

либо сообщить, что эти прямые совпадают, не пересекаются или вовсе не существуют.

7. Составьте алгоритм в виде блок-схемы. Даны числа *a*, *b* и *c* (*a ≠ 0*). Найти вещественные корни уравнения *a x4 + b x2 + c = 0*. Если корней нет, то сообщить об этом.

8. Составьте алгоритм в виде блок-схемы. Для целого числа *k* от 1 до 99 напечатать фразу "мне *k* лет", учитывая при этом, что при некоторых значениях *k* слово "лет" надо заменить словом "год" или "года".

9. Составьте алгоритм в виде блок-схемы. Для натурального числа *k* напечатать фразу "мы нашли *k* грибов в лесу", согласовав окончание слова "гриб" с числом *k*.

10. Составьте алгоритм в виде блок-схемы. Даны целые числа *а, b, c*. Если *a ≤ b ≤ c*, то все числа заменить их квадратами, если *a > b > c*, то каждое число заменить наибольшим из них, в противном случае сменить знак каждого числа.

11.Составьте алгоритм в виде блок-схемы, которая определяет, является ли введённый символ гласной или согласной буквой английского алфавита.

12. Составьте алгоритм в виде блок-схемы. Ввести номер дня недели. Вывести его название.

13. Составьте алгоритм в виде блок-схемы. Написать вычисление функции *sign(x, y)*. Её значение равно *|x|*, если *y ≥ 0*, и *-|x|*, если *y < 0*.

**ЛПЗ №4. Задачи с использованием операторов цикла. Структура операторов цикла.**

**План: Составление программ с применением операторов цикла for, while.**

**Задания**

Определите:

a) значение параметра а после окончания работы следующей инструкции:

for а in range(10, 21):

print(a)

б) значение параметра n после окончания работы следующей инструкции

m = [1, 2, 3, 4, 5]

for n in m:

print(n)

Это будет важный результат, который надо учитывать при разработке программ.

Как должна быть оформлена функция range(), если действия в теле инструкции for должны выполняться для всех значений:

а) от 0 до n(n> 0) с шагом, равным 1;

б) от 4 до k(k> 4) с шагом, равным 1;

в) от 19 до 0 с шагом, равным –1;

г) от 18 до 8 с шагом, равным –1.

5. Может ли тело инструкции for с функцией range()при шаге изменения параметра, равном 1, не выполниться ни разу?

6. Сколько раз будет выполняться тело инструкции for со следующим «заголовком»:

• for а in range(1, 10):

• for b in range(10, 20):

• for i in range(n, m): #(m ≥ n)

**ЛПЗ №5-6. Массивы. Сортировка.**

**План: Обработка одномерных массивов, обработка двумерных массивов. Методы сортировки массивов.**

**Задания**

1. Составьте алгоритм в виде блок-схемы. Подсчитать количество элементов целочисленного массива *(x1, x2, …, xn)*, кратных трём.

2. Составить алгоритм в виде блок-схемы вычисления среднего арифметического элементов массива *(x1, x2, …, xn), n ≤ 100*.

3. Составьте алгоритм в виде блок-схемы нахождения наибольшего элемента матрицы A(5, 10) и номера строки и столбца, на пересечении которых он находится.

4. Составьте алгоритм в виде блок-схемы. Дана вещественная квадратная матрица. Сформировать вектор, в котором элементы вычисляются как разность наибольших и наименьших элементов соответствующих строк.

5. Составьте алгоритм в виде блок-схемы. Дана квадратная матрица порядка *n*. Построить массив *b1, b2, …, bn*, в котором *bi=1*, когда в *i*-м столбце есть хотя бы один отрицательный элемент, и *bi=0* в противном случае.

6. Составьте алгоритм в виде блок-схемы . Дана матрица размерности *m×n*. Переставляя её строки и столбцы, добиться того, чтобы наибольший элемент (один из них) оказался в левом верхнем углу.

7. Составьте алгоритм в виде блок-схемы. Дана действительная матрица *m×n*. Сформируйте вектор *b*, в котором элементы *bi* вычисляются как среднее арифметическое элементов соответствующих столбцов.

8. Составьте алгоритм в виде блок-схемы. Элементы массива *x* циклически сдвинуть на *k* позиций влево. Это значит, что в новом массиве *x'*: *x1' = xk+1, x2' = xk+2, …, xn-k' = xn, xn-k+1' = x1, xn-k+2' = x2, …, xn' = xk*.

9. Составьте алгоритм в виде блок-схемы. Дан массив целых чисел. Определить в нём количество инверсий (то есть таких пар элементов, в которых большее число находится слева от меньшего: *xi > xj* при *i < j*).

10. Составьте алгоритм в виде блок-схемы. Элемент матрицы назовём седловой точкой, если он является наименьшим в своей строке и одновременно наибольшим в своём столбце или, наоборот, является наибольшим в своей строке и одновременно наименьшим в своём столбце. Для заданной целой матрицы размером *m×n* напечатать индексы всех её седловых точек.

**ЛПЗ №7. Алгоритмы обработки строк.**

**План: Работа со строками. Алгоритмы обработки строковых данных. Составление алгоритмов в виде блок-схем.**

**Задания**

Дан текст.

а) Подсчитать количество символов "!" и "?" в данной строке.

б) Подсчитать число букв "о" в последнем слове данной строки.

в) Найдите количество слов, начинающихся с буквы "м".

г) Найдите количество слов с удвоенной буквой "н".

д) Найти длину самого длинного слова.

2. составьте алгоритм программы, которая во введённой с клавиатуры строке преобразует строчные буквы русского алфавита в прописные. (Учтите, что функция UpCase с символами русского алфавита не работает).

3. Дана строка s и строка s1. Найти первое вхождение s1 в s **справа** и удалить его из s.

4. Дана строка символов. Напечатать только строчные русские буквы, входящие в этот текст.

5. Дан текст из строчных русских букв. Напечатать этот текст прописными русскими буквами.

6. Дан текст из прописных русских букв. Определить, упорядочены ли эти буквы по алфавиту.

7. Напечатать в алфавитном порядке все различные строчные русские буквы, входящие в заданный текст из не более чем 255 литер.

8. Дана строка символов из латинских букв. Напечатать её, предварительно преобразовав следующим образом:

а) все вхождения abc заменить на def;

б) удалить все вхождения w, если такие есть (образовавшиеся "дыры" заполнить последующими буквами, а в конец добавить пробелы);

в) удалить все вхождения th;

г) заменить на ks все вхождения x, если такие есть;

д) после каждой буквы q добавить букву u;

е) заменить все вхождения ph на f, а все вхождения ed на ing.

**ЛПЗ № 8. Списки.**

**План: Использование списков. Операции со списками. Составление алгоритмов в виде блок-схем.**

**Задания.**

1. Заполнить список двадцатью символами «#».

2. Заполнить список из n элементов случайными целыми числами из интервала от а до b.

3. Заполнить список двадцатью пятью первыми натуральными числами (1, 2, …, 25), после чего добавить в него числа 100 и 200.

4. Дан список a из десяти элементов с числами, среди которых есть отрицательные. Записать все отрицательные числа во второй список. Разработать два варианта программы:

1) без использования генератора списка;

2) с использованием генератора списка.

5. Дан список. Получить новый список, в котором будут все элементы заданного списка, кроме элемента с индексом k. Разработать два варианта программы:

1) без использования генератора списка;

2) с использованием генератора списка.

6. Дан список, в котором есть числа 13. Получить новый список, в котором будут все элементы заданного списка, кроме числа 13. Разработать два варианта программы:

1) без использования генератора списка;

2) с использованием генератора списка.

7. Заполнить список десятью первыми числами последовательности Фибоначчи.

8. Начиная поиск с числа 100, найти первые 10 простых чисел и записать их в список.

**ЛПЗ № 9. Работа с файлами**

**План: Работа с файлами. Чтение информации из файла.**

**Задачи для разработки программ**

1. Используя файл, созданный при решении задачи, рассмотренной в лабораторной работе, получить список, в котором будут два элемента: 'Здравствуйте,'и 'люди!'.

2. Записать в текстовый файл строку из нескольких слов, значение которой задается в ходе выполнения программы (между всеми словами строки должен быть один пробел). Используя этот файл, получить список, в котором будут все слова заданной строки.

3. В текстовом файле gosud.txt на отдельных строках записаны названия 10 государств, в файле stolitsi.txt– их столицы (также на отдельных строках и в том же порядке, что и названия государств). Даны названия двух столиц. Определить названия соответствующих государств.

4. Имеется файл, в каждой строке которого записано слово.

Определить:

a) среднюю «длину» слова;

б) количество слов, в которых больше пяти символов;

в) количество символов в самом коротком слове;

г) номер строки, в которой записано первое самое длинное слово;

д) количество символов в слове, больше которого только в самом длинном слове;

е) количество слов, начинающихся на букву «м»;

ж) имеется ли в файле заданное слово.

Списки не использовать.

5. Имеется файл, в каждой строке которого записано целое число. Определить:

a) сумму всех чисел;

б) среднее арифметическое всех чисел;

в) сумму чисел, записанных на 2-й, 4-й, 6-й, … строках;

г) количество четных чисел;

д) среднее арифметическое отрицательных чисел;

е) максимальное число в файле;

ж) номер строки, в которой записано первое минимальное число файла;

з) имеется ли в файле заданное число.

Списки не использовать.

6. Имеется файл, в каждой строке которого записано несколько слов, разделенных одним пробелом. В конце некоторых строк записана точка. Определить:

а) количество строк, оканчивающихся точкой;

б) максимальное число слов в отдельной строке;

в) общее количество всех слов в файле.

Списки не использовать.

**ЛПЗ № 10. Подпрограммы-функции.**

**План: Создание собственных функций. Оформление функции в программах на языке Python. Вызов функций для выполнения. Функция с параметрами. Локальные и глобальные переменные.**

**Разработка программ**

1. Даны два одномерных массива из целых чисел. Напишите процедуру отыскания элементов, принадлежащих: а) первому и второму массивам; б) первому или второму массивам; в) первому массиву, но не принадлежащих второму массиву.

2. Даны действительные числа *x1, y1, x2, y2,…, xN, yN (N>2)*. Найдите периметр N-угольника, вершины которого имеют соответственно координаты *(x1, y1), (x2, y2), …, (xN, yN)*.

3. Даны координаты вершин двух треугольников. Определить, какой из них имеет б**о**льшую площадь.

4. Вычислите суммы положительных и отрицательных чисел, записанных в текстовом файле, и выведите на печать значения сумм и количества тех и других чисел.

5. Составьте программу подсчёта числа сочетаний *C(n,m)* = . Создайте функцию вычисления факториала числа *Fact(n)*.

6. Напишите программу нахождения наибольшего общего делителя трёх чисел.

7. Найти наименьшее общее кратное четырёх заданных натуральных чисел.

8. Составьте подпрограмму распознания по трём заданным вещественным значениям *a, b, c* являются ли они сторонами прямоугольного треугольника. Если это так, подпрограмма должна также посчитать его площадь *s*.

9. Даны координаты вершин треугольника и координаты некоторой точки внутри него. Найти расстояние от данной точки до ближайшей стороны треугольника.

10. Для произвольного вещественного вектора *x=(x1, x2, …, xn)* проверить соотношения между нормами:

||*x*||1 ≤ ||*x*||2 ≤ ||*x*||C ≤ **||*x*||2 ≤ *n* ||*x*||1

11. Даны три вещественные матрицы *n\*n*. Напечатать ту из них, норма которой наименьшая.

12. Даны *n*-элементные вещественные векторы *x* и *y* и матрицы *A, B, C* порядка *n\*n*. Вычислить комбинацию скалярных произведений *(Ax, By) + (Cx, y) / (x, By)*.

13. Даны *n*-элементные вещественные векторы *x,* *y, z* и матрицы *A, B, C* порядка *n\*n*. Вычислить вектор *u = Ax + By - Cz*.

**ЛПЗ № 11. Модули в Python**

**План: Создание собственных модулей и использование стандартных модулей.**

**Формы и методы обучения: Обсуждение, закрепление темы.**

**Задачи для разработки программ**

1. Создайте модуль, включающий в себя подпрограммы, выполняющие умножение матрицы на число, сложение, умножение и транспонирование матриц.

2. Создайте модуль, включающий в себя подпрограммы определения типа треугольника, вычисления его периметра и площади.

3. Создайте модуль, включающий в себя подпрограммы определения типа четырёхугольника, вычисления его периметра и площади.

4. Создайте модуль, включающий в себя подпрограммы вычисления длины окружности и площади круга, длины и площади эллипса, длины дуги окружности и эллипса, площади сектора круга и эллипса.

**ЛПЗ № 12. Множества.**

**План: Создание множеств, применение операции над множествами.**

**Задачи для разработки программ**

1. Дано множество A: **var** A: **set of** 0..99;
2. Подсчитайте количество элементов в этом множестве.
3. Дано множество A: **var** A: **set of** 'a'..'z';
4. Напечатайте в алфавитном порядке все элементы этого множества.
5. Дана матрица A и множества K и L:

**const** N = 3;

**type** TNomer = 1..N;

**var**

A: **array**[TNomer,TNomer] **of** Integer;

K, L: **set of** TNomer;

1. Найдите сумму тех элементов матрицы A, номера строк которых принадлежат множеству K, а номера столбцов – множеству L.
2. Не используя дополнительные переменные, поменять местами значения переменных-множеств A и B.
3. Дано натуральное число *n*. Напечатать в возрастающем порядке все цифры, не входящие в десятичную запись этого числа.
4. Дан текст из строчных латинских букв. Напечатать все буквы, входящие в текст не менее двух раз.
5. Дан текст из строчных латинских букв. Напечатать все буквы, входящие в текст по одному разу.
6. Дан текст. В алфавитном порядке напечатать (по разу) все строчные русские гласные буквы (а, е, ё, и, о, у, ы, э, ю, я), входящие в этот текст.
7. Дан текст из строчных латинских букв. Определите, каких букв больше: гласных (a, e, i, o, u) или согласных.
8. В данном массиве значений из диапазона от 1 до 50 найти количество чисел Фибоначчи и количество чисел, в десятичной записи которых первая значащая цифра 1 или 2.

**ЛПЗ № 13. Рекурсия.**

**План: создание рекурсивных алгоритмов.**

**Задачи для разработки программ**

1. Написать функцию *Root(f,a,b,Eps)*, которая методом деления отрезка пополам находит с точностью *Eps* корень уравнения *f(x)=0* на отрезке [*a, b*]. Считать, что *Eps>0, a<b, f(a) •\_f(b)<0* и *f(x)* – непрерывная и монотонная функция на отрезке [*a, b*].

2. Написать функцию *Newton(f,df,a,b,Eps)*, которая методом касательных (Ньютона) находит с точностью *Eps* корень уравнения *f(x)=0* на отрезке [*a, b*]. Считать, что *Eps>0, a<b, f(a)•\_f(b)<0*, *f(x)* – непрерывно дифференцируемая и монотонная функция на отрезке [*a, b*], а *df(x)* – её производная.

3. Дано *n* различных натуральных чисел (*n<6*). Написать рекурсивную процедуру, печатающую все перестановки этих чисел.

4. Описать рекурсивную функцию *digits* без параметров, которая подсчитывает количество цифр в тексте, заданном во входном символьном файле.

5. Рекурсивно описать функцию *C(m, n)*, где 0 ≤ *m* ≤ *n*, для вычисления биномиального коэффициента по следующей формуле: 

6. Во входном файле задана последовательность положительных вещественных чисел, за которой следует отрицательное число. Описать рекурсивную функцию *sum* без параметров для нахождения суммы этих положительных чисел.

7. Напечатать в обратном порядке заданный во входном файле текст.

8. Дана последовательность ненулевых целых чисел, за которой следует 0. Напечатать сначала все отрицательные числа этой последовательности, а затем – все положительные (в любом порядке).

9. Даны неотрицательные целые числа *n, m*. Вычислить функцию Аккермана:

⎛*m + 1*, если *n = 0*,

*A(n, m) =* ⎨*A(n - 1, 1)*, если *n > 0, m = 0*,

⎝*A(n - 1, A(n, m - 1))*, если *n > 0, m > 0*.

10. Числа Фибоначчи определяются следующим образом:

*u1 = u2 = 1, un = un-1 + un-2*, для *n = 3, 4, 5, …*

Написать рекурсивную функцию, вычисляющую *un* для заданного *n*.

11. Напишите рекурсивную функцию вычисления наибольшего общего делителя двух натуральных чисел, основанную на соотношении: НОД(*n, m*) = НОД(*n, r*), где *r* – остаток от деления *n* на *m*. Чем эта функция хуже нерекурсивной?

12. Напишите рекурсивную логическую функцию *simm(s, i, j)*, проверяющую, является ли симметричной часть строки *s*, начинающаяся *i*-м и кончающаяся *j*-м её элементами.

13. Напишите рекурсивную процедуру *Strange(N)*, где *N* – целое. Процедура завершает работу при *N=1*, при чётном *N* вызывается *Strange(N/2)*, при нечётном *N* вызывается *Strange((3•N+1)/2)*. Программа должна напечатать число рекурсивных вызовов этой процедуры.

**ЛПЗ №14. Алгоритмы поиска.**

**План: Работа с алгоритмами поиска. Поиск, удаление, поиск. Сортировка, поиск минимальных, определение индексов.**

**Составить алгоритм в виде блок-схемы.**

Напишите функцию, которая возвращает разность между наибольшим и наименьшим значениями из списка целых случайных чисел.

2. Напишите программу, которая для целочисленного списка из 1000 случайных элементов определяет, сколько отрицательных элементов располагается между его максимальным и минимальным элементами.

3. Найти элемент, наиболее близкий к среднему значению всех элементов списка.

4. Дан список, состоящий из чисел. Найти сумму простых чисел в списке.

5. Дан список целых чисел. Определить, есть ли в нем хотя бы одна пара соседних нечетных чисел. В случае положительного ответа определить номера элементов первой из

таких пар.

6. Дан список целых чисел. Определить количество четных элементов и количество

элементов, оканчивающихся на цифру 5.

7. Задан список из целых чисел. Определить процентное содержание элементов, превышающих среднеарифметическое всех элементов списка.

8. Задан список из целых чисел. Определить количество участков списка, на котором элементы монотонно возрастают (каждое следующее число больше предыдущего).

9. Дан список из 20 элементов. Найти пять соседних элементов, сумма значений которых максимальна.

**ЛПЗ №15. Указатели.**

**План: Работа с указателями. Операции с указателями**

**Составить алгоритм в виде блок-схемы.**

1. Даны два динамических вектора A и B из N элементов. Найти их скалярное произведение и освободить память после этого. Проконтролируйте объём свободной памяти в начале и конце работы.

2. Даны две динамических матрицы A и B размерности M×N. Найти их сумму и записать в динамическую матрицу C. Матрицу C записать в файл и после этого освободить всю динамическую память.

3. Дана динамическая матрица. Найти максимальное и минимальное из чисел, встречающихся в ней более одного раза. При неудачном поиске выдать сообщение о том, что все элементы различные. Освободить память после этого.