ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«ЛЕНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ИМЕНИ А.С. ПУШКИНА»**

УТВЕРЖДАЮ

Проректор по учебно-методической

работе

\_\_\_\_\_\_\_\_\_\_\_\_ С.Н.Большаков

РАБОЧАЯ ПРОГРАММА

дисциплины

**Б1.О.08.04 ПРОГРАММИРОВАНИЕ**

Направление подготовки **44.03.05 Педагогическое образование   
(с двумя профилями подготовки)**

Направленность (профиль) **Информатика и математика**

(год начала подготовки - 2022)

Санкт-Петербург

2022

**1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ**

Процесс изучения дисциплины направлен на формирование следующих компетенций:

| **Индекс компетенции** | **Содержание компетенции**  **(или ее части)** | **Индикаторы компетенций (код и содержание)** |
| --- | --- | --- |
| ОПК-8 | Способен осуществлять педагогическую деятельность на основе специальных научных знаний | ИОПК-8.1. Знает основные источники информации, необходимые для успешного освоения содержания дисциплины; терминологию и базовые понятия программирования; основные алгоритмические конструкции языка программирования.  ИОПК-8.2. Умеет выражать конструкции алгоритмического языка средствами языка программирования.  ИОПК-8.3. Владеет навыками работы в электронной образовательной среде для приобретения новых теоретических и фактических знаний, теоретических и практических умений; технологией создания, тестирования и отладки программ в выбранной среде программирования. |
| ПК-3 | Способен применять предметные знания при реализации образовательного процесса | ИПК-3.1. Знает определение и свойства алгоритмов; базовые алгоритмические конструкции, правила их замены и особенности использования; этапы решения задач с помощью компьютера; основные этапы проектирования и разработки программ; понятие «тип данных», основные типы и структуры данных; синтаксис и семантику выбранного для обучения языка программирования (название и правила записи основных конструкций языка программирования); основные методы отладки и тестирования программ; технологию использования компьютерной техники и программного обеспечения в поиске источников информации, позволяющих планировать и реализовывать образовательный процесс по информатике и ИКТ в образовательных учреждениях основного и среднего общего образования. |
| ИПК-3.2. Умеет составлять и записывать на языке блок-схем алгоритмы решения различных задач; распознавать необходимость использования той или иной алгоритмической конструкции; записывать, отлаживать и исполнять программы на выбранном для обучения языке программирования; использовать при написании программ стандартные функции и подпрограммы; использовать при написании программ структурный стиль программирования; выводить на экран и принтер результаты исполнения программы; организовать отчет по лабораторной работе (подготовить в текстовом редакторе); использовать компьютерную технику и программное обеспечение в поиске источников информации, позволяющих планировать и реализовывать образовательный процесс по информатике и ИКТ в образовательных учреждениях основного и среднего общего образования. |
| ИПК-3.3. Владеет навыками работы в выбранной среде программирования; навыками использования компьютерной техники и программного обеспечения в поиске источников информации, позволяющих планировать и реализовывать образовательный процесс по информатике и ИКТ в образовательных учреждениях основного и среднего общего образования. |

**2. Место дисциплины в структуре ОП**

Цель дисциплины: знакомство с основными методами и технологиями программирования.

Задачи дисциплины:

* раскрытие понятия «алгоритм», формирование представления о свойствах алгоритма и способах представления алгоритмов;
* знакомство обучающихся с понятием «тип данных», основными скалярными и структурными типами данных;
* формирование представления об основных технологиях программирования: структурном, объектно-ориентированном, событийном программировании и визуальном проектировании;
* знакомство обучающихся с основными методами тестирования и отладки алгоритмов и программ;
* формирование представления о критериях оценки качества программ;
* демонстрация, как конструкции алгоритмического языка могут быть выражены средствами языка программирования;
* знакомство обучающихся с синтаксисом и семантикой языка программирования Python;
* изучение вопросов составления и написания программ на этом языке.

Дисциплина относится к обязательной части блока 1. Дисциплины (модули), модуль Вычислительная техника и программирование.

Освоение дисциплины и сформированные при этом компетенции необходимы в последующей деятельности.

**3. Объем дисциплины и виды учебной работы**

Общая трудоемкость освоения дисциплины составляет 8 зачетных единиц, 288 академических часов *(1 зачетная единица соответствует 36 академическим часам).*

Очная форма обучения

| **Вид учебной работы** | **Трудоемкость в акад.час** | |
| --- | --- | --- |
|  |  | Практическая подготовка |
| **Контактная работа (аудиторные занятия) (всего):** | 120 | |
| В том числе: |  | |
| Лекции | 30 | - |
| Лабораторные работы / Практические занятия (в т.ч. зачет) | 90/- | -/- |
| **Самостоятельная работа (всего)** | 141 | |
| **Вид промежуточной аттестации (экзамен):** | 27 | |
| контактная работа | 2,35 | |
| самостоятельная работа по подготовке к экзамену | 24,65 | |
| **Общая трудоемкость дисциплины (в час./з.е.)** | 288/8 | |

Заочная форма обучения

| **Вид учебной работы** | **Трудоемкость в акад.час** | |
| --- | --- | --- |
|  |  | Практическая подготовка |
| **Контактная работа (аудиторные занятия) (всего):** | 24 | |
| В том числе: |  | |
| Лекции | 8 | - |
| Лабораторные работы / Практические занятия | 16/- | -/- |
| **Самостоятельная работа (всего)** | 251 | - |
| **Вид промежуточной аттестации (зачет):** | 4 | - |
| контактная работа | 0,25 | - |
| самостоятельная работа по подготовке к зачету | 3,75 | - |
| **Вид промежуточной аттестации (экзамен):** | 9 | |
| контактная работа | 2,35 | |
| самостоятельная работа по подготовке к экзамену | 6,65 | |
| **Общая трудоемкость дисциплины (в час./з.е.)** | 288/8 | |

**4. Содержание дисциплины**

При проведении учебных занятий обеспечивается развитие у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств (включая при необходимости проведение интерактивных лекций, групповых дискуссий, ролевых игр, тренингов, анализ ситуаций и имитационных моделей, составленных на основе результатов научных исследований, проводимых организацией, в том числе с учетом региональных особенностей профессиональной деятельности выпускников и потребностей работодателей).

**4.1. Блоки (разделы) дисциплины**

| **№** | **Наименование блока (раздела) дисциплины** |
| --- | --- |
| 1 | Введение в программирование |
| 2 | Структура программы языка Python |
| 3 | Типы данных языка Python |
| 4 | Выражения в языке Python |
| 5 | Программирование ветвлений |
| 6 | Программирование циклов |
| 7 | Программирование с использованием функций |
| 8 | Строки |
| 9 | Массивы |
| 10 | Записи |
| 11 | Файлы |
| 12 | Множества |

**4.2. Примерная тематика курсовых работ (проектов)**

1. Реализация алгоритмов работы с комплексными числами средствами языка программирования Python.
2. Реализация векторно-матричных операций средствами языка программирования Python.
3. Реализация классических алгоритмов над данными целочисленных типов средствами языка программирования Python.
4. Реализация обработки натуральных чисел, представленных в унарной форме, средствами языка программирования Python.
5. Разработка программы «Двоичный калькулятор» средствами языка программирования Python.
6. Разработка программы «Шестнадцатеричный калькулятор» средствами языка программирования Python.
7. Реализация операций с полиномами средствами языка программирования Python.
8. Реализация рекурсивных математических алгоритмов средствами языка программирования Python.
9. Реализация алгоритмов вычисления функций при помощи их разложения в степенной ряд средствами языка программирования Python.
10. Реализация алгоритмов обработки символьной информации средствами языка программирования Python.
11. Решение комбинаторных задач средствами языка программирования Python.
12. Разработка программы «Расчет стоимости компьютера» средствами языка программирования Python.
13. Разработка программы «Англо-русский словарь по основам вычислительной техники» средствами языка программирования Python.
14. Разработка программы «Исполнитель Буквоед» средствами языка программирования Python.
15. Разработка программы «Электронная касса» средствами языка программирования Python.
16. Разработка программы «Телефонная книга» средствами языка программирования Python.
17. Разработка программы «Кулинарная книга» средствами языка программирования Python.
18. Реализация алгоритмов двоичного кодирования цифр средствами языка программирования Python.
19. Реализация алгоритмов сортировки массивов методами прямого выбора и подсчета средствами языка программирования Python.
20. Реализация алгоритмов сортировки массивов методами прямого обмена и слияния средствами языка программирования Python.

Требования к курсовой работе (на примере конкретной темы)

Тема работы «Реализация алгоритмов работы с комплексными числами средствами языка программирования Python»

**1.** Общие требования к программе:

* + в каждый момент времени пользователю должны быть доступны только те операции, которые могут быть выполнены (например, пока пользователь не задал исходные данные, он не может провести вычисления);
  + каждая из основных задач должна быть представлена отдельной подпрограммой;

пользователь должен иметь возможность работать как с терминалом, так и с файлами (как при вводе, так и при выводе данных).

**2.** Программа должна позволять пользователю:

* + осуществить выбор формы представления комплексных чисел (алгебраическая, тригонометрическая, показательная);
  + выбрать способ получения операндов (с терминала, из файла);
  + выбрать операцию (*смена знака*, *нахождение абсолютной величины*, возведение в степень, извлечение корня, *сложение, вычитание,* ***умножение, деление***);
  + ввести операнд (для унарных операций) и операнды (для бинарных операций);
  + получить на экране результат выполнения операции;
  + сохранить результат работы в файле.

**3.** Курсовая работа должна содержать:

* + теоретические сведения о комплексных числах и операциях над ними (с подробно разобранными примерами);
  + описание всех использованных в программе структур данных и алгоритмических конструкций;
  + информационную модель;
  + блок-схему;
  + текст файла программы с подробными комментариями;
  + подробную инструкцию пользователя по работе с программой;
  + систему тестов.

**Общие требования к курсовой работе**

***Курсовая работа*** должна содержать:

* + Титульный лист;
  + Оглавление в виде нумерованного или многоуровневого списка;
  + Введение;
  + Текст курсовой работы;
  + Заключение;
  + Список источников.

***Во введении*** должна быть обоснована актуальность выбранной темы работы, сформулированы ее цель и задачи, описана структура работы.

***Текст курсовой работы*** должен содержать:

* + Теоретическую часть (первая глава);
  + Практическую часть (вторая глава).

*Теоретическая часть* должна содержать все необходимые теоретические сведения для понимания работы (теорию по теме курсовой работы и дополнительные сведения при необходимости).

*Практическая часть* должна содержать решения задач (темы 1-12). Для каждой задачи должны быть построены информационная модель задачи, блок-схема, приведен текст программы и система тестов.

Для тем 13-30 практическая часть должна содержать: постановку задачи, информационную модель задачи, блок-схему программы, текст программы, систему тестов, инструкцию пользователя по работе с программой.

***В заключении*** должны быть подведены итоги работы.

Все страницы курсовой работы (начиная, со второй) должны быть пронумерованы, т.е. оглавление — страница №2, и т.д..

Курсовая работа должна содержать ссылки на использованную литературу. Ссылки оформляются следующим образом:

* + На теоретический фрагмент: [№ книги в списке литературы, с.№ страницы] (например, [9, с.23–26]);
  + На задачу: [№ книги в списке литературы, с.№ страницы, № задачи] (например, [5, с.46, №3]).

***Список источников*** должен быть разбит на две части: ***Литература*** и ***Интернет-Источники***.

Каждая часть оформляется в виде нумерованного списка. Общее количество источников – не менее 15.

# Список источников

# Литература:

1. Вирт Н. Алгоритмы и структуры данных. – М.: ДМК Пресс, 2010. – 272 с.

# Интернет-источники:

1. Официальный сайт Python. [Электронный ресурс]. URL: www.python.org/ (дата обращения \_\_.\_\_.\_\_\_\_).

**4.3. Перечень занятий, проводимых в активной и интерактивной формах, обеспечивающих развитие у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств. Практическая подготовка\***

| **№ п/п** | **Наименование блока (раздела) дисциплины** | **Занятия, проводимые в активной  и интерактивной формах** | | **Практическая подготовка\*** |
| --- | --- | --- | --- | --- |
| **Форма проведе-ния занятия** | **Наименование видов занятий** |
| 1. | Введение в программирование | лекционное занятие | Использование презентаций |  |
| 2. | Структура программы языка Python | лекционное занятие | Использование презентаций |  |
| 3. | Типы данных языка Python | лекционное занятие | Использование презентаций |  |
| 4. | Выражения в языке Python | лекционное занятие | использование презентаций |  |
| лабораторное занятие | решение ситуационных задач, работа в группах |  |
| 5. | Программирование ветвлений | лекционное занятие | использование презентаций |  |
| лабораторное занятие | решение ситуационных задач, работа в группах |  |
| 6. | Программирование циклов | лекционное занятие | использование презентаций |  |
| лабораторное занятие | решение ситуационных задач, работа в группах |  |
| 7. | Программирование с использованием функций | лекционное занятие | использование презентаций |  |
| лабораторное занятие | решение ситуационных задач, работа в группах |  |
| 8. | Строки | лекционное занятие | использование презентаций |  |
| лабораторное занятие | решение ситуационных задач, работа в группах |  |
| 9. | Массивы | лекционное занятие | использование презентаций |  |
| лабораторное занятие | решение ситуационных задач, работа в группах |  |
| 10. | Записи | лекционное занятие | использование презентаций |  |
| лабораторное занятие | решение ситуационных задач, работа в группах |  |
| 11. | Файлы | лекционное занятие | использование презентаций |  |
| лабораторное занятие | решение ситуационных задач, работа в группах |  |
| 12. | Множества | лекционное занятие | использование презентаций |  |
| лабораторное занятие | решение ситуационных задач, работа в группах |  |

**\***Практическая подготовка при реализации учебных предметов, курсов, дисциплин (модулей) организуется путем проведения практических занятий, практикумов, лабораторных работ и иных аналогичных видов учебной деятельности, предусматривающих участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.

**5. Учебно-методическое обеспечение для самостоятельной работы обучающихся по дисциплине**

**5.1. Темы для творческой самостоятельной работы обучающегося**

Темы для творческой самостоятельной работы студента формулируются обучающимся самостоятельно, исходя из перечня тем занятий текущего семестра.

**5.2. Темы конспектов**

1. Понятие алгоритма. Свойства алгоритмов. Способы записи алгоритмов.
2. Языки программирования. Классификация языков программирования.
3. Основные алгоритмические конструкции. Исполнители алгоритмов (назначение, среда, режим работы, система команд).
4. Технология решения задач с помощью компьютера (моделирование, формализация, алгоритмизация, программирование).
5. Технология решения задач с помощью компьютера (тестирование и отладка программы).
6. Структура программы на языкеPython.
7. Константы и переменные Простейшие типы данных языкаPython.
8. Встроенные функции языкаPython.
9. Процедуры ввода и вывода данных.
10. Линейная алгоритмическая конструкция. Оператор присваивания.
11. Алгоритмическая конструкция «ветвление». Оператор ветвления.
12. Алгоритмическая структура «цикл». Циклы с параметрами.
13. Алгоритмическая структура «цикл». Циклы с предусловием.
14. Массивы. Одномерные массивы. Описание. Ввод и вывод элементов массива.
15. Массивы. Двумерные массивы. Описание. Ввод и вывод элементов массива.
16. Массивы. Стандартные алгоритмы обработки: арифметические операции; действия, согласно условию; нахождение max или min, сортировка.
17. Массивы. Алгоритмы поиска.
18. Массивы. Алгоритмы сортировки.
19. Строки: Описание, методы работы со строками.
20. Строки: Описание, функции работы со строками.
21. Записи. Описание, ввод, вывод, обращение к отдельным полям.
22. Файлы. Описание, ввод, вывод информации.

**5.3. Вопросы для подготовки к устным собеседованиям (опросам)**

**Тема №1 «Введение в программирование»**

1. Понятие алгоритма. Свойства алгоритмов. Способы записи алгоритмов.
2. Языки программирования. Классификация языков программирования.
3. Основные алгоритмические конструкции. Исполнители алгоритмов (назначение, среда, режим работы, система команд).
4. Технология решения задач с помощью компьютера (моделирование, формализация, алгоритмизация, программирование).
5. Технология решения задач с помощью компьютера (тестирование и отладка программы).

**Тема №2 «Структура программы языка Python»**

1. Структура программы на языке Python.

**Тема №3 «Типы данных языка Python»**

1. Константы и переменные
2. Простейшие типы данных языка Python.

**Тема №4 «Выражения в языке Python»**

1. Встроенные функции языка Python.
2. Ввод и вывод данных.
3. Линейная алгоритмическая конструкция. Оператор присваивания.
4. Расположите в правильном порядке строки программы, вычисляющей площадь параллелограмма:

|  |  |
| --- | --- |
| 1) S=a\*h |  |
| 2) p=2\*a+b) |  |
| 3) print("p=%d"%p, "s=%d"%s) |  |
| 4) a=6  b=3  h=4 |  |

1. Дан фрагмент программы:

A=int(input())

B=int(input())

C=int(input())

X=int(input())

Y=(A+C)/B\*X

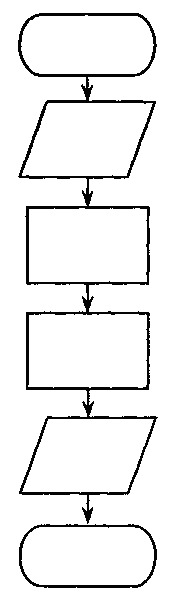
При вводе значений: **?**, 3, 14, 4, значение переменной *Y* будет равно 32. Какое значение было введено для переменной *A*?

1. Вычислите значение выражения:

**1)**220//10%3

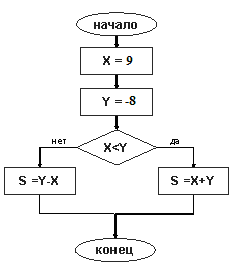
**2)**-16%11\*3

1. Составьте блок-схему алгоритма, вычисляющего квадрат суммы и разность квадратов двух заданных вещественных чисел.



**Тема №5 «Программирование ветвлений»**

1. Алгоритмическая конструкция «ветвление».
2. Оператор ветвления. Полная форма.
3. Оператор ветвления. Сокращенная форма.
4. Оператор выбора.
5. Дана блок-схема



После выполнения алгоритма значение S равно…

1. Что будет выведено на экран, после выполнения данной программы (пользователь ввел значения *a*=7, *n*=4):

a=int(input())

n=int(input())

**if** ((a%2==0) **or not**(n==2)):

print("\n%d\n"%int(a\*2))

**else:**

print("\n%d\n"%int(n\*2))

1. Вычислите значение выражения **(4\*b+a\*c-с<>0)** при **a=-3, b=0, c=2**.
2. Запишите условия:

а) **p** является нечетным числом, не равным 13.

б) точка **a** является серединой отрезка [**b;c]**.

1. Предлагается фрагмент программы:

Предлагается фрагмент программы:

**if** ((x>=0) **and** (y>=0)):

**if** (x\*x+y\*y<16):

F=0

**else:**

F=1

**else:**

F=-1

Запишите результаты ее выполнения, если *x*=–1; *y*=2.

1. Определите, что будет выведено на экран после выполнения фрагмента программы:

k=10

**if** (k%4==0):

print("k=4m")

**elif** (k%4==1):

print("k=4m+1")

**elif** (k%4==2):

print("k=4m+2")

**elif** (k%4==3):

print("k=4m+3")

**else:**

print("Неверный ввод")

**Тема №6 «Программирование циклов»**

1. Алгоритмическая структура «цикл». Циклы с параметрами.
2. Алгоритмическая структура «цикл». Циклы с предусловием.
3. Предлагается фрагмент программы:

P=1

S=0

**for** **in** range(1,4,1):

S=S+i

P=P\*S

Запишите результаты его выполнения: *S* = \_\_\_\_ *P* = \_\_\_\_

1. Предлагается фрагмент программы:

x=0

S=0

**while** (x<=2):

x=x+2

S=S+x

* Запишите результаты его выполнения: *x* = \_\_\_\_ *S* = \_\_\_\_
* Сколько раз выполнится цикл?

1. Предлагается фрагмент программы:

S=0

j=1

**for** i **in** range(2,0,-1):

**while** (j<=3):

S=S-i\*j

j=j+1

Запишите результаты его выполнения: *S* = \_\_\_\_

**Тема №7 «Программирование с использованием подпрограмм»**

1. Понятие подпрограммы.
2. Функции.
3. Определите результат работы программы.

**def** f(x):

**return** x\*x-2\*x-1

a=-1

b=1

d=0.2

z=a

N=a

R=f(a)

**while** (z<b):

**if** (f(z)<R):

N=z

R=f(z)

z=z+d

printf("%f"%N)

1. Опишите функцию *Sign*(*X*) целого типа, возвращающую для числа *X* следующие значения:

. С помощью этой функции найдите значение выражения *Sign*(*A*) + *Sign*(*B*) для данных вещественных чисел *A* и *B*.

**Тема №8 «Строки»**

1. Строки: Описание, методы работы со строками.
2. Строки: Описание, функции работы со строками.
3. Дан фрагмент программы:
4. Дан фрагмент программы:

print("Введите строку\n")

S=input()

S1=""

S2=""

S3=""

K=""

D=""

**for** i **in** range(0,len(S)//2,1):

D=S[i]

K=S[len(S)-i-1]

**for** j **in** range(0,i,1):

S1=S1+S[j]

S1=S1+K

**for** k **in** range(i+1,len(S)-i-1,1):

S2=S2+S[k]

S2=S2+D

**for** l **in** range(len(S)-i,len(S),1):

S2=S2+S[l]

S=S1+S2+S3

print("\nРезультат: \n")

print(S)

Укажите значение переменной *S* после выполнения фрагмента программы, если на входе было введено: том.

1. Дано целое число *N* (>0) и символ *C*. Вывести строку длины *N*, которая состоит из символов *C*.
2. Определите результат сравнения: 'информация'<='Алгоритм'
3. Определите значение переменной *S* для фрагмента программного кода.

S=input()

S1="па"

**for** j **in** range(2,5,1):

S1=S1+S[j]

S=S1

print(S)

Известно, что изначально пользователь ввел карусель.

1. Определите результат работы программы:

destination=input()

st=input()

**if** (st==destination):

print("Строки равны.\n")

**else:**

**if** (st>destination):

print("Первая строка больше.\n")

**else:**

print("Первая строка меньше.\n")

Известно, что изначально пользователь ввел Информация и ИНФОРМАЦИЯ.

**Тема №9 «Массивы»**

1. Массивы. Одномерные массивы. Описание. Ввод и вывод элементов массива.
2. Массивы. Двумерные массивы. Описание. Ввод и вывод элементов массива.
3. Массивы. Стандартные алгоритмы обработки: арифметические операции; действия, согласно условию; нахождение max или min, сортировка.
4. Массивы. Алгоритмы поиска.
5. Массивы. Алгоритмы сортировки.
6. Дан фрагмент программы:

a=[]

n=int(input("Задайте количество строк и столбцов квадратной матрицы\n"))

**for** I **in** range(0,n,1):

b=[]

**for** J **in** range(0,n,1):

print("a[%d"%I,"%d"%J,"]=")

b=b+[input()]

a=a+[b]

pr=1

print(a)

**for** I **in** range(0,n,1):

mina=int(a[I][0])

for J **in** range(0,n,1):

**if** (int(a[I][J])<mina):

mina=int(a[I][J])

pr=pr\*mina

print("%d "%pr)

Определите, что будет напечатано на экране в результате его работы, если известно, что пользователь ввел 3, 1, 2, -1, 9, -18, 34, 1, 8, 16.

1. Дан фрагмент программы:

Y=[]

m=int(input("Задайте количество элементов\n"))

**for** I **in** range(0,m,1):

print("Y[%d"%I,"]=")

Y=Y+[input()]

**for** I **in** range(0,m,1):

**if** **not**(float(Y[I])==0):

Y[I]=I

**for** I **in** range(0,m,1):

print("%5.5f "%float(Y[int(I)]))

Определите, что будет напечатано на экране в результате его работы, если известно, что пользователь ввел 4, 0, 0, 2, -10.

1. Дана квадратная матрица. Определите количество столбцов матрицы, сумма элементов которых равна 6.

**Тема №10 «Записи»**

1. Записи. Описание, ввод, вывод, обращение к отдельным полям.
2. Дать описание записи содержащей информацию о планетах солнечной системы: название планеты, удаленность планеты от Солнца, диаметр планеты, отношение массы планеты к массе Земли, наличие спутников.
3. Инициализировать описанную в задании 1 запись, используя следующую информацию: Юпитер – самая большая из планет солнечной системы. Период его обращения вокруг Солнца составляет 11,86 земных лет. Масса Юпитера в 318 раз больше массы Земли. Экваториальный диаметр Юпитера в 11,2 раза больше земного и составляет 142 тыс. км. Расстояние от Юпитера до Солнца составляет 0,387 астрономических единиц или 58 млн. км. Известно 13 спутников Юпитера.

**Тема №11 «Файлы»**

1. Файлы. Описание, ввод, вывод информации.
2. Дан фрагмент программы:

f = open('numbers.txt', 'r+')

sm=0

s=''

n=0

st=f.readline()

**for** i **in** range(0,len(st)-1,1):

s=s+st[i]

**if** **not**(s==''):

sm=sm+int(s)

n=n+1

**while** **not**(st==''):

s=''

st=f.readline()

**for** i **in** range(0,len(st)-1,1):

s=s+st[i]

**if not**(s==''):

sm=sm+int(s)

n=n+1

f.close

sr=sm/n

print("%f"%sr)

Укажите значение переменной *sr* после выполнения фрагмента программы, если файл C:\numbers.txt содержит числа 1, 3, 5, 7, 9 (числа записаны в файл по одному в строке).

**Тема №12 «Множества»**

1. Множества. Описание, ввод, вывод.
2. Действия над множествами.
3. Вычислите значение выражения

**1)** {4, 8, 6, 7, 5}=={4, 5, 6, 7, 8}

**2)**{11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21}>=

{15, 17, 10, 5, 6, 13}

**3)**6 **in** {3, 5, 2, 6, 1}

**4)**{23, 17, 50, 26, 8}&{10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}|{17, 6}

1. Определите значение множества y1 после выполнения приведенного фрагмента. По какому принципу отбирались элементы y1?

y1=set()

**for** i **in** range(1,101,1):

y1=y1|{i}

**for** i **in** range(1,101,1):

**if** **not**(500%i==0):

y1=y1-{i}

**5.4. Вопросы для подготовки к коллоквиумам**

**Коллоквиум №1**

**Тема №1 «Введение в программирование»**

1. Понятие алгоритма. Свойства алгоритмов. Способы записи алгоритмов.
2. Языки программирования. Классификация языков программирования.
3. Основные алгоритмические конструкции. Исполнители алгоритмов (назначение, среда, режим работы, система команд).
4. Технология решения задач с помощью компьютера (моделирование, формализация, алгоритмизация, программирование).
5. Технология решения задач с помощью компьютера (тестирование и отладка программы).

**Тема №2 «Структура программы языка** **Python»**

1. Структура программы на языке Python.

**Тема №3 «Типы данных языка** **Python»**

1. Константы и переменные
2. Простейшие типы данных языка Python.

**Тема №4 «Выражения в языке** **Python»**

1. Встроенные функции языка Python.
2. Процедуры ввода и вывода данных.
3. Линейная алгоритмическая конструкция. Оператор присваивания.

**Коллоквиум №2**

**Тема №5 «Программирование ветвлений»**

1. Алгоритмическая конструкция «ветвление».
2. Оператор ветвления. Полная форма.
3. Оператор ветвления. Сокращенная форма.

**Тема №6 «Программирование циклов»**

1. Алгоритмическая структура «цикл». Циклы с параметрами.
2. Алгоритмическая структура «цикл». Циклы с предусловием.

**Коллоквиум №3**

**Тема №7 «Программирование с использованием подпрограмм»**

1. Понятие подпрограммы.
2. Функции.

**Тема №8 «Строки»**

1. Строки: Описание, методы работы со строками.
2. Строки: Описание, функции работы со строками.

**Тема №9 «Массивы»**

1. Массивы. Одномерные массивы. Описание. Ввод и вывод элементов массива.
2. Массивы. Двумерные массивы. Описание. Ввод и вывод элементов массива.
3. Массивы. Стандартные алгоритмы обработки: арифметические операции; действия, согласно условию; нахождение max или min, сортировка.
4. Массивы. Алгоритмы поиска.
5. Массивы. Алгоритмы сортировки.

**Коллоквиум №4**

**Тема №10 «Записи»**

1. Записи. Описание, ввод, вывод, обращение к отдельным полям.

**Тема №11 «Файлы»**

1. Файлы. Описание, ввод, вывод информации.

**Тема №12 «Множества»**

1. Множества. Описание, ввод, вывод.
2. Действия над множествами.

**5.5. Темы рефератов**

Выполнение рефератов не предполагается

**6. Оценочные средства для текущего контроля успеваемости**

**6.1. Текущий контроль**

| **№**  **пп** | **№ и наименование блока (раздела) дисциплины** | **Форма текущего контроля** |
| --- | --- | --- |
| 1 | Введение в программирование | Устное собеседование  Коллоквиум №1 |
| 2 | Структура программы языка Python | Устное собеседование  Коллоквиум №1 |
| 3 | Типы данных языка Python | Устное собеседование  Коллоквиум №1 |
| 4 | Выражения в языке Python | Устное собеседование Коллоквиум №1 |
| 5 | Программирование ветвлений | Устное собеседование Контрольная работа № 1  Коллоквиум №2 |
| 6 | Программирование циклов | Устное собеседование  Контрольная работа № 2  Коллоквиум №2  Итоговый тест  Итоговая контрольная работа |
| 7 | Программирование с использованием функций | Устное собеседование  Коллоквиум №3 |
| 8 | Строки | Устное собеседование  Коллоквиум №3 |
| 9 | Массивы | Устное собеседование  Контрольная работа № 3  Коллоквиум №3 |
| 10 | Записи (списки) | Устное собеседование  Коллоквиум №4 |
| 11 | Файлы | Устное собеседование  Коллоквиум №4 |
| 12 | Множества | Устное собеседование  Контрольная работа № 4  Коллоквиум №3 |

**6.2. Примеры оценочных средств для текущего контроля по дисциплине**

**Темы конспектов**

Представлены в разделе 5.2.

**Вопросы для сдачи коллоквиумов**

Представлены в разделе 5.4.

**Вопросы для проведения устных опросов**

Представлены в разделе 5.3.

**Задания для лабораторных занятий**

**Тема: «Язык программирования** **Python. Оператор присваивания»**

**Задание 1.** Запросите у пользователя валютный курс на сегодняшний день, затем имеющуюся у него рублёвую сумму и рассчитайте, сколько долларов он может купить.

**Задание 2.** Вычислите площадь параллелограмма по известным стороне и высоте, опущенной на эту сторону.

**Задание 3.** С клавиатуры вводим координаты векторов **x{a;b}**, **y{c;d}**. Вычислите координаты вектора **z = x + y**.

**Задание 4.** Напишите программу пересчёта расстояния из милей в километры (1 миля = 1,6 км).

Результат должен быть выведен на экран в виде:

<число> миль = <число> км

**Тема: «Язык программирования** **Python. Условный оператор»**

**Задание 1.** Определите, в норме ли вес обследуемого пациента (нормой считается вес, равный (рост(см) - 100) ± 5 кг). Значения роста и веса должны вводиться с клавиатуры, результат выводиться на экран в виде одного из следующих сообщений: “вес ниже нормы”, “вес в норме”, “избыточный вес”.

**Задание 2.**Составьте программу, которая моделирует работу простейшего калькулятора: позволяет ввести первое число, знак операции (+,-,\*,/) и второе число, после чего выводит результат в виде:

<1-е число><знак операции><2-е число>=<результат>

Или

<1-е число><знак операции><2-е число> не может быть вычислено

**Задание 3.** Вычислите значение функции по заданному значению аргумента:



**Задание 4.** Напишите программу, которая по знаку арифметической операции выводит ее название. Например, "+" – сложение.

**Задание 5.**Напишите программу, которая определяет стоимость разговора по телефону с учетом скидки, предоставляемой по субботам и воскресеньям.

**Задание 6.**Напишите программу, которая после введённого с клавиатуры числа (в диапазоне от 1 о 999), обозначающего денежную единицу, дописывает слово “рубль” в правильной форме. Например: 12 рублей, 21 рубль и т.д.

**Тема: «Язык программирования** **Python. Циклы с предусловием, с заданным числом повторений. Вложенные циклы»**

**Задание 1.**Напишите программу, которая вычисляет сумму первых *n* членов ряда . Количество суммируемых членов ряда вводится во время работы программы.

**Задание 2.** Напечатайте все нечётные числа из промежутка от *A* до *B* (*A*<*B*).

**Задание 3.**Вычислите значение выражения  без использования операции умножения.

**Задание 4.** Даны *n* чисел. Найдите разность максимального и минимального из них.

**Задание 5.** Разработайте программу, вычисляющую множество значений функции на отрезке [*a*,*b*] с шагом *h.*

**Задание 6.** Дано целое число *N* (*N*>0). Последовательность вещественных чисел *Ak* определяется следующим образом:



Выведите элементы *A*1, *A*2, …, *AN*.

**Задание 7.**Дано вещественное число *X* и целое число *N* (*N*>0). Найдите значение выражения . Полученное число является приближённым значением функции *sin* в точке *X*.

**Тема: «Язык программирования** **Python.Функции»**

**Задание 1.**Даны отрезки, имеющие длину *a*, *b*, *c*, *d*. Для каждой тройки этих отрезков, из которых можно построить треугольник, напечатайте площадь данного треугольника. Определите функцию *Plo*(*x*, *y*, *z*), печатающую площадь треугольника со сторонами *x*, *y*, *z*.

**Задание 2.**Даны действительные числа. Вычислите значение следующего выражения с использованием функции:



## Задание 3. Составьте программу вычисления значения выражения u=ex+y+z, где на отрезке [b,c] с шагом h и распечатайте.

Для вычисления *x*, *y*, *z* используйте функцию 

**Задание 4.**Опишите функцию *DigitCountSum*(*K*, *C*, *S*) находящую количество *C* цифр целого положительного числа *K*, а также их сумму *S* (*K* – входной, *C* и *S* – выходные параметры целого типа). С помощью этой функции найдите количество и сумму цифр для каждого из пяти данных целых чисел.

**Задание 5.**Опишите функцию *DegToRad*(*D*) вещественного типа, находящую величину угла в радианах, если дана его величина *D* в градусах (*D* – вещественное число, 0<*D*<360). С помощью данной функции переведите из градусов в радианы пять данных углов.

**Тема: «Язык программирования Python. Строки»**

**Задание 1.** Дана строка. Напечатайте номера позиций, на которых во введенной строке стоят арабские цифры.

**Задание 2.** Составьте программу для подсчета числа букв "*а*" в слове *Х*, стоящих на местах, номер которых кратен 3.

**Задание 3.** Составьте программу, для вычеркивания из слова X каждой третьей буквы.

**Задание 4.**Напишите программу для проверки, есть ли в слове *X* буква "*о*". Если есть, то замените все буквы "*о*" на "*ку*".

**Задание 5.**Дана строка X. Модифицируйте строку следующим образом: удалите из строки все слова, начинающиеся с буквы "А".

**Задание 6.** Дана строка X. Напечатайте все слова нечетной длины из этой строки, заканчивающиеся на букву "ь".

**Тема: «Язык программирования** **Python. Массивы»**

**Задание 1.**Задан одномерный массив. Все его нулевые элементы, перепишите в начало массива, а ненулевые элементы – в конец массива с сохранением порядка следования.(Дополнительного массива не заводите!!!)

**Задание 2.**Задан целочисленный двумерный массив. Найдите произведение минимальных элементов строк этого массива.

**Задание 3.** Задан двумерный вещественный массив. Постройте одномерный массив, элементы которого равны сумме элементов столбцов заданного массива.

**Задание 4.**Дан одномерный массив. Замените положительные элементы этого массива на их индексы.

**Задание 5.**Задан целочисленный двумерный массив. Найдите сумму тех элементов этого массива, которые делятся на 2.

**Задание 6.**Дан целочисленный одномерный массив. Найдите произведение всех четных элементов этого массива.

**Задание 7.**Дан двумерный массив. Определите, есть ли в этом массиве хоть один столбец, среднее арифметическое элементов которого больше заданного числа.

**Задание 8.**Дан двумерный массив. Преобразуйте массив следующим образом: расположите строки в порядке убывания сумм их элементов.

**Тема: «Язык программирования** **Python. Записи»**

**Задание 1.**Опишите, используя структуру данных запись, информацию об абитуриентах (фамилия, имя, отчество, результаты трех экзаменов, факультет). Напишите программу, выдающую список абитуриентов, набравших не менее 10 баллов и поступающих на факультет математики.

**Задание 2.**Опишите, используя структуру данных запись, записную книжку (фамилия и инициалы, год рождения, дата рождения, месяц рождения). Напишите программу, определяющую есть ли среди записанных в эту книжку те, кому в этом году исполнится 30 лет.

**Задание 3.**Хранятся сведения о лесе: вид дерева, общая численность, численность здоровых деревьев. Напишите программу, выдающую: 1) суммарное число деревьев на контрольном участке; 2) суммарное число здоровых деревьев; 3) относительную численность больных деревьев.

**Тема: «Язык программирования** **Python. Файлы»**

**Задание 1.**Дан файл f, компоненты которого являются действительными числами. Найдите модуль суммы компонент файла.

**Задание 2.**С клавиатуры вводится последовательность целых чисел. Запишите введенные числа в файл и замените все отрицательные компоненты файла, стоящие на четных местах, их квадратами.

**Задание 3.**Дан файл, который содержит сведения о сотрудниках учреждения: указывается фамилия сотрудника, его инициалы и заработная плата. Найдите размер средней заработной платы сотрудников.

**Задание 4.**Дан файл, содержащий сведения о веществах: указывается название вещества, его удельный вес и проводимость (проводник, полупроводник, изолятор). Найдите суммарный удельный вес и названия всех проводников.

**Задание 5.**Дан файл, содержащий сведения о книгах. Сведения о каждой из книг – это фамилия автора, название книги, год издания. Выясните, имеются ли книги, изданные до 1905 года.

**Тема: «Язык программирования** **Python. Множества»**

**Задание 1.** С клавиатуры вводятся три множества символьного типа: *Y*1, *Y*2, *Y*3. Сформируйте новое множество *X*=(*Y*1∪*Y*2)/(*Y*1∩*Y*3). Выведите его элементы на печать.

**Задание 2.** Дана матрица размерности *n*×*n* (двумерный массив). Массив состоит из чисел 0, 1, 2, 3. Требуется распечатать индексы таких элементов матрицы, которые являются верхней левой вершиной квадрата 2×2, состоящего из одинаковых элементов.

**Задание 3.** Дан текст. Подсчитайте количество цифр в этом тексте.

**Замечание.** Для решения задачи необходимо сформировать множество цифр.

**Задание 4.**С клавиатуры вводится число. Напечатайте все цифры, которые встречаются в записи числа.

**Контрольная работа №1**

# Вариант №1

**1.**Расположите в правильном порядке строки программы, вычисляющей периметр и площадь параллелограмма:

|  |  |
| --- | --- |
| 1) S=a\*h |  |
| 2) p=2\*a+b) |  |
| 3) print("p=%d"%p, "s=%d"%s) |  |
| 4) a=6  b=3  h=4 |  |

**2.**Дан фрагмент программы:

A=int(input())

B=int(input())

C=int(input())

X=int(input())

Y=(A+C)/B\*X

При вводе значений: **?**, 3, 14, 4, значение переменной *Y* будет равно 32. Какое значение было введено для переменной *A*?

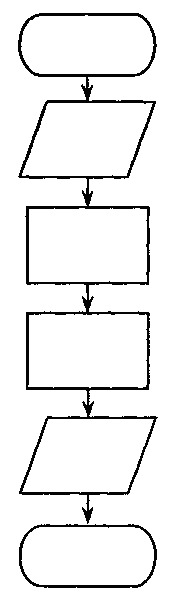
**3.**Вычислите значение выражения:

**1)**220//10%3

**2)**-16%11\*3

**4.** Дано двузначное число. Вычислите куб суммы цифр этого числа.

**5.** Составьте блок-схему алгоритма, который по двум заданным вещественным числам вычисляет коэффициенты приведенного квадратного уравнения, корнями которого являются эти числа.



# Вариант №2

**1.**Расположите в правильном порядке строки программы, вычисляющей периметр и площадь квадрата:

|  |  |
| --- | --- |
| 1) a=6 |  |
| 2) p=4\*a |  |
| 3) print("p=%d"%p, "s=%d"%s) |  |
| 4) S=a\*a |  |

**2.**Дан фрагмент программы:

A=int(input())

B=int(input())

C=int(input())

X=int(input())

Y=(A+C)/B\*X

При вводе значений: 8, **?**, 6, 4, значение переменной *Y* будет равно 28. Какое значение было введено для переменной *B*?

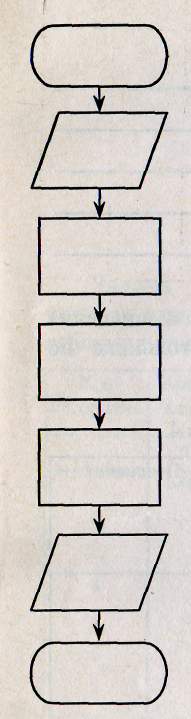
**3.**Вычислите значение выражения:

**1)**3//(5+3)

**2)**(16%6)//2

**4.** Дано двузначное число. Вычислите разность кубов цифр этого числа.

**5.** Дана длина ребра куба. Составьте блок-схему алгоритма нахождения площади грани, площади полной поверхности и объема этого куба.



# Вариант № 3

**1.**Расположите в правильном порядке строки программы, вычисляющей площадь треугольника:

|  |  |
| --- | --- |
| 1) S=1/2\*a\*h |  |
| 2) print("s=%f"%s) |  |
| 3) a=6  h=3 |  |

**2.**Дан фрагмент программы:

A=int(input())

B=int(input())

C=int(input())

X=int(input())

Y=(A+C)/B\*X

При вводе значений: 10, 3, **?**, 4, значение переменной *Y* будет равно 20. Какое значение было введено для переменной *C*?

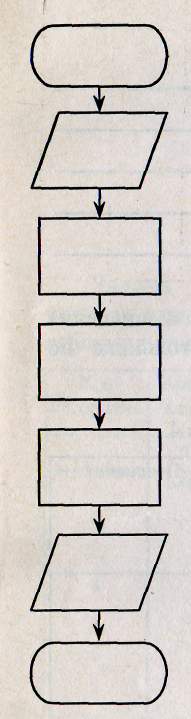
**3.**Вычислите значение выражения:

**1)**220%10//3

**2)**3//5+3

**4.** Дано двузначное число. Вычислите сумму кубов цифр этого числа.

**5.** Дано *а*. Не используя встроенных функций и операций, кроме умножения, получите *а*8за три операции.



# Вариант № 4

**1.**Расположите в правильном порядке строки программы, вычисляющей площадь параллелограмма:

|  |  |
| --- | --- |
| 1) a=6  h=3 |  |
| 2) printf("s=%f"%s) |  |
| 3) S=a\*h |  |

**2.**Дан фрагмент программы:

A=int(input())

B=int(input())

C=int(input())

X=int(input())

Y=(A+C)/B\*X

При вводе значений: 12, 3, 11, **?**, значение переменной *Y* будет равно 44. Какое значение было введено для переменной *X*?

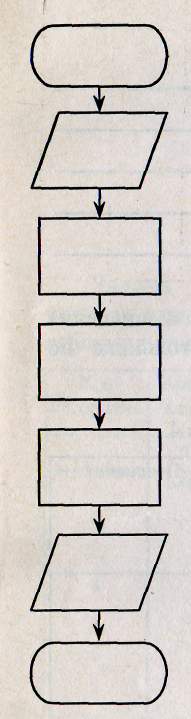
**3.**Вычислите значение выражения:

**1)**(-19+9)//-5

**2)**16%6//2

**4.** Дано трехзначное число. Вычислите куб разности цифр этого числа.

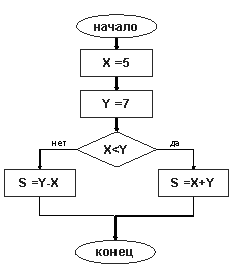
**5.** Составьте блок-схему алгоритма вычисления периметра и площади прямоугольного треугольника по длинам двух его катетов.



**Контрольная работа №2**

**Вариант №1.**

**1.** Дана блок-схема



После выполнения алгоритма значение *S* равно…

**2.** Что будет выведено на экран, после выполнения данной программы (пользователь ввел значения *a*=8, *n*=5):

a=int(input())

n=int(input())

**if** (a<=10):

printf("\n%d\n"%int(a\*2))

**else:**

print("\n%d\n"%int(n\*2))

**3.** Вычислите значение выражения (a%7==1) **and** (a//7==1) при *a*=8.

**4.** Запишите условия:

**1)** *x* лежит вне отрезков [*a*,*b*] и [*c*,*d*].

**2)** из отрезков длиной *a*, *b*, *c* можно построить треугольник.

**5.** Предлагается фрагмент программы:

**if** ((x>=0) **and** (y>=0)):

**if** (x\*x+y\*y<16):

F=0

**else:**

F=1

**else:**

F=-1

Запишите результаты ее выполнения, если *x*=–4; *y*=-3.

**6.**Предлагается фрагмент программы:

x=0

S=0

**while** (x<=2):

S=S+x

x=x+2

* Запишите результаты его выполнения: *x*= \_\_\_\_ *S*= \_\_\_\_
* Сколько раз выполнится цикл?

**7.**Предлагается фрагмент программы:

P=1

S=0

**for** i **in** range(1,4,1):

P=P\*S

S=S+i

Запишите результаты его выполнения: *S*= \_\_\_\_ *P*= \_\_\_\_

**8.** Разработайте программу, вычисляющую множество значений функции *y*=*x*2+5 для *x*, изменяющихся от –10 до 10 с шагом 2. Результат должен быть представлен в виде:

*x*=-10 *y*(*x*)=…

*x*=-8 *y*(*x*)=…

……

**9.**Предлагается фрагмент программы:

S=3

j=1

**for** i **in** range(1,4,1):

**while** (j<=3):

S=S+i\*j

j=j+1

Запишите результаты его выполнения:   
*S*= \_\_\_\_

**Вариант №2**

**1.** Дана блок-схема



После выполнения алгоритма значение *S* равно…

**2.** Что будет выведено на экран, после выполнения данной программы (пользователь ввел значения *a*=3, *n*=3):

a=int(input())

n=int(input())

**if** **not**(a%3==0):

print("\n%d\n"%int(a+5))

**else:**

print("\n%d\n"%int(n+2))

**3.** Вычислите значение выражения (a>=1) **and** (a<=1) при *a*=1.5.

**4.** Запишите условия:

**1)** *x* лежит вне отрезка [*a*,*b*].

**2)** среди чисел *a*, *b*, *c*, *d* есть взаимно-противоположные.

**5.** Предлагается фрагмент программы:

**if** ((x>=0) and (y>=0)):

**if** (x\*x+y\*y<16):

F=0

**else:**

F=1

**else:**

F=-1

Запишите результаты ее выполнения, если *x*=2; *y*=5.

**6.**Предлагается фрагмент программы:

x=0

S=0

**while** (x<=2):

x=x+2

S=S+x

* Запишите результаты его выполнения: *x*= \_\_\_\_ *S*= \_\_\_\_
* Сколько раз выполнится цикл?

**7.**Предлагается фрагмент программы:

P=1

S=0

**for** **in** range(1,4,1):

S=S+i

P=P\*S

Запишите результаты его выполнения: *S*= \_\_\_\_ *P*= \_\_\_\_

**8.**Разработайте программу, вычисляющую множество значений функции *y*=*x*3-4 для *x*, изменяющихся от 0 до 15 с шагом 3. Результат должен быть представлен в виде:

*x*=0 *y*(*x*)=…

*x*=3 *y*(*x*)=…

…………

**9.**Предлагается фрагмент программы:

S=0

j=1

**for** i **in** range(2,0,-1):

**while** (j<=3):

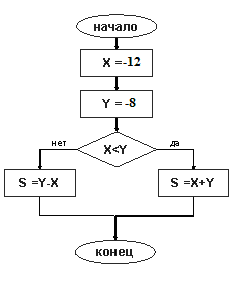
S=S-i\*j

j=j+1

Запишите результаты его выполнения:   
*S*= \_\_\_\_

**Вариант №3**

**1.** Дана блок-схема



После выполнения алгоритма значение *S* равно…

**2.** Что будет выведено на экран, после выполнения данной программы (пользователь ввел значения *a*=9, *n*=2):

a=int(input())

n=int(input())

**if** ((a%2==0)**and not**(n==2)):

print("\n%d\n"%int(a\*5))

**else:**

print("\n%d\n"%int(n\*5))

**3.** Вычислите значение выражения (a<1) **or** (a>1,2) при *a*=1.5.

**4.** Запишите условия:

**1)** *k* является нечетным числом, принадлежащим отрезку [*a*,*b*].

**2)** среди чисел *a*, *b*, *c*есть хотя бы два четных.

**5.** Предлагается фрагмент программы:

**if** ((x>=0) and (y>=0)):

**if** (x\*x+y\*y<16):

F=0

**else:**

F=1

**else:**

F=-1

Запишите результаты ее выполнения, если *x*=–4; *y*=1.

**6.**Предлагается фрагмент программы:

x=0

S=1

**while** (x>0):

S=S\*2

x=x–1

* Запишите результаты его выполнения: *x*= \_\_\_\_ *S*= \_\_\_\_
* Сколько раз выполнится цикл?

**7.**Предлагается фрагмент программы:

P=1

S=0

**for** i **in** range(1,4,1):

P=P\*S

S=S-i

Запишите результаты его выполнения: *S*= \_\_\_\_ *P*= \_\_\_\_

**8.**Разработайте программу, вычисляющую множество значений функции *y*=3*x*4-7 для *x*, изменяющихся от -7 до 12 с шагом 4. Результат должен быть представлен в виде:

*x*=-7 *y*(*x*)=…

*x*=-3 *y*(*x*)=…

……

**9.**Предлагается фрагмент программы:

P=1

i=1

**while** (i<=3):

**for** j **in** range(1,3,1):

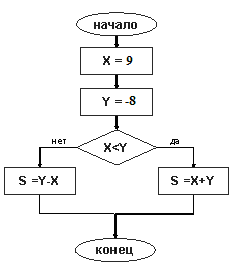
P=P\*(i+j)

i=i+1

Запишите результаты его выполнения:   
*P*= \_\_\_\_

**Вариант №4**

**1.** Дана блок-схема



После выполнения алгоритма значение S равно…

**2.** Что будет выведено на экран, после выполнения данной программы (пользователь ввел значения *a*=7, *n*=4):

a=int(input())

n=int(input())

**if** ((a%2==0) **or not**(n==2)):

print("\n%d\n"%int(a\*2))

**else:**

print("\n%d\n"%int(n\*2))

**3.** Вычислите значение выражения (b\*b-4\*a\*c<0) при *a*=2, *b*=1, *c*=-2.

**4.** Запишите условия:

**1)** *k* является трехзначным числом, кратным пяти.

**2)** из чисел *a*, *b*, *c*меньшим является *с*, а большим *b*.

**5.** Предлагается фрагмент программы:

**if** ((x>=0) **and** (y>=0)):

**if** (x\*x+y\*y<16):

F=0

**else:**

F=1

**else:**

F=-1

Запишите результаты ее выполнения, если *x*=–1; *y*=2.

**6.**Предлагается фрагмент программы:

x=5

S=1

**while** (x>0):

S=S\*x

x=x–1

* Запишите результаты его выполнения: *x*= \_\_\_\_ *S*= \_\_\_\_
* Сколько раз выполнится цикл?

**7.**Предлагается фрагмент программы:

P=1

S=0

**for** i **in** range(1,4,1)

S=S-i

P=P\*S

Запишите результаты его выполнения: *S*= \_\_\_\_ *P*= \_\_\_\_

**8.**Разработайте программу, вычисляющую множество значений функции *y*=9*x*2+7 для *x*, изменяющихся от –5 до 5 с шагом 0.5. Результат должен быть представлен в виде:

*x*=-5 *y*(*x*)=…

*x*=-4,5 *y*(*x*)=…

……

**9.**Предлагается фрагмент программы:

P=1

i=1

**while** (i<=3):

**for** j in range(2,0,-1)

P=P\*(i-j)

i=i+1

Запишите результаты его выполнения:   
*P*= \_\_\_\_

**Контрольная работа №3**

**Вариант №1**

**1.**Определите результат работы программы.

**def** f(x):

**return** x\*x-2\*x-1

a=-2

b=2

d=0.1

z=a

N=a

R=f(a)

**while** (z<b):

**if** (f(z)<R):

N=z

R=f(z)

z=z+d

printf("%f"%N)

**2.**Опишите функцию *PowerA*4(*A*), вычисляющую четвертую степень числа *A*. С помощью этой функции найдите четвертые степени трех данных чисел.

**3.**Дан программный фрагмент.

a=[]

n=int(input("Задайте количество строк и столбцов квадратной матрицы\n"))

**for** I **in** range(0,n,1):

b=[]

**for** J **in** range(0,n,1):

print("a[%d"%I,"%d"%J,"]=")

b=b+[input()]

a=a+[b]

pr=1

print(a)

**for** I **in** range(0,n,1):

mina=int(a[I][0])

for J **in** range(0,n,1):

**if** (int(a[I][J])<mina):

mina=int(a[I][J])

pr=pr\*mina

print("%d "%pr)

Определите, что будет напечатано на экране в результате его работы, если известно, что пользователь ввел 3, 1, 2, -1, 9, -18, 34, 1, 8, 16.

**4.**Дан фрагмент программы:

print("Введите строку\n")

S=input()

S1=""

S2=""

S3=""

K=""

D=""

**for** i **in** range(0,len(S)//2,1):

D=S[i]

K=S[len(S)-i-1]

**for** j **in** range(0,i,1):

S1=S1+S[j]

S1=S1+K

**for** k **in** range(i+1,len(S)-i-1,1):

S2=S2+S[k]

S2=S2+D

**for** l **in** range(len(S)-i,len(S),1):

S2=S2+S[l]

S=S1+S2+S3

print("\nРезультат: \n")

print(S)

Укажите значение переменной *S* после выполнения фрагмента программы, если на входе было введено: код.

**5.**Дан фрагмент программы:

S=input()

kol=0

**for** i **in** range(0,len(S)-1,1):

**if** ((S[i]=='о') **and** (S[i+1]=='л')):

kol=kol+1

print("%d\n"%kol)

Укажите значение переменной *kol* после выполнения фрагмента программы, если на входе было задано: молоко.

**3.**Определите значение переменной *S* для фрагмента программного кода.

S=input()

S1=""

**for** j **in** range(2,7,1):

S1=S1+S[j]

S=S1+"ы"

print(S)

Известно, что изначально пользователь ввел автоматизация.

**Вариант № 2**

**1.**Определите результат работы программы.

**def** f(x):

**return** x\*x-2\*x-1

a=-1

b=1

d=0.2

z=a

N=a

R=f(a)

**while** (z<b):

**if** (f(z)<R):

N=z

R=f(z)

z=z+d

printf("%f"%N)

**2.**Опишите функцию *Mean*(*X*,*Y*), вычисляющую среднее геометрическое двух положительных чисел *X* и *Y*. С помощью этой функции найдите среднее геометрическое для пар (*A*,*B*), (*A*,*C*), (*A*,*D*), если даны *A*, *B*, *C*, *D*.

**3.**Дан программный фрагмент.

**for** I **in** range(0,n,1):

b=[]

**for** J **in** range(0,n,1):

print("a[%d"%I,"%d"%J,"]=")

b=b+[input()]

a=a+[b]

pr=1

print(a)

**for** I **in** range(0,n,1):

mina=int(a[I][0])

for J **in** range(0,n,1):

**if** (int(a[I][J])<mina):

mina=int(a[I][J])

pr=pr\*mina

print("%d "%pr)

Определите, что будет напечатано на экране в результате его работы, если известно, что пользователь ввел 3, -4, 2, 2, -9, -18, 3, 5, 1, 2.

**4.**Дан фрагмент программы:

print("Введите строку\n")

S=input()

S1=""

S2=""

S3=""

K=""

D=""

**for** i **in** range(0,len(S)//2,1):

D=S[i]

K=S[len(S)-i-1]

**for** j **in** range(0,i,1):

S1=S1+S[j]

S1=S1+K

**for** k **in** range(i+1,len(S)-i-1,1):

S2=S2+S[k]

S2=S2+D

**for** l **in** range(len(S)-i,len(S),1):

S2=S2+S[l]

S=S1+S2+S3

print("\nРезультат: \n")

print(S)

Укажите значение переменной *S* после выполнения фрагмента программы, если на входе было введено: ром.

**5.**Дан фрагмент программы:

S=input()

kol=0

**for** i **in** range(0,len(S)-1,1):

**if** ((S[i]=='о') **and** (S[i+1]=='л')):

kol=kol+1

print("%d\n"%kol)

Укажите значение переменной *kol* после выполнения фрагмента программы, если на входе было задано: колодец.

**6.**Определите значение переменной *S* для фрагмента программного кода.

S=input()

S1=""

**for** j **in** range(0,3,1):

S1=S1+S[j]

S=S1+"та"

print(S)

Известно, что изначально пользователь ввел пинцет.

**Вариант № 3**

**1.**Определите результат работы программы.

**def** f(x):

**return** x\*x-2\*x-1

a=-3

b=3

d=0.5

z=a

N=a

R=f(a)

**while** (z<b):

**if** (f(z)<R):

N=z

R=f(z)

z=z+d

printf("%f"%N)

**2.**Опишите функцию *Triangle*(*a*), вычисляющую по стороне *a* равностороннего треугольника его площадь . С помощью этой функции найдите площади трех равносторонних треугольников с данными сторонами.

**3.**Дан программный фрагмент.

**for** I **in** range(0,n,1):

b=[]

**for** J **in** range(0,n,1):

print("a[%d"%I,"%d"%J,"]=")

b=b+[input()]

a=a+[b]

pr=1

print(a)

**for** I **in** range(0,n,1):

mina=int(a[I][0])

for J **in** range(0,n,1):

**if** (int(a[I][J])<mina):

mina=int(a[I][J])

pr=pr\*mina

print("%d "%pr)

Определите, что будет напечатано на экране в результате его работы, если известно, что пользователь ввел 3, 5, -5, -4, 3, -9, 18, 4, 2, 19.

**4.**Дан фрагмент программы:

print("Введите строку\n")

S=input()

S1=""

S2=""

S3=""

K=""

D=""

**for** i **in** range(0,len(S)//2,1):

D=S[i]

K=S[len(S)-i-1]

**for** j **in** range(0,i,1):

S1=S1+S[j]

S1=S1+K

**for** k **in** range(i+1,len(S)-i-1,1):

S2=S2+S[k]

S2=S2+D

**for** l **in** range(len(S)-i,len(S),1):

S2=S2+S[l]

S=S1+S2+S3

print("\nРезультат: \n")

print(S)

Укажите значение переменной *S* после выполнения фрагмента программы, если на входе было введено: том.

**5.**Дан фрагмент программы:

S=input()

kol=0

**for** i **in** range(0,len(S)-1,1):

**if** ((S[i]=='о') **and** (S[i+1]=='л')):

kol=kol+1

print("%d\n"%kol)

Укажите значение переменной *kol* после выполнения фрагмента программы, если на входе было задано: молоток.

**6.**Определите значение переменной *S* для фрагмента программного кода.

S=input()

S1="па"

**for** j **in** range(2,5,1):

S1=S1+S[j]

S=S1

print(S)

Известно, что изначально пользователь ввел карусель.

**Вариант № 4**

**1.**Определите результат работы программы.

**def** f(x):

**return** x\*x-2\*x-1

a=-7

b=7

d=1

z=a

N=a

R=f(a)

**while** (z<b):

**if** (f(z)<R):

N=z

R=f(z)

z=z+d

printf("%f"%N)

**2.**Опишите функцию *Rectangle*(*a*,*b*), вычисляющую по сторонам *a* и *b* прямоугольника его периметр . С помощью этой функции найдите периметры трех прямоугольников с данными сторонами.

**3.**Дан программный фрагмент.

**for** I **in** range(0,n,1):

b=[]

**for** J **in** range(0,n,1):

print("a[%d"%I,"%d"%J,"]=")

b=b+[input()]

a=a+[b]

pr=1

print(a)

**for** I **in** range(0,n,1):

mina=int(a[I][0])

for J **in** range(0,n,1):

**if** (int(a[I][J])<mina):

mina=int(a[I][J])

pr=pr\*mina

print("%d "%pr)

Определите, что будет напечатано на экране в результате его работы, если известно, что пользователь ввел 3, 5, 2, 1, -9, -4, 6, -1, -2, 20.

**4.**Дан фрагмент программы:

print("Введите строку\n")

S=input()

S1=""

S2=""

S3=""

K=""

D=""

**for** i **in** range(0,len(S)//2,1):

D=S[i]

K=S[len(S)-i-1]

**for** j **in** range(0,i,1):

S1=S1+S[j]

S1=S1+K

**for** k **in** range(i+1,len(S)-i-1,1):

S2=S2+S[k]

S2=S2+D

**for** l **in** range(len(S)-i,len(S),1):

S2=S2+S[l]

S=S1+S2+S3

print("\nРезультат: \n")

print(S)

Укажите значение переменной *S* после выполнения фрагмента программы, если на входе было введено: кот.

**5.**Дан фрагмент программы:

S=input()

kol=0

**for** i **in** range(0,len(S)-1,1):

**if** ((S[i]=='о') **and** (S[i+1]=='л')):

kol=kol+1

print("%d\n"%kol)

Укажите значение переменной *kol* после выполнения фрагмента программы, если на входе было задано: полотенце.

**6.**Определите значение переменной *S* для фрагмента программного кода.

S=input()

S1=""

**for** j **in** range(0,3,1):

S1=S1+S[j]

S=S1

print(S)

Известно, что изначально пользователь ввел пирамида.

**Контрольная работа №4**

**Вариант №1**

**1.**Дайте описание списка содержащего информацию о жителях города (фамилия; улица; дом; корпус; квартира).

**2.**Введите сведения об *N* студентах (фамилия, курс, группа, стипендия). Определите фамилии студентов, получающих самую низкую стипендию.

**3.**Дан фрагмент программы:

f = open('numbers.txt', 'r+')

sm=0

s=''

n=0

st=f.readline()

**for** i **in** range(0,len(st)-1,1):

s=s+st[i]

**if** **not**(s==''):

sm=sm+int(s)

n=n+1

**while** **not**(st==''):

s=''

st=f.readline()

**for** i **in** range(0,len(st)-1,1):

s=s+st[i]

**if not**(s==''):

sm=sm+int(s)

n=n+1

f.close

sr=sm/n

print("%f"%sr)

Укажите значение переменной *sr* после выполнения фрагмента программы, если файл C:\numbers.txt содержит числа 1, 3, 5, 7, 9 (числа записаны в файл по одному в строке).

**4.** Найдите объединение, пересечение и разность множества букв в слове mathematics и множества букв в слове geometry*.*

**5.** Найдите объединение, пересечение и разность множеств *A* и *B*, если

**1)** *A*={1, 2, 3, 4, 5, 6}, *B*={1, 3, 5};

**2)***A*={1, 2, 3, 4, 5, 6}, *B*=∅.

**Вариант №2**

**1.**Дайте описание списка, содержащего информацию об учениках школьного класса (фамилия и инициалы; дата рождения; месяц рождения; год рождения; средний балл).

**2.**Введите сведения об *N* студентах (фамилия, курс, группа, стипендия). Определите фамилии студентов, обучающихся на втором курсе.

**3.**Дан фрагмент программы:

f = open('numbers.txt', 'r+')

sm=0

s=''

n=0

st=f.readline()

**for** i **in** range(0,len(st)-1,1):

s=s+st[i]

**if** **not**(s==''):

sm=sm+int(s)

n=n+1

**while** **not**(st==''):

s=''

st=f.readline()

**for** i **in** range(0,len(st)-1,1):

s=s+st[i]

**if not**(s==''):

sm=sm+int(s)

n=n+1

f.close

sr=sm/n

print("%f"%sr)

Укажите значение переменной *sr* после выполнения фрагмента программы, если файл C:\numbers.txt содержит числа 2, 3, 7, 8 (числа записаны в файл по одному в строке).

**4.** Найдите объединение, пересечение и разность множества букв в слове informatika и множества букв в слове information.

**5.** Найдите объединение, пересечение и разность множеств *A* и *B*, если

**1)** *A*={-2, -1, 0, 1, 2}, *B*={1, 0, -1};

**2)** *A*=∅, *B*={3, 7, 9}.

**Вариант №3**

**1.**Дайте описание списка, содержащего информацию о товарах (наименование товара; страна, импортирующая товар; объем поставляемой партии в штуках; стоимость единицы).

**2.**Введите сведения об *N* студентах (фамилия, курс, группа, стипендия). Определите фамилии студентов, обучающихся в 31 группе.

**3.**Дан фрагмент программы:

f = open('numbers.txt', 'r+')

sm=0

s=''

n=0

st=f.readline()

**for** i **in** range(0,len(st)-1,1):

s=s+st[i]

**if** **not**(s==''):

sm=sm+int(s)

n=n+1

**while** **not**(st==''):

s=''

st=f.readline()

**for** i **in** range(0,len(st)-1,1):

s=s+st[i]

**if not**(s==''):

sm=sm+int(s)

n=n+1

f.close

sr=sm/n

print("%f"%sr)

Укажите значение переменной *sr* после выполнения фрагмента программы, если файл C:\numbers.txt содержит числа 11, 12, 14, 15 (числа записаны в файл по одному в строке).

**4.** Найдите объединение, пересечение и разность множества букв в слове mathematic и множества букв в слове psychology.

**5.** Найдите объединение, пересечение и разность множеств *A* и *B*, если

**1)***A*={1, 2, 3, 4, 5}, *B*={2, 4, 6};

**2)***A*={-1, 0, 1}, *B*=∅.

**Вариант №4**

**1.**Дайте описание списка, содержащего информацию о выборы (фамилия кандидата; количество набранных голосов; партия, от которой выдвинут кандидат; прошел ли процедуру выборов).

**2.**Введите сведения об *N* студентах (фамилия, курс, группа, стипендия). Определите средний размер стипендии.

**3.**Дан фрагмент программы:

f = open('numbers.txt', 'r+')

sm=0

s=''

n=0

st=f.readline()

**for** i **in** range(0,len(st)-1,1):

s=s+st[i]

**if** **not**(s==''):

sm=sm+int(s)

n=n+1

**while** **not**(st==''):

s=''

st=f.readline()

**for** i **in** range(0,len(st)-1,1):

s=s+st[i]

**if not**(s==''):

sm=sm+int(s)

n=n+1

f.close

sr=sm/n

print("%f"%sr)

Укажите значение переменной *sr* после выполнения фрагмента программы, если файл C:\numbers.txt содержит числа 3, 5, 7, 9 (числа записаны в файл по одному в строке).

**4.** Найдите объединение, пересечение и разность множества букв в слове circleи множества букв в слове square.

**5.** Найдите объединение, пересечение и разность множеств *A* и *B*, если

**1)** *A*={-6, -4, 0, 2, 6}, *B*={-4, 0, 3}; **2)** *A*=∅, *B*={2, 3, 7}.

**7. ПЕРЕЧЕНЬ УЧЕБНОЙ ЛИТЕРАТУРЫ:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № п/п | Наименование | Авторы | Место издания | Год издания | Наличие | |
| печатные издания | ЭБС (адрес в сети Интернет) |
| 1. | Программирование на языке высокого уровня Python: учебное пособие для вузов | Федоров Д.Ю. | М.: Издательство Юрайт | 2022 |  | <https://urait.ru/bcode/492920> |
| 2. | Программирование: учебник и практикум для вузов | Зыков С. В. | М.: Издательство Юрайт | 2022 |  | <https://urait.ru/bcode/489754> |
| 3. | Программирование: типовые задачи, алгоритмы, методы: учебное пособие | Златопольский Д. М. | М.: Лаборатория знаний | 2020 |  | <http://biblioclub.ru> |
| 4. | Программирование: учебно-методическое пособие | Гунько А.В. | Новосибирск: НГТУ | 2019 |  | <http://biblioclub.ru> |
| 5. | Программирование: математическая логика: учебное пособие для вузов | Швецкий М.В., Демидов М.В., Голанова А.В., Кудрявцева И.А. | М.: Издательство Юрайт | 2022 |  | <https://urait.ru/bcode/495357> |

**8. Ресурсы информационно-телекоммуникационной сети «Интернет»:**

1. «НЭБ». Национальная электронная библиотека. – Режим доступа: [http://нэб.рф/](http://www.biblioclub.ru/)

2. «eLibrary». Научная электронная библиотека. – Режим доступа: [https://elibrary.ru](https://elibrary.ru/)

3. «КиберЛенинка». Научная электронная библиотека. – Режим доступа: <https://cyberleninka.ru/>

4. ЭБС «Университетская библиотека онлайн». – Режим доступа: [http://www.biblioclub.ru/](http://www.knigafund.ru/)

5. Российская государственная библиотека. – Режим доступа: <http://www.rsl.ru/>

**9. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ИСПОЛЬЗУЕМЫЕ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ:**

В ходе осуществления образовательного процесса используются следующие информационные технологии:

- средства визуального отображения и представления информации (LibreOffice) для создания визуальных презентаций как преподавателем (при проведении занятий) так и обучаемым при подготовке докладов для семинарского занятия.

- средства телекоммуникационного общения (электронная почта и т.п.) преподавателя и обучаемого.

- использование обучаемым возможностей информационно-телекоммуникационной сети «Интернет» при осуществлении самостоятельной работы.

**9.1. Требования к программному обеспечению учебного процесса**

Для успешного освоения дисциплины, обучающийся использует следующие программные средства:

* Windows 10 x64
* MicrosoftOffice 2016
* LibreOffice
* Firefox
* GIMP

**9.2. Информационно-справочные системы (при необходимости):**

Не используются

**10. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

Для проведения занятий лекционного типа предлагаются наборы демонстрационного оборудования и учебно-наглядных пособий.

Для изучения дисциплины используется следующее оборудование: аудитория, укомплектованная мебелью для обучающихся и преподавателя, доской, ПК с выходом в интернет, мультимедийным проектором и экраном.

Для самостоятельной работы обучающихся используется аудитория, укомплектованная специализированной мебелью и техническими средствами (ПК с выходом в интернет и обеспечением доступа в электронно-информационно-образовательную среду организации).