

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования «Кабардино-Балкарский государственный
университет им. Х.М. Бербекова» (КБГУ)

Институт электроники, робототехники и искусственного интеллекта

УТВЕРЖДАЮ

Руководитель ОПОП

 Р.Ш. Тешев

« 12 » февраля 2025 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПО ДИСЦИПЛИНЕ

Б1.О.12.02 «АЛГОРИТМИЗАЦИЯ И ПРОГРАММИНГ»

Специальность

11.05.01 Радиоэлектронные системы и комплексы

Специализация

Радиоэлектронные системы передачи информации

Квалификация (степень) выпускника

Инженер

Форма обучения

Очная

Нальчик 2025

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Таблица 1

Код и формулировка компетенции	Индикаторы достижения	Планируемые результаты обучения по дисциплине (ЗУН)
<p>ОПК-9. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.</p>	<p>ОПК-9.1. Способен формировать последовательность вычислительных шагов приводящих к преобразованию</p>	<p>Знать методы алгоритмизации, языки и технологии программирования, пригодные для практического применения</p>
	<p>входных данных в выходные.</p> <p>ОПК-9.2 Способен применять современный инструментарий</p>	<p>Уметь применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач</p>
	<p>технологии программирования в специализированных программных продуктах.</p> <p>ОПК-9.3. Способен формировать программный код и компилировать его в исполняемый файл пригодный для практического применения.</p>	<p>Владеть навыками программирования, отладки и тестирования программного обеспечения..</p>

2 Шкала оценивания планируемых результатов обучения

2.1 Текущий контроль

Оценка результатов текущей успеваемости в рамках контрольных точек осуществляется посредством 70-балльной системы, при этом за добросовестное посещение занятий обучающийся может набрать до 10 баллов, за качественное прохождение оценочных мероприятий - до 60 баллов.

Таблица 2

**Карта распределения рейтинговых баллов в рамках текущего контроля
в 3 семестре**

№	Оценочное средство	Форма проведения	Порядок проведения	Максимальное количество баллов	Критерии оценивания
1	Лабораторная работа №1 «Составление алгоритмов для решения задач»	письменная	Работа включает в себя два задания, выполняется студентами попарно.	6	6- все задания выполнены верно, выводы по работе обоснованы; 4 - все задания выполнены верно, выводы по работе некорректны; 2 – задания выполнены частично или одно из заданий выполнено не верно, выводы содержат ошибки. 0 – задания не выполнены или все задания выполнены неверно
2	Лабораторная работа №2 «Основные элементы языка Python»	письменная	Работа включает в себя два задания, выполняется студентами попарно.	6	6- все задания выполнены верно, выводы по работе обоснованы; 5 - все задания выполнены верно, выводы по работе некорректны; 2 – задания выполнены частично или одно из заданий выполнено не верно, выводы содержат ошибки. 0 – задания не выполнены или все задания выполнены неверно
3	Лабораторная работа №3 «Операторы управления и функции»	письменная	Работа включает в себя два задания, выполняется студентами попарно.	6	6- все задания выполнены верно, выводы по работе обоснованы; 4 - все задания выполнены верно, выводы по работе некорректны;

					2 – задания выполнены частично или одно из заданий выполнено не верно, выводы содержат ошибки. 0 – задания не выполнены или все задания выполнены неверно
4	Лабораторная работа №4 «Указатели. Массивы».	письменная	Работа включает в себя два задания, выполняется студентами попарно.	5	5- все задания выполнены верно, выводы по работе обоснованы; 4 - все задания выполнены верно, выводы по работе некорректны; 2 – задания выполнены частично или одно из заданий выполнено не верно, выводы содержат ошибки. 0 – задания не выполнены или все задания выполнены неверно
	Лабораторная работа №5. «Работа с файлами. Структуры и классы»	письменная	Работа включает в себя два задания, выполняется студентами попарно.	5	5- все задания выполнены верно, выводы по работе обоснованы; 4 - все задания выполнены верно, выводы по работе некорректны; 2 – задания выполнены частично или одно из заданий выполнено не верно, выводы содержат ошибки. 0 – задания не выполнены или все задания выполнены неверно
11	Тесты по 1 контрольной точке	с применением	Студент проходит	6	Количество баллов пропорционально

		ДТ	компьютерное тестирование в ЭИОС.		количеству правильных ответов
12	Тесты по 2 контрольной точке	с применением ДТ	Студент проходит компьютерное тестирование в ЭИОС.	6	Количество баллов пропорционально количеству правильных ответов
14	Коллоквиум по 1 контрольной точке	письменная	Студенты отвечают письменно на вопросы коллоквиума	10	10– ответы полные, точные, демонстрируют глубокое понимание темы, аргументация логична; 6 – ответы в основном правильные, но содержат незначительные ошибки; 3- ответы недостаточно полные; 2 – ответы частичные, содержат ошибки или требуют наводящих вопросов; 1-ответы не на все вопросы, частичные. 0 – ответы отсутствуют или полностью неверные.
	Коллоквиум по 2 контрольной точке	письменная	Студенты отвечают письменно на вопросы коллоквиума	10	10– ответы полные, точные, демонстрируют глубокое понимание темы, аргументация логична; 6 – ответы в основном

					<p>правильные, но содержат незначительные ошибки;</p> <p>3- ответы недостаточно полные;</p> <p>2 – ответы частичные, содержат ошибки</p> <p>или требуют наводящих вопросов;</p> <p>1-ответы не на все вопросы, частичные.</p> <p>0 – ответы отсутствуют или полностью неверные.</p>
	Итого:			60	

Карта распределения баллов в рамках промежуточной аттестации

№	Оценочное средство	Форма проведения	Порядок проведения	Максимальное количество баллов	Критерии оценивания
1	Билеты на зачет	Устный опрос	Билет содержит 2 теоретических вопроса. На теоретические вопросы студент должен ответить устно.	Теоретические вопросы – 30 баллов.	<p>Критерии оценивания теоретических вопросов:</p> <p>25 до 30 баллов: Глубокий уровень владения материалом, точное знание ключевых концепций, способность анализировать и интерпретировать факты, грамотно строить высказывания, привести примеры,</p>

				<p>свободно оперировать терминологией. От 19 до 24 баллов: Базовое владение предметом, умение последовательно раскрыть основную мысль вопроса, грамотное применение терминов, наличие существенных элементов анализа и обобщений, но недостаточное развертывание или отдельные неточности. От 13 до 18 баллов: Частичное освоение материала, попытка объяснить основной смысл вопроса, использование некоторых базовых терминов, но отсутствие глубокого понимания сложных моментов, логические недостатки изложения, отсутствие выводов. От 7 до 12 баллов: Ошибочные представления, слабо выраженное владение основными понятиями, значительные затруднения в интерпретации вопросов, существенные фактологические ошибки, отсутствие обоснованных</p>
--	--	--	--	--

					выводов и примеров. От 0 до 6 баллов: Полное непонимание темы, неспособность сформулировать адекватный ответ, грубые ошибки, несоответствие требованиям задания.
--	--	--	--	--	---

3. Оценочные материалы для текущего и промежуточного контроля успеваемости

3.1. Оценочные материалы для текущего контроля

ЛАБОРАТОРНАЯ РАБОТА № 1

Лабораторная работа № 1

Цель работы: Приобрести навыки математической формулировки содержательной постановки задачи, выбора метода, разработки алгоритма и программы её решения.

Процесс решения задачи состоит из следующих основных этапов:

1. Постановка задачи.

Математическая формулировка задачи.

Выбор и разработка метода её решения.

Разработка алгоритма решения.

Программирование задачи.

Отладка программы на ПЭВМ.

Решение задачи по составленной программе на ПЭВМ.

Анализ результатов решения.

При непосредственном решении задач на ПЭВМ можно выделить три процедуры: ввод информации, обработка информации, вывод результата.

Наиболее наглядным способом составления алгоритма является графический, т.е. изображение алгоритма решения в виде блок-схемы.

При составлении разветвляющихся алгоритмов необходимо указать дальнейшее направление вычислительного процесса по одному из нескольких заранее определённых направлений в зависимости от выполнения некоторого логического условия.

Пример 1.1

Составить алгоритм вычисления y для любых введённых значений x .

$$y = \begin{cases} ax^2 + bx - c, & \text{если } x < -2 \\ 4x - ab, & \text{если } -2 \leq x < 3 \\ (cx - bx) / 5, & \text{если } x \geq 3 \end{cases}$$

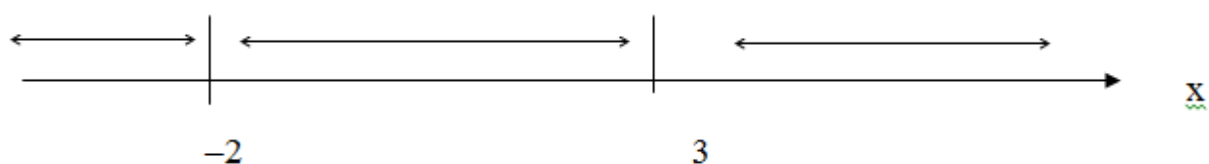
В зависимости от положения x на числовой оси значение y необходимо вычислить по следующим формулам:

В зависимости от положения x на числовой оси значение y необходимо вычислить по следующим формулам:

$$y = ax^2 + bx - c$$

$$y = 4x - ab$$

$$y = (cx - bx) / 5$$



Данные вычисления могут быть реализованы по следующей блок-схеме алгоритма (рис.1.1).

Пояснения к алгоритму. Ввод исходных данных a, b, c, x осуществляется в блоке 2. Затем в блоке 3 анализируется значение x , и если

$x < -2$, то в блоке 4 вычисляется y по формуле: $y = ax^2 + bx - c$, – и результат выводится в блоке 8. Если же $x \geq -2$, то из блока 3 идём по направлению «нет», и тогда значение x анализирует блок 5. Если $x < 3$, то из блока 5 идём по направлению «да», и значение y вычисляется по формуле: $y = 4x - ab$, – и также выводится в блоке 8. Если же $x \geq 3$, то из блока 5 идём по направлению «нет», и значение y вычисляется по формуле: $y = (cx - bx) / 5$ – в блоке 7 и в блоке 8 выводится.

Замечание. Следует обратить внимание на то, что для любого конкретного набора исходных данных вычисления, предписанные алгоритмом, пойдут только по одной ветви, хотя алгоритмом должны быть предусмотрены и определены все возможные ветви данного вычислительного процесса, т.е. обеспечиваться свойство массовости алгоритма.

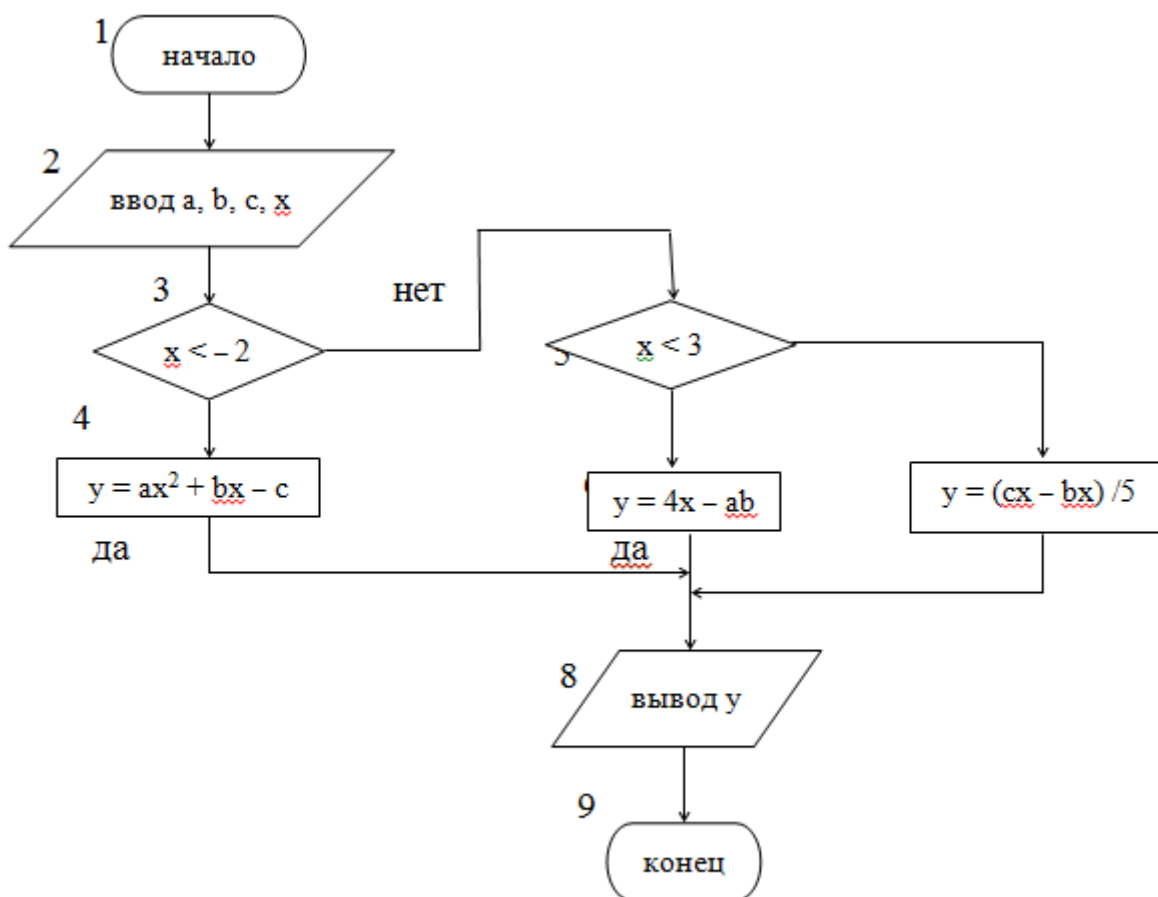


Рис. 1.1. Блок-схема алгоритма к примеру 1.1

Пример 1.2.

Составить алгоритм определения наибольшего из трёх неравных между собой чисел a , b , c .

Подумаем, какие действия необходимо выполнить для решения поставленной задачи (словесный способ записи алгоритма). Очевидно, что для определения наибольшего из трёх чисел можно предложить не один вариант.

Первый вариант. Вначале следует сравнить первые два из трёх, затем большее сравним с третьим оставшимся числом, и большее число в этом последнем сравнении и будет наибольшим из трёх. Перед выводом этого числа на печать надо сначала его значение присвоить переменной p – это позволит сократить запись операторов вывода программы. Записав приведённые ниже рассуждения с помощью известных нам блоков, получим графический алгоритм её решения.

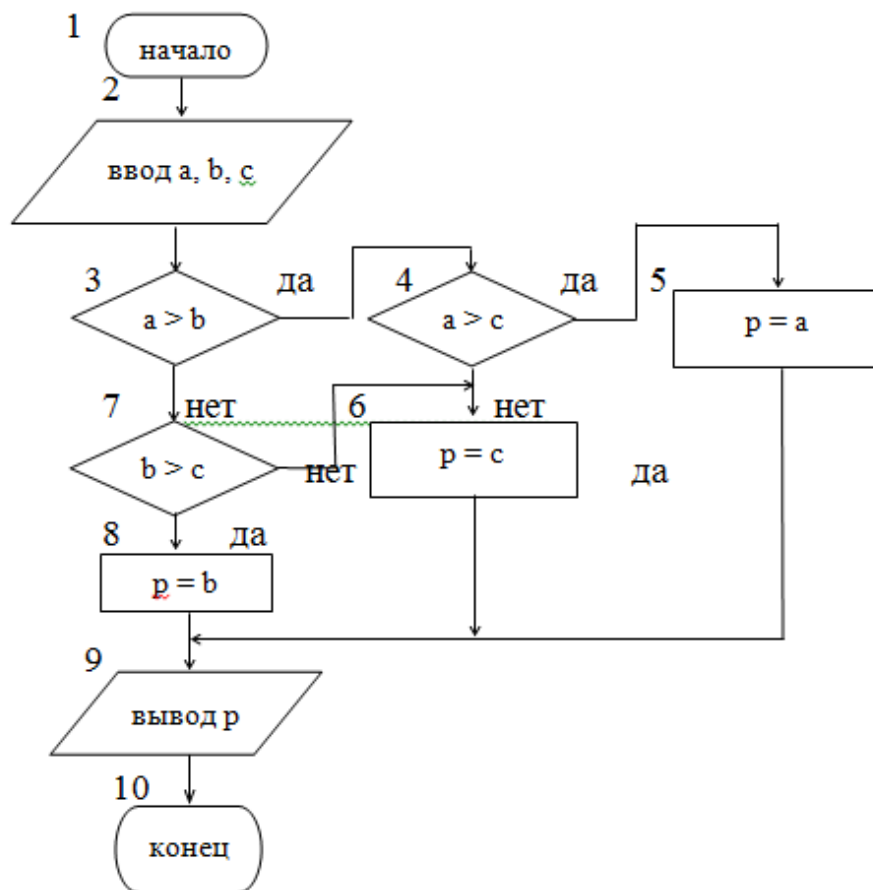
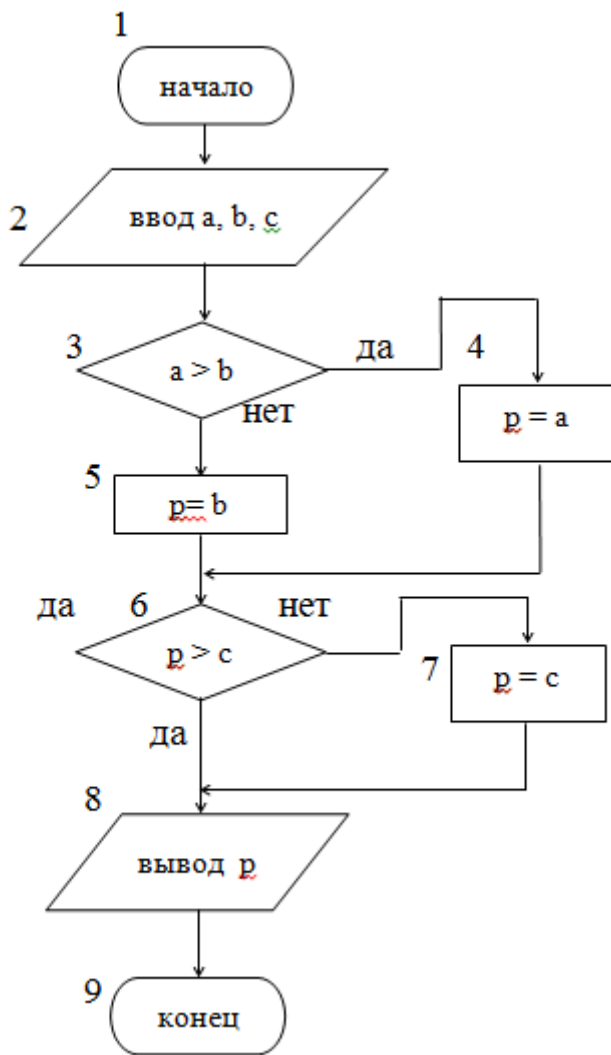


Рис. 1.2. Первый вариант блок-схемы алгоритма решения к примеру 1.2

Второй вариант. Вначале сравниваем между собой первые два числа a и b (блок 3). Далее промежуточной переменной p присваиваем значение большего из них (блоки 4 или 5), а затем сравниваем её с третьим числом (блок 6). Если третье число c окажется больше этой промежуточной переменной p , то присвоим его ей (блок 7) и далее выведем её на печать (блок 8). Блок-схема этого алгоритма представлена на рис. 1.3.



Пример 1.3.

Определить, можно ли поместить круг радиусом r в прямоугольник со сторонами a и b .

Чтобы выяснить, какой должен быть ответ при любых a , b и r , обратимся к рис. 1.4. Как видно из рисунка, круг войдёт в прямоугольник, если диаметр круга будет меньше, чем меньшая из сторон прямоугольника. Тогда, возможные варианты блок-схем алгоритма решения представлены на рис. 1.5.

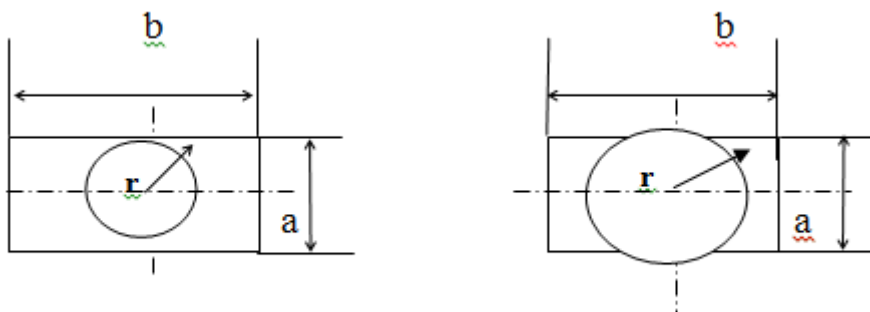


Рис. 1.4. Вписывание круга в квадрат

Как видно из рисунка, круг войдёт в прямоугольник, если диаметр круга будет меньше, чем меньшая из сторон прямоугольника. Тогда, возможны варианты блок-схем алгоритма решения представлены на

рис. 1.5.

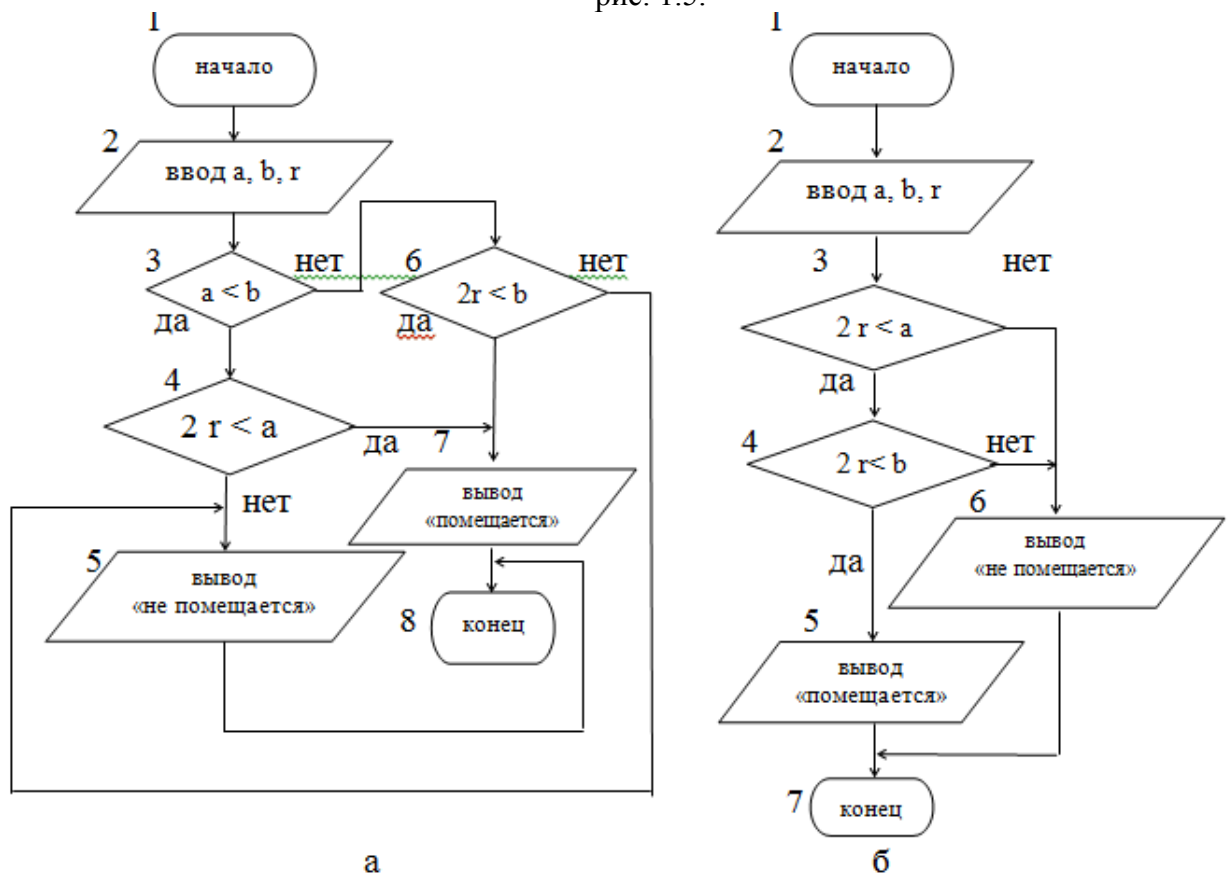


Рис. 1.5. Блок-схемы алгоритма решения к примеру 1.3

На рис. 1.5, а определяется меньшая сторона прямоугольника (блок 3), а затем проверяется, чтобы диаметр круга был меньше, чем меньшая из сторон прямоугольника (блоки 4 и 6). На рис. 1.5, б проверяется вхождение круга в прямоугольник путём поочерёдного сравнения его диаметра с каждой из сторон (блоки 3 и 4). И, если он меньше их обоих, то круг можно поместить в прямоугольник.

Индивидуальные задания к лабораторной работе № 1

1-й вариант

$$z = \begin{cases} ax^2/2 + bx - c, & \text{если } x \leq 1 \\ \min(x; 1, 5), & \text{если } 1 < x < 2 \\ 2x^2 - b, & \text{если } x \geq 2 \end{cases}$$

Проверить, является ли треугольник со сторонами a, b, c равнобедренным.

Определить, попадёт ли команда «Динамо» в лигу сильнейших (т.е. наберёт не менее 20 очков). За выигрыш команда получает 2 очка, за ничью – 1 очко, за проигрыш – 0 очков. Всего команда провела n встреч.

Из них m встреч выиграла, k – проиграла, а p завершила вничью.

2-й вариант

$$d = \begin{cases} (x - a)^2, & \text{если } x > 0 \\ 2x + a, & \text{если } x = 0 \\ \frac{a^2 - x}{8}, & \text{если } x < 0 \end{cases}$$

Определить, является ли число b наименьшим из трёх не равных между собой a , b , c .

Для доставки на перерабатывающий завод продукции совхоз её сначала перевозит автотранспортом на железнодорожную станцию, а затем по железной дороге отправляет на завод. Недалеко от совхоза имеются две железнодорожные станции: станция А находится на расстоянии a_1 км от совхоза и a_2 км от завода; станция В – на расстоянии b_1 км от совхоза и b_2 км от завода. Определить, услугами какой станции совхозу выгоднее пользоваться, если стоимость перевозки 1 т продукции на расстояние

1 км составляет для автотранспорта c_1 руб, для железнодорожного транспорта – c_2 руб.

Лабораторная работа №2

Цель работы: Научиться выполнять операции над строковыми переменными, организовывать вывод данных с использованием f- строк.

Порядок выполнения работы

1. Для строки вывести статистику по количеству входящих в нее цифр. Если цифр в строке нет, должна выводиться фраза: 'В данной строке цифр нет'

Формат ввода

Введите строку: Hello 123 ** hello 1234 мама

Формат вывода

В строке 7 цифр

2. Написать программу, которая из исходной строки оставляет только уникальные слова. Слова разделены пробелом, ввод и вывод слов в нижнем регистре. Порядок слов вывода не важен.

Формат ввода

Введите строку: мама мыла раму мыла мама папа привет

Формат вывода

Уникальные слова в введенной строке: мама мыла раму папа привет

3. Дана строка текста (кириллица) со словами через пробел. Необходимо все слова «кот» заменить на «ток». Порядок слов меняться не должен.

Формат ввода

КОТ НОС КОТ СОН КЛАД КОТ ВХОД КНИГА КОТ

Формат вывода

ТОК НОС ТОК СОН КЛАД ТОК ВХОД КНИГА ТОК

4. Написать программу для транслитерации фамилии, имени, отчества для загранпаспорта по установленным МВД РФ требованиям:

А -> A	Ж -> ZH	Н -> N	Ф -> F	Ъ -> IE
Б -> B	З -> Z	О -> O	Х -> KH	Э -> E
В -> V	И -> I	П -> P	Ц -> TS	Ю -> IU
Г -> G	Й -> I	Р -> R	Ч -> CH	Я -> IA
Д -> D	К -> K	С -> S	Ш -> SH	Ь -> не пишется
Е -> E	Л -> L	Т -> T	Щ -> SHCH	
Ё -> E	М -> M	У -> U	Ы -> Y	

Формат ввода

ПОПОВ ВАСИЛИЙ ВЯЧЕСЛАВОВИЧ

Формат вывода

POPOV VASILII VIACHESLAVOVICH

5. Написать программу для сжатия строки, в которой алгоритм работает следующим образом: string = 'ссххххтттсаахха' преобразуется в 'x4t3c1y2a3', то есть для каждого символа, следом за ним, записывается число повторений в строке. Порядок следования символов может быть произвольным.

Формат ввода

Введите строку: ссххххтттсаахха

Формат вывода

Результат сжатия строки: x6t3c3a3

Лабораторная работа № 3

Цель работы: изучить базовые, встроенные и пользовательские функции, а также функции с параметрами, возвращаемым значением и типом данных в Python

Функции в Python представляют собой фрагменты кода в блоке, который имеет назначенное имя. Функции принимают ввод, осуществляют вычисления либо какое-нибудь действие и возвращают вывод. И, разумеется, функции упрощают работу с кодом, делая возможным его повторное использование.

Базовые функции Python

Давайте рассмотрим пример функции Python, принимающей 2 параметра, а также вычисляющей сумму и возвращающей вычисленное значение:

```
#определяем и объявляем функцию
def calculate_sum(a,b):
    sum = a+b
    return sum

#инструкция, приведённая ниже, называется вызовом функции
print(calculate_sum(2,3)) # 5
```

Пользовательские функции Python

Объявление пользовательской функции осуществляется с применением ключевого слова **def**. При этом оно должно сопровождаться именем пользовательской функции:

```
def calculate_si_amount(principal, rate, time):
    interest = (principal*rate*time)/100
    return principal+interest
```

В данной функции окончательная сумма может быть рассчитана посредством использования простого процента к основной сумме. Именем функции является **Calculate_si_amount**. Что касается `principal`, `time` и `rate` — то это параметры, а функция возвращает рассчитанные данные.

Для пользовательской функции можно не принимать возвращаемые значения и параметры. На нижеследующем примере мы видим пользовательскую функцию, не принимающую никаких параметров, зато возвращающую данные.

```
from random import seed, random
from random import random

def generate_random_number():
    seed(10)
    return random()
```

Встроенные функции Python

В Python существует много встроенных функций. Одна из наиболее часто используемых — `print()`. Её работа чрезвычайно проста:

```
print("Всем привет")
print(len("Меня зовут Андрей"))
```

Ещё популярны такие функции, как `len()`, `abs()`, `sum()`, `str()`, `int()` и другие.

Параметры функции в Python

В языке программирования Python функция может иметь параметры по умолчанию:

```
def multiply(a, b=10):  
    return a*b  
  
multiply(12) # 120  
multiply(2, 3) # 6  
multiply(b=9) # Ошибка: None*9 недопустимо
```

В вышеописанной функции, когда пользователь не задает 2-й параметр `b`, он предполагает, что параметр равен 10, однако при этом нужно предоставить 1-й параметр.

Неизвестное количество параметров в функции Python

Когда в функции, допустим, четыре параметра, а для второго параметра определено значение по умолчанию, то третьему и четвертому параметрам тоже необходимо присвоить значение по умолчанию.

Когда число параметров неизвестно, тогда в определении функции в качестве одного из параметров добавляется `*args`. Данный параметр ожидает кортеж. В нашем случае звездочка (*) очень важна, т. к. название `args` просто является соглашением, то есть можно дать любое другое имя.

```
def calculate_sum(a, *args):  
    sum = a  
    for i in args:  
        sum += i  
    return sum  
  
calculate_sum(10) # 10  
calculate_sum(10, 11, 12) # 33  
calculate_sum(1, 2, 94, 6, 2, 8, 9, 20, 43, 2) # 187
```

Так же `**kwargs` ожидает словарь в качестве параметра.

```
def print_names(f1, l1, **kwargs):  
    print(f1, l1, end=' ')  
    for key in kwargs:  
        print(key, kwargs[key], end=' ')  
  
print_names("andrey", "master")
```

```
print_names("andrey", "master", alex="john", leon="elene")
# andrey master andrey master alex john leon elene
```

Фрагмент выше имеет ссылку на цикл for.

Тип данных для возвращаемого значения и параметров в Python

Определение типов данных для параметров функции в Python может быть полезным:

```
def prime_numbers(x:int) -> (int, list):
```

```
    l=[]
    for i in range(x+1):
        if checkPrime(i):
            l.append(i)
    return len(l), l
```

В нашем примере определение функции указывает, что нужен 1 параметр типа int и вернёт два значения типа list и int соответственно.

Возвращаемое значение функции в Python

Язык программирования Python даёт возможность функции возвращать несколько значений.

```
def prime_numbers(x):
```

```
    l=[]
    for i in range(x+1):
        if checkPrime(i):
            l.append(i)
    return len(l), l
```

```
no_of_primes, primes_list = prime_numbers(100)
```

В нашем случае возвращаются 2 значения. Если данная функция вызывается, то возвращаемые значения сохраняются одновременно в 2-х переменных. Если же функция не возвращает ничего, то она неявно возвращает None.

Лабораторная работа №4

Цель работы: Изучение одномерных массивов в Python.

Методы списков.

Метод	Что делает
<code>list.append(x)</code>	Добавляет элемент в конец списка

list.extend(L)	Расширяет список list, добавляя в конец все элементы списка L
list.insert(i, x)	Вставляет перед i-ым элементом значение x
list.remove(x)	Удаляет первый элемент в списке, имеющий значение x. ValueError, если такого элемента не существует
list.pop([i])	Удаляет i-ый элемент и возвращает его. Если индекс не указан, удаляется последний элемент
list.index(x, [start [, end]])	Возвращает положение первого элемента со значением x (при этом поиск ведется от start до end)
list.count(x)	Возвращает количество элементов со значением x
list.reverse()	Разворачивает список
list.copy()	Поверхностная копия списка
list.clear()	Очищает список

Задания:

1. В массиве из 20 целых чисел найти наибольший элемент и поменять его местами с первым элементом.
2. В массиве из 10 целых чисел найти наименьший элемент и поменять его местами с последним элементом.
3. В массиве из 15 вещественных чисел найти наибольший элемент и поменять его местами с последним элементом.
4. В массиве из 25 вещественных чисел найти наименьший элемент и поменять его местами с первым элементом.
5. Дан массив X, содержащий 27 элементов. Вычислить и вывести элементы нового массива Y, где $y_i = 6.85x_i^2 - 1.52$. Если $y_i < 0$, то вычислить и вывести $a = x_i^3 - 0.62$ и продолжить вычисления; если $y_i \geq 0$, то вычислить и вывести $b = 1/x_i^2$ и продолжить вычисления.
6. Дан массив F, содержащий 18 элементов. Вычислить и вывести элементы нового массива $p_i = 0.13f_i^3 - 2.5f_i + 8$. Вывести отрицательные элементы массива P.
7. Вычислить и вывести элементы массива Z, где $z_i = i^2 + 1$, если i – нечетное, и $z_i = 2i - 1$, если i – четное. Сформировать и вывести массив D: $d_i = 2.5z_i$, если $z_i < 2.5$ и $d_i = z_i / 2.5$, если $z_i \geq 2.5$.
8. Заданы массивы D и E. Вычислить и вывести значения $f_i = (2d_i + \sin e_i) / d_i$, где $i = 1, 2, \dots, 16$; вывести $1 < f_i < 3$.
9. В массиве R, содержащем 25 элементов, заменить значения отрицательных элементов квадратами значений, значения положительных увеличить на 7, а нулевые значения оставить без изменения. Вывести массив R.
10. Дан массив A целых чисел, содержащий 30 элементов. Вычислить и вывести сумму тех элементов, которые кратны 5.
11. Дан массив A целых чисел, содержащий 30 элементов. Вычислить и вывести сумму тех элементов, которые нечетны и отрицательны.
12. Дан массив A целых чисел, содержащий 30 элементов. Вычислить и вывести сумму тех элементов, которые удовлетворяют условию $|a_i| < i^2$.
13. Дан массив A целых чисел, содержащий 30 элементов. Вычислить и вывести количество и сумму тех элементов, которые делятся на 5 и не делятся на 7.
14. Дан массив A вещественных чисел, содержащий 25 элементов. Вычислить и вывести число отрицательных элементов и число членов, принадлежащих отрезку [1,2].
15. Дан массив C, содержащий 23 элемента. Вычислить и вывести среднее арифметическое всех значений $c_i > 3.5$.

16. Дан массив Z целых чисел, содержащий 35 элементов. Вычислить и вывести $R=S+P$, где S – сумма четных элементов, меньших 3, P – произведение нечетных элементов, больших 1.

17. Дан массив Q натуральных чисел, содержащий 20 элементов. Найти и вывести те элементы, которые при делении на 7 дают остаток 1, 2 или 5.

18. Дан массив Q натуральных чисел, содержащий 20 элементов. Найти и вывести те элементы, которые обладают тем свойством, что корни уравнения $q_i^2+3q_i-5=0$ действительны и положительны.

19. Дан массив, содержащий 10 элементов. Вычислить произведение элементов, стоящих после первого отрицательного элемента. Вывести исходный массив и результат вычислений.

20. Дан массив, содержащий 14 элементов. Вычислить сумму элементов, стоящих до первого отрицательного элемента. Вывести исходный массив и результат вычислений.

21. Дан массив содержащий 12 элементов. Все четные элементы сложить, вывести массив и результат.

22. Дан массив, содержащий 15 элементов. Все положительные элементы возвести в квадрат, а отрицательные умножить на 2. Вывести исходный и полученный массив.

23. Дан массив, содержащий 14 элементов. Все отрицательные элементы заменить на 3. Вывести исходный и полученный массив.

24. Дан массив из 15 целых чисел. Найти количество нечетных положительных элементов.

25. Массив задан датчиком случайных чисел на интервале $[-33, 66]$. Найти наименьший нечетный элемент.

26. Разработать программу, выводящую количество максимальных элементов в массиве из пятидесяти целочисленных элементов.

Лабораторная работа №5

Цель работы: изучение основных функций при работе с текстовыми файлами и получения практических навыков.

Порядок выполнения работы

Для формирования шаблонных данных, необходимо подключить библиотеку NumPy.

Ввод:

```
# Подключение необходимых библиотек
import pandas as pd
from numpy import random
import matplotlib.pyplot as plt
import sys

%matplotlib inline
plt.rc('font', family='Arial')

print("Python version", sys.version)
print("Pandas version", pd.__version__)
```

Вывод:

```
Python version 3.4.3 |Continuum Analytics, Inc.| (default, Jan 6 2015, 12:08:17) [MSC v.1600 32 bit (Intel)]
Pandasversion 0.16.1
```

2 Создание данных

Набор данных будет состоять из 1000 имен и количества рожденных детей с таким именем в России в 2015 году. Мы будем дублировать имена, чтобы одно и то же имя можно было встретить несколько раз.

Ввод

```
# Начальный набор данных
names = ['Саша', 'Маша', 'Гоша', 'Тоша', 'Антоша']
```

Для того чтобы создать 1000 имен сделаем следующее: Возьмем случайное число между 0 и 4, при помощи функций `seed`, `randint`, `len`, `range` и `zip`. `seed(500)` – создает начальное заполнение генератора случайных чисел. `randint(low=0,high=len(names))` - генерируется случайное целое число от 0 до 4; `names[n]` –получаем значение по индексу `n` из набора `for i in range(n)` – цикл по `n` элементам с шагом 1; `random_names[:n]`- созданная нами функция, возвращающая `n` случайных имен.

Ввод:

```
random.seed(500)
random_names =
[names[random.randint(low=0,high=len(names))] for i in
range(1000)]
```

```
# Вывод на экран первых 10 имен
random_names[:10]
```

Вывод:

```
['Гоша',
 'Маша',
 'Маша',
 'Саша',
 'Маша',
 'Маша',
 'Маша',
 'Гоша',
 'Гоша',
 'Гоша']
```

Далее, создадим набор из 10 случайных чисел от 0 до 1000.

Ввод:

```
births = [random.randint(low=0,high=1000) for i in
range(1000) ]
births[:10]
```

Вывод:

[968, 155, 77, 578, 973, 124, 155, 403, 199, 191]

Вопросы для тестирования **Вопросы, выносимые на коллоквиумы**

1. Геометрическая фигура ромб используется в блок-схемах для обозначения:
 - а) принятия решения
 - б) начала или конца алгоритма
 - в) ввода или вывода

2. Геометрическая фигура прямоугольник используется в блок-схемах для обозначения:
 - а) принятия решения
 - б) выполнения действия
 - в) ввода или вывода

3. Алгоритм называется линейным, если:
 - а) ход его выполнения зависит от истинности тех или иных условий
 - б) представлен в табличной форме
 - в) его команды выполняются в порядке следования друг за другом

4. Алгоритм:
 - а) последовательность действий, которая приводит к решению задачи
 - б) набор команд для компьютера
 - в) ориентированный граф, указывающий порядок выполнения команд

5. Наибольшей наглядностью обладают следующие формы записи алгоритмов:
 - а) рекурсивные
 - б) словесные
 - в) графические

6. Как называется свойство алгоритма, означающее, что данный алгоритм применим к решению целого класса задач:
 - а) определенность
 - б) массовость
 - в) понятность

7. Если алгоритм предназначен для исполнения техническим устройством, например станком с числовым программным управлением или компьютером, он представляется в виде:
 - а) процессора

- б) файлов
 - в) программы
8. Формульно-словесный способ записи алгоритма характеризуется тем, что описание осуществляется с помощью:
- а) слов
 - б) цифр
 - в) специальных знаков
9. Формульно-словесный способ записи алгоритма характеризуется тем, что описание осуществляется с помощью:
- а) аксиом
 - б) специальных знаков
 - в) формул
10. Алгоритм, в котором все действия выполняются последовательно друг за другом и только один раз:
- а) одиночный алгоритм
 - б) линейный алгоритм
 - в) не повторяющийся алгоритм
11. Специальное средство, предназначенное для записи алгоритмов в аналитическом виде:
- а) алгоритмические языки
 - б) алгоритмические навыки
 - в) алгоритмические эксперименты
12. Перевод программ с языка высокого уровня на язык более низкого уровня обеспечивает программа :
- а) паскаль
 - б) ассемблер
 - в) компилятор
13. Когда необходимо составлять блок-схему программы:
- а) До начала составления самой программы
 - б) В процессе составления программы
 - в) После составления программы
14. Языком высокого уровня является:
- а) Ассемблер
 - б) Фортран
 - в) Макроассемблер
15. Выберите, какой метод применяется для поиска в упорядоченных массивах:
- а) бинарный поиск
 - б) прямой выбор
 - в) прямой обмен

16. Определите, если число повторений цикла известно и задано наибольшее допустимое значение n , то лучше использовать:
- а) цикл с предусловием
 - б) цикл с постусловием
 - в) цикл со счетчиком
17. Как называется набор однотипных данных, имеющий общее для всех своих элементов имя:
- а) множество
 - б) массив
 - в) запись
18. Определите, как называется процесс перестановки элементов массива с целью упорядочивания их в соответствии с каким-либо критерием:
- а) поиск
 - б) перебор
 - в) сортировка+
19. Выясните, в основе какого метода сортировки лежит обмен соседних элементов массива:
- а) прямой обмен
 - б) прямой выбор
 - в) прямой вариант
20. Как называется свойство алгоритма, означающее, что он всегда приводит к результату через конечное, возможно, очень большое, число шагов:
- а) дискретность
 - б) определённости
 - в) результативность
21. Как называется свойство алгоритма, означающее, что он задан с помощью таких предписаний, которые исполнитель может воспринимать и по которым может выполнять требуемые действия:
- а) массовость
 - б) понятность
 - в) определённости
22. Выберите верные утверждения о языке программирования Python.
- а) высокоуровневый,
 - б) низкоуровневый,
 - в) интерпретируемый,
 - г) компилируемый.
23. Что хранит в себе переменная?

Выберите один из 4 вариантов ответа:

- 1) имя
- 2) значение
- 3) тип
- 4) длину своего значения

24. Что обозначает тип данных int?

Выберите один из 4 вариантов ответа:

- 1) целочисленное
- 2) вещественное
- 3) строковое
- 4) логическое

25. Выберите правильную запись оператора присваивания:

Выберите один из 4 вариантов ответа:

- 1) 10=x
- 2) y=7,8
- 3) a=5
- 4) a==x+y

26. Укажите оператор ввода:

Выберите один из 4 вариантов ответа:

- 1) input ()
- 2) print ()
- 3) int ()
- 4) random ()

27. Сколько возможных значений у переменной типа bool?

Выберите один из 4 вариантов ответа:

- 1) 2
- 2) 4
- 3) 8
- 4) Сколько угодно

28. Какой оператор здесь используется?

```
if n < 100:  
    b = n + a
```

Выберите один из 4 вариантов ответа:

- 1) Условный оператор
- 2) Оператор присваивания
- 3) Оператор ввода
- 4) Оператор умножения

29. Дана программа:

```
a, b, c = map(int, input().split())  
s = a+b*c  
print(s)
```

Что выведет программа, если с клавиатуры будут введены числа 4 12 2

Запишите число:

_____50_____

30. В какой строке допущена ошибка?

1. a = int(input)
2. if a>0:
3. print(a):
4. else:
5. print(A)

Если таких строк несколько, то укажите их номера в порядке возрастания без пробелов и разделителей.

Запишите число:

_____1345_____

31. Автомат получает на вход четырехзначное число. По этому числу строится новое число по следующим правиламЖ

- 1) Складываются первая и третья, а также вторая и четвертая цифры.
- 2) Полученные два числа записываются друг за другом в порядке невозрастания без разделителей.

Найдите наименьшее число, при обработке которого автомат выдает 175.

Запишите число:

_____1098_____

32. Что будут выведено после выполнения программы:

a=26

b=7

b=a//b

c=a%b

print (c)

Запишите число:

_____1_____

33. Язык программирования Python подходит для разработки:

- a. Компьютерных и мобильных приложений
- b. Аналитика и машинное обучение
- c. Игр
- d. Ничего из этого.

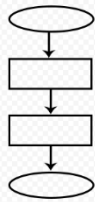
34. Назовите тип алгоритма:

- a. Разветвляющийся
- b. Линейный
- c. Циклический
- d. Смешанный

35. Язык программирования Python подходит для разработки:

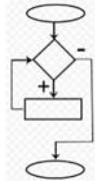
- a. Компьютерных и мобильных приложений
- b. Аналитика и машинное обучение
- c. Игр
- d. Ничего из этого.

36. Назовите тип алгоритма:



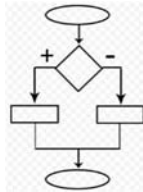
- a. Разветвляющийся
- b. Линейный
- c. Циклический
- d. Смешанный

37. Назовите тип алгоритма:



- a. Линейный
- b. Разветвляющийся
- c. Циклический
- d. Смешанный

38. Назовите тип алгоритма:



- a. Разветвляющийся
- b. Линейный
- c. Смешанный
- d. Циклический

39. Что хранит в себе переменная?

- a. Имя
- b. Значение
- c. Тип
- d. Длину своего значения

40. Что обозначает тип данных int?

- a. Целочисленное
- b. Вещественное
- c. Строковое
- d. Булево

41. Выберите правильную запись оператора присваивания:

- a. $10 = x$
- b. $y = 7,8$
- c. $a = 5$
- d. $a == b + x$

42. Укажите оператор ввода:

- a. input()
 - b. print()
 - c. int()
 - d. random()
43. Сколько возможных значений у переменной типа bool?
- a. 2
 - b. 4
 - c. 10
 - d. Сколько угодно
44. Какой оператор здесь используется?
If n < 100:
 b = n + a
- a. Условный оператор
 - b. Оператор присваивания
 - c. Оператор сложения
 - d. Оператор умножения
45. Что лучше использовать для множественного ветвления?
- a. if – elif – else
 - b. Много if
 - c. if – else – elif
 - d. while
46. Оператор цикла в языке Python:
- a. while
 - b. for
 - c. if
 - d. print
47. Сколько раз произойдет итерация цикла? (Итерация – единичное выполнение тела цикла)
print(“Осталось”, total)
- ```
total = 100
i = 0
while i < 0:
 n = int(input())
 total = total - n
 i = i + 1
```
- a. 4
  - b. 5
  - c. 6
  - d. 0
48. Для чего нужен оператор break?
- a. Для завершения программы
  - b. Для выхода из цикла
  - c. Для поломки компьютера

- d. Для удаления программы
49. Где находятся параметры, а где аргументы функции?
- a. Параметры пишутся при объявлении функции, аргументы при вызове
  - b. Аргументы пишутся при объявлении функции, параметры при вызове
  - c. Это одно и то же!
  - d. У функции есть только параметры
50. Что делает функция len()?
- a. Возвращает длину строки
  - b. Возвращает случайное число
  - c. Возвращает номер символа
  - d. Возвращает модуль числа
51. Как добавить модуль в программу?
- a. `import math`
  - b. `import math()`
  - c. `import (math)`
  - d. `import.math`
52. На каких операционных системах может работать Python?
- a. Windows
  - b. Linux
  - c. macOS
  - d. Ничего из этого
53. От чего язык программирования называется «Питон»?
- a. В честь змеи
  - b. В честь ТВ-шоу
  - c. В честь игры
  - d. В честь блюда
54. Сколько уже лет языку программирования Python?
- a. 30
  - b. 19
  - c. 7
  - d. 15
55. Создатель языка программирования Python
- a. Гвидо Ван Россум
  - b. Дэвид Паттерсон
  - c. Эрвин Дональд Кнут
  - d. Джеймс Артур Гослинг
56. `a = 345`. Что выведет команда `print(//100)`
- a. 3
  - b. 5
  - c. 4
  - d. 34
57. Выберите циклический алгоритм
- a. `k = 0`  
`while k < 10:`

```
print("Привет")
k += 1
b. a = int(input())
 b = int(input())
 c = int(input())
 s = a+b+c
 print(c)
c. a = int(input())
 if a > 0:
 print(a)
 else:
 print(a)
```

58. В какой строке правильно записан ввод числа с клавиатуры?

|                                                                                                                                                                |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"><li>1. a = int(input())</li><li>2. b=input(int())</li><li>3. c=int(input())</li><li>4. s=a+b+c</li><li>5. print(s)</li></ol> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|

- a. 1
- b. 2
- c. 3
- d. 4
- e. 5

59. В какой строке допущена ошибка?

|                                                                                                                                                    |
|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"><li>1. a = int(input())</li><li>2. if a&gt;0:</li><li>3. print(a):</li><li>4. else</li><li>5. print(A)</li></ol> |
|----------------------------------------------------------------------------------------------------------------------------------------------------|

- a. 1
- b. 2
- c. 5
- d. 4

60. Что такое «else»?

- a. Так как
- b. Иначе
- c. Если
- d. Потому что

61. Сколько раз программа напишет слово «Пока»?

```
k=0
while k<10:
 print(«Привет»)
 k += 1
```

- a. 9
- b. 0
- c. 10
- d. Бесконечно

62. Каким языком является Python?

- a) аспектно-ориентированный
- б) логический
- в) функциональный
- г) объектно-ориентированный

63. К какому уровню относится язык Python?

- a) к высокому
- б) к низкому
- в) к среднему
- г) нет правильного ответа

64. Основные архитектурные черты

- a) динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса
- б) динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных
- в) арифметические операции с плавающей точкой относятся к небезопасному коду  
некоторые особенности синтаксиса
- г) все выше сказанное

65. В каком году был основан язык программирования Python?

- a) 1992

б) 1991

в) 1990

г) 1993

66. По какой команде выдается философия Python?

а) `import this`

б) `import thas`

в) `import for me`

г) `import`

67. Присваивание в Python обозначается знаком

а) «+»

б) «-»

в) «=»

г) «\*»

68. Сколькими величинами представлены логические значения в Python?

а) двумя

б) одной

в) тремя

г) пятью

69. Как вычисляется длина строки?

а) `s1 + s2`

б) `len(s)`

в) `s[i:j:k]`

г) `s[i]`

70. Что такое кортеж в Python?

- а) набор разнородных элементов
- б) обработка данных, выполняемая, в основном, средствами вычислительной техники
- в) специалист, отвечающий за нормальное функционирование и использование ресурсов автоматизированной системы и/или вычислительной сети
- г) одно из альтернативных решений, разработанных с одинаковыми условиями поставленной задачи

71. Чем задается кортеж?

- а) вычитаем в квадратных скобках через точку
- б) умножением в круглых скобках через точку
- в) перечислением в круглых скобках через запятую
- г) делением без скобок через запятую

72. Для кортежа из одного элемента обязательно добавлять ... после имени или значения этого элемента

- а) точку
- б) восклицательный знак
- в) тире
- г) запятую

73. Функция `len (t)` отвечает за:

- а) n-кратное повторение кортежа `t`
- б) Определяется количество элементов кортежа `t`
- в) Объединение кортежей
- г) Выбор из `t` элемента с номером `i`

74. Самый простой способ сформировать список

- а) перечисление в квадратных скобках через запятую
- б) перечисление в квадратных скобках через точку
- в) деление элементов
- г) умножение элементов в круглых скобках

75. Для чего используется функция `zip()`?

а) используется для применения одной и той же операции к элементам одного или нескольких списков или кортежей.

б) можно применять и к кортежам, а также «смешивать» в её аргументах списки и кортежи

в) Для списков и кортежей, состоящих только из чисел

г) делит строку по заданному символу-разделителю

76. Используется для применения одной и той же операции к элементам одного или нескольких списков или кортежей

а) `zip()`

б) `map()`

в) `sum()`

г) `list()`

77. Для преобразования строки или кортежа в список используется функция

а) `zip()`

б) `map()`

в) `sum()`

г) `list()`

78. Функция (метод) `split()`

а) делит строку по заданному символу-разделителю и создаёт список из фрагментов строки

б) формирует строку из элементов списка

в) создаёт список как числовую арифметическую прогрессию

г) можно применять и к кортежам, а также «смешивать» в её аргументах списки и кортежи

79. Создаёт список как числовую арифметическую прогрессию

а) `zip()`

б) `map()`

в) range ()

г) list()

80. Эталонной реализацией Python является интерпретатор

а) C

б) CPython

в) C++

г) Java

81. Кто является разработчиком Python?

а) Гвидо ван Россум

б) Джеймс Гослинг

в) Бьёрн Страуструп

г) Никлаус Вирт

### **Вопросы, выносимые на коллоквиум**

#### **Коллоквиум 1**

1. Основные понятия алгоритмизации.
2. Свойства алгоритма.
3. Общие принципы построения алгоритмов.
4. Основные алгоритмические конструкции: линейные, разветвляющиеся, циклические.
5. Логические основы алгоритмизации.
6. Основные базовые и структурированные типы данных, их характеристика
7. Развитие языков программирования. Обзор языков программирования.
8. Области применения языков программирования. Стандарты языков программирования.
9. Среда проектирования. Компиляторы и интерпретаторы. Жизненный цикл программы. Программа. Программный продукт и его характеристики.
10. Основные этапы решения задач на компьютере.
11. Типы данных. Простые типы данных. Производные типы данных.
12. Структурированные типы данных

#### **Коллоквиум 2**

- 1.Операции и выражения. Правила формирования и вычисления выражений.
- 2.Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор.
- 3.Условный оператор. Оператор выбора.
- 4.Цикл с постусловием. Цикл с предусловием.
5. Цикл с параметром. Вложенные циклы.
- 6.Массивы. Двумерные массивы. Строки.
- 7.Стандартные процедуры и функции для работы со строками
- 8.Структурированный тип данных – множество.
- 9.Операции над множествами. Комбинированный тип данных – запись.
10. Файлы последовательного доступа. Файлы прямого доступа
- 11.Общие сведения о подпрограммах. Определение и вызов подпрограмм.
- 12.Область видимости и время жизни переменной. Механизм передачи параметров.
- 13.Организация функций. Рекурсия. Программирование рекурсивных алгоритмов. Основы структурного программирования. Методы структурного программирования.
14. Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы. Стандартные модули
15. История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. Классы объектов. Компоненты и их свойства.

## **1.2 Оценочные материалы для промежуточной аттестации**

Зачет проводится по билетам. В каждом билете 2 теоретических вопроса.

### **Вопросы к зачету**

- 1.Основные понятия алгоритмизации.
2. Свойства алгоритма.
- 3.Общие принципы построения алгоритмов.
4. Основные алгоритмические конструкции: линейные, разветвляющиеся, циклические.
5. Логические основы алгоритмизации.
6. Основные базовые и структурированные типы данных, их характеристика
7. Развитие языков программирования. Обзор языков программирования.
8. Области применения языков программирования. Стандарты языков программирования.

9. Среда проектирования. Компиляторы и интерпретаторы. Жизненный цикл программы. Программа. Программный продукт и его характеристики.
10. Основные этапы решения задач на компьютере.
11. Типы данных. Простые типы данных. Производные типы данных.
12. Структурированные типы данных
13. Операции и выражения. Правила формирования и вычисления выражений.
14. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор.
15. Условный оператор. Оператор выбора.
16. Цикл с постусловием. Цикл с предусловием.
17. Цикл с параметром. Вложенные циклы.
18. Массивы. Двумерные массивы. Строки.
19. Стандартные процедуры и функции для работы со строками
20. Структурированный тип данных – множество.
21. Операции над множествами. Комбинированный тип данных – запись.
22. Файлы последовательного доступа. Файлы прямого доступа
23. Общие сведения о подпрограммах. Определение и вызов подпрограмм.
24. Область видимости и время жизни переменной. Механизм передачи параметров.
25. Организация функций. Рекурсия. Программирование рекурсивных алгоритмов. Основы структурного программирования. Методы структурного программирования.
26. Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы. Стандартные модули
27. История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. Классы объектов. Компоненты и их свойства.