Задание 1.

Написать программу для вычисления суммы, имеющей своими слагаемыми a_n.

a) первые m слагаемых данной суммы (m вводится);

b) вычислять до тех пор пока $|a_n| \geq e$ (e вводится)

Пример (для пункта а). В секретной лаборатории выводят полезные бактерии. Экспериментально было установлено, что количество бактерий (в млн) завит от номера дня, в который проводится эксперимент следующим образом: $a_n = \frac{n^3}{\sqrt{n^3} - n + 1}$. Определите, сколько бактерий вывели за `m` дней.

Этапы выполнения задания.

I. Исходные данные: `m` (число дней).

II. Результат: `s` (общее количество бактерий).

III. Алгоритм решения задачи.

1. Ввод числа `m`.
2. Для вычисления общего количества бактерий необходимо последовательно прибавлять количество бактерий, выведенных в текущий день, к уже полученному количеству бактерий. Начальное значение суммы равно 0.
3. Так как количество дней заранее известно, для вычисления суммы можно воспользоваться циклом **`for`**.
4. Количество бактерий в текущий день будем хранить в переменной `a`. Значение `a` зависит от значения `n` —

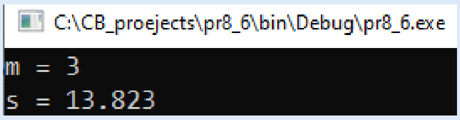
V. Программа:

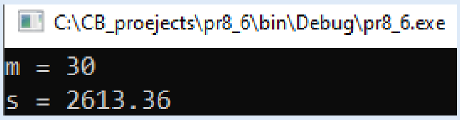
```
import math

m = int(input('m = '))
s = 0
for n in range(1, m + 1):
    x = 1. * n ** 3
    a = x / (math.sqrt(x) - n + 1)
    s += a

print(f's = {s :.2f}')
```

VI. Тестирование





VII. Анализ результата. Для проверки правильности результата можно посчитать значение суммы на калькуляторе:

$a_1 = 1;$

$a_2 = \frac{2^3}{\sqrt{8} - 2 + 1} \approx 4.38;$

$a_3 = \frac{3^3}{\sqrt{27} - 3 + 1} \approx 8.44,$

$S \approx 13.82$

счетчика дней. Переменная `n` изменяется от 1 до `m`.

5. Вывод результата `s`.

IV. Описание переменных: `m`–`int`, `s`, `a`–`float`.

Пример (для пункта b). Написать программу для вычисления суммы, имеющей своими слагаемыми элементы последовательности $a_n = \frac{n+2^n}{n!}$. Вычисления производить до тех пор, пока не найдется слагаемое, для которого верно неравенство $|a_n| < eps$. Значение *eps* вводится ($0 < eps < 1$).

Этапы выполнения задания.

I. Исходные данные: `eps` (точность вычислений).

II. Результат: переменная `s` (сумма).

III. Алгоритм решения задачи.

1. Ввод числа `eps`.
2. Для вычисления суммы необходимо последовательно прибавлять очередное слагаемое, удовлетворяющее условию задачи, к уже полученной сумме.

2.1 Так как количество слагаемых заранее не известно, то для вычисления суммы воспользуемся циклом **while**.

2.2 Начальное значение суммы s = 0.

2.3 Текущее значение слагаемого будем хранить в переменной `a`. Значение `a` зависит от значения `n` — счетчика слагаемых в сумме.

Пример 6.16.

V. Программа:

```
eps = float(input('eps = '))

n = 1
f, d = 1, 2

a = (n + d) / f
s = a

while abs(a) >= eps:
    n += 1
    f, d = f * n, d * 2

    a = (n + d) / f
    s += a

print(f's = {s :.2f}')
```

VI. Тестирование

```
C:\CB_proejects\pr8_8\bin\Debug\pr8_8.exe
eps = 0.1
summa= 9.07222
```

VII. Анализ результата. Поскольку факториал является чрезвычайно быстро растущей функцией, то элементы последовательности убывают. Выпишем элементы:

$a_1 = \frac{1+2}{1} = 3;\ a_2 = \frac{2+2^2}{2!} = 3;$

$a_3 = \frac{3+2^3}{3!} = \frac{11}{6} \approx 1{,}83;$

$a_4 = \frac{4+2^4}{4!} = \frac{20}{24} \approx 0{,}83;$

$a_5 = \frac{5+2^5}{5!} = \frac{37}{120} \approx 0{,}31;$

$a_6 = \frac{6+2^6}{6!} = \frac{70}{720} \approx 0{,}097.$

Шестой элемент меньше 0.1. Это последнее слагаемое в сумме. Сумма первых шести элементов — ≈9.07.

Если `eps` = 0.01, то к сумме, полученной

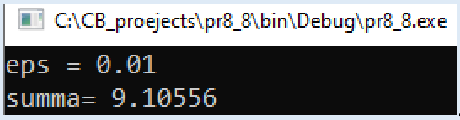
Переменная n будет изменяться на 1 до тех пор, пока выполняется условие задачи. Начальное значение `n = 0`.

2.4 Отдельно нужно получать 2^n и `n!` Для этого заведем переменные `d` (для хранения значения 2^n) и `f` (для хранения значения `n!`). Начальные значения переменных: `d = 1`, `f = 1`. Для каждого следующего слагаемого предыдущее значение переменной `d` увеличивается в два раза, значение переменной `f` увеличивается в `n` раз.

3. Вывод результата `s`.

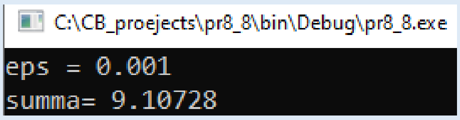
IV. Описание переменных: `n – int`, `eps, a, s, d, f – float`.

для `eps` = 0.1, будут добавляться слагаемые, которые меньше 0.1, которые незначительно изменят значение суммы.



.

Разница в значениях суммы – ≈0.03. Чем меньше точность, тем меньше будут отличаться суммы.



Индивидуальные задания.

1. a) $a_n = \frac{1}{n^3}$; b) $a_n = \frac{1}{4n}$;

2. a) $a_n = \frac{n^2-3}{n^2+n}$ b) $a_n = \frac{1}{5^n}$

3. $a_n = \frac{n^2-3n+7}{n^2+n+2}$ b) $a_n = \frac{2+n}{3^n}$

4. a) $a_n = \frac{n^2}{n^3+1}$; b) $a_n = \frac{n+1}{n!}$;

5. a) $a_n = \frac{\sqrt{n}}{n^3+8}$; b) $a_n = \frac{2^n}{3^n}$;

6. a) $a_n = \frac{2n+1}{2n^3+5}$; b) $a_n = \frac{2^n}{n!}$;

7. a) $a_n = \frac{\sqrt{n+1}}{n+3}$ b) $a_n = \frac{2^n+n}{5^n}$

8. a) $a_n = \frac{2\cos n}{(n+1)^2}$; b) $a_n = \frac{3^n}{n!+2}$;

9. a) $a_n = \frac{\sin(n+1)}{n^2}$; b) $a_n = (-1)^n \frac{3n}{4^n+7}$;

10. a) $a_n = \frac{\sqrt{n^3}}{n^2+n+3}$; b) $a_n = (-1)^n \frac{3^n+1}{n!+5}$;

11. a) $a_n = \frac{\sqrt{n^2+1}}{n^3+4n}$ b) $a_n = \frac{2}{3^n} + \frac{2}{n!}$

12. a) $a_n = \frac{\sqrt[4]{n+3}}{n^3+4n}$; b) $a_n = (-1)^n \frac{2^{3n}}{3^n+n!}$.

Задание 2.

Пример. Вывести на экран таблицу значений функции $y = x^2 \sin x$. Количество значений вводится. Начальное значение x = –3, значения аргумента выводятся с шагом h = 0.5.

Этапы выполнения задания

I. Исходные данные: k (количество точек).

II. Результат: k значений аргумента и соответствующих им значений функции.

III. Алгоритм решения задачи.

1. Ввод числа k.
2. Для получения таблицы нужно в цикле вычислять и выводить значение аргумента и соответствующие ему значение функции.
   2.1 начальное значение аргумента x = –3. Для получения очередного значения аргумента нужно к текущему значению прибавить шаг *h*.
   2.2 значение функции вычисляется по формуле $y = x^2 \sin x$.
3. Поскольку количество точек известно, воспользуемся циклом **for**.

IV. Описание переменных: k – int, x, y, h – float.

V. Программа:

```
import math

k = int(input('k = '))
x0, h = -3, 0.5

print('|    x    |    y    |')

for i in range(k):
    x = x0 + i * h
    y = x ** 2 * math.sin(x)

    print(f'| {x :7.2f} |
{y :7.2f} |')
```

VI. Тестирование.

```
|    x    |    y    |
|  -3.00 |  -1.27 |
|  -2.50 |  -3.74 |
|  -2.00 |  -3.64 |
|  -1.50 |  -2.24 |
|  -1.00 |  -0.84 |
|  -0.50 |  -0.12 |
|   0.00 |   0.00 |
|   0.50 |   0.12 |
```

1. Замените в решении задачи параметры цикла for так, чтобы не использовать переменную k в заголовке цикла. В качестве условия можно использовать следующее: *x*<=3.
2. Замените в решении задачи цикл for на цикл while.

Индивидуальные задания.

1. $y=x^2-5x-3$, $x\in[-3, 3]$;

2. $y=1-2\sin\pi x$, $x\in[-3, 3]$;

3. $y=\cos x+x^2$, $x\in[-\pi, \pi]$;

4. $y=\sin 2x-2x$, $x\in[-\pi, \pi]$;

5. $y=x\sin 2x+3$, $x\in[-\pi, \pi]$;

6. $y=x^2-4\sin\pi x$, $x\in[-3, 3]$;

7. $y=x^3-2\cos\pi x$, $x\in[-3, 3]$;

8. $y=x\cos\pi x-x^3$, $x\in[-3, 3]$;

9. $y=\frac{10x-5}{x^2}\sin\pi x$, $x\in[-3, 3]$;

10. $y=\frac{2-x}{x}-\sin\pi x$, $x\in[-3, 3]$;

11. $y=\frac{1}{x}-\cos\pi x$, $x\in[\mathrm{a}, \mathrm{b}]$;

12. $y=2+\frac{1}{x}\cos x$, $x\in[\mathrm{a}, b]$.

Задание 3.

Написать программу для вычисления произведения. Значение n вводится.

1. $a(a+1)...(a+n-1)$. Число a вводится.

2. $(1+\sin 0.1)(1+\sin 0.2)...(1+\sin 0.1n)$.

3. $a(a-n)(a-2n)...(a-n^2)$. Число a вводится.

4. $(1+\frac{1}{\sin 1})(1+\frac{1}{\sin 2})(1+\frac{1}{\sin 3})...(1+\frac{1}{\sin n})$

5. $(1+\frac{1}{1^2})(1+\frac{1}{2^2})(1+\frac{1}{3^2})...(1+\frac{1}{n^2})$;

6. $(1-\frac{1}{2^2})(1-\frac{1}{2^3})(1-\frac{1}{2^4})...(1-\frac{1}{2^{n+1}})$;

7. $(1-\frac{1}{2^2})(1+\frac{1}{3^3})(1-\frac{1}{4^4})...(1-\frac{(-1)^n}{n^n})$;

8. $\frac{(x-2)}{(x-1)}\cdot\frac{(x-4)}{(x-3)}\cdot...\cdot\frac{(x-(2n))}{(x-(2n-1))}$;

9. $(x+2x^2+x^3)(x^2+2x^3+x^4)(x^3+2x^4+x^5)...(x^n+2x^{n+1}+x^{n+2})$. Число x – вводится.

10. $(x+1)(x^2+x+1)(x^3+x^2+x+1)...(x^n+x^{n-1}+...+x+1)$. Число x – вводится.

11. $\dfrac{\cos 1}{\sin 1}\cdot\dfrac{\cos 1+\cos 2}{\sin 1+\sin 2}\cdot...\cdot\dfrac{\cos 1+...+\cos n}{\sin 1+...+\sin n}$.

12. $\underbrace{\sqrt{x+\sqrt{x^2+\sqrt{x^3+...\sqrt{x^n}}}}}_{n\ \text{корней}}$. Число x – вводится.

Задание 4.

Пример. Дано натуральное число n. Сформировать новое число, состоящее из нечетных цифр числа n.

Этапы выполнения задания

I. Исходные данные: n (число).

II. Результат: s (новое число)

III. Алгоритм решения задачи.

1. Ввод исходных данных — число n.
2. Определение начального значения нового числа: s = 0.
3. В переменной r будем хранить значение разрядной единицы для текущей четной цифры. Начальное значение r = 1.
4. В цикле, пока в числе есть цифры:

4.1 Найти остаток от деления текущего числа на 10 (z — текущая цифра числа).

4.2 Проверить, является ли полученная цифра нечетной.

4.3 Если цифра нечетная, то умножить ее значение разрядной единицы и прибавить

V. Программа:

```
n = int(input('n = '))
s, r = 0, 1

while n > 0:
    if (n % 10) % 2:
        s += n % 10 * r
        r *= 10
    n //= 10

if s:
    print('s = ', s)
else:
    print('Нет нечётных')
```

VI. Тестирование.

```
C:\CB_proejects\pr8_10\bin\Debug\pr8_10.exe
n = 123456789
s = 13579
```

```
C:\CB_proejects\pr8_10\bin\Debug\pr8_10.exe
n = 42786
s = 7
```

```
C:\CB_proejects\pr8_10\bin\Debug\pr8_10.exe
n = 9753
s = 9753
```

```
C:\CB_proejects\pr8_10\bin\Debug\pr8_10.exe
n = 242
net nechetnyh
```

к значению s, увеличить значение разрядной единицы в 10 раз.

4.4 Уменьшить текущее число в 10 раз.

5. Вывод значения переменной `s`. Если `s` = 0, то в числе нет нечетных цифр.

IV. Описание переменных: `n`, `s`, `r`, `z` – `int`.

<u>Индивидуальные задания.</u>

1. Определить первую цифру числа.
2. Посчитать сумму цифр числа.
3. Определить количество четных цифр числа.
4. Определить каких цифр в числе больше четных или нечетных.
5. Вывести на экран те цифры числа, которые больше введенного *x*.
6. Вывести на экран цифру, стоящую на средней позиции числа, если число имеет нечетное количество цифр, или 2 средние для числа с четным количеством цифр.
7. Поменять порядок цифр числа на обратный.
8. Переставить первую и последнюю цифры числа.
9. Приписать слева и справа к числу по 1.
10. Выбросить из записи числа n все цифры равные 1, оставив при этом прежним порядок остальных цифр. Например, из 5101234 →50234
11. Выбросить из записи числа все четные цифры.
12. После каждой цифры 1 в числе вставить еще одну 1. Например, из 51214 →5112114.

Задание 5.

Пример. Написать программу, которая будет генерировать случайные целые числа из промежутка [1; *x*] до тех пор, пока не будет сгенерировано число, кратное *k*. Вывести это число и количество сгенерированных чисел. Значения *k* и *x* вводятся (*k* < *x*).

Этапы выполнения задания

I. Исходные данные: числа `k` и `r`.

II. Результат: `r` (искомое число) и `n` (количество чисел).

III. Алгоритм решения задачи.

1. Ввод исходных значений.
2. Инициализация счетчика `n` = 0.
3. Поскольку нам известно условие окончания цикла — получить число, кратное `k`, то будем использовать цикл с постусловием. Условие можно записать так: `(r % k)`. Если значение этого выражения не равно нулю, то условие истинно.

3.1 Генерируем случайное число `r`. В Python для этого используется функция `random.randint(a, b)`, которая генерирует случайное число в промежутке [a; b].

3.2 Увеличим значение счетчика на 1.

3.3 Если число кратно k - выходим из цикла

V. Программа:

```
import random

k = int(input('k = '))
x = int(input('x = '))
n = 0

while True:
    r = random.randint(1, x)
    n += 1

    if r % k == 0:
        break

print('r =', r)
print('Попыток:', n)
```

VI. Тестирование.

```
k = 7
x = 100
r = 98
Попыток: 3
```

```
k = 34
x = 100
r = 34
Попыток: 25
```

```
k = 34
x = 1000
r = 816
Попыток: 38
```

```
k = 34
x = 1000
r = 306
Попыток: 8
```

4. Вывод результата.

IV. Описание переменных: `n, k, x — int`.

VII. Поскольку числа генерируются случайно, то для одних и тех же исходных данных могут получаться различные результаты (тесты 3—4).

Индивидуальные задания.

1. Написать программу, которая будет генерировать случайные числа из промежутка [-x; x]. Количество элементов вводится. Чему равно максимальное сгенерированное число?
2. Написать программу, которая будет генерировать случайные числа из промежутка [-x; x]. Количество элементов вводится. Чему равно минимальное сгенерированное число?
3. Написать программу, которая будет генерировать случайные числа из промежутка [1; x] до тех пор, пока не будет сгенерировано число, большее середины промежутка. Вывести это число и количество сгенерированных чисел.
4. Написать программу, которая будет генерировать случайные числа из промежутка [-x; x] до тех пор, пока не будет сгенерировано число, которое по модулю меньше x/2. Вывести это число и количество сгенерированных чисел.
5. Написать программу, которая будет генерировать случайные четные числа из промежутка [-x; x] до тех пор, пока не будет сгенерировано число 0. Каких чисел среди сгенерированных больше: положительных или отрицательных? (0 не учитывать).
6. Написать программу, которая будет генерировать случайные нечетные числа из промежутка [-x; x] до тех пор, пока не будет сгенерировано число -1. Каких чисел среди сгенерированных больше: кратных трем или кратных 5?
7. Написать программу, которая будет генерировать последовательность из возрастающих чисел. Первое число в последовательности и количество элементов вводятся. Чему равно максимальное сгенерированное число?

8. Написать программу, которая будет генерировать последовательность из убывающих чисел. Первое число в последовательности и количество элементов вводятся. Чему равно минимальное сгенерированное число?
9. Написать программу, которая будет генерировать последовательность из *n* чисел с чередующимися знаками из промежутка [-x; x]. Сколько среди них полных кубов?
10. Написать программу, которая будет генерировать последовательность из *n* чисел с чередующимися знаками из промежутка [-x; x]. Каких чисел среди сгенерированных больше четных или нечетных?
11. Написать программу, которая будет генерировать последовательность из чисел кратных 5 из промежутка [0; x] до тех пор пока не будет сгенерировано число, являющее полным квадратом. Каких чисел среди сгенерированных больше четных или нечетных?
12. Написать программу, которая будет генерировать последовательность из *n* шестизначных чисел кратных 5. Сколько среди сгенерированных чисел тех, у которых цифры расположены в неубывающем порядке?

Задание 6.

Пример. Резервуар наполнен `m` литрами водного раствора, содержащего `s` кг сахара. Каждую минуту забирают `x` литров раствора и добавляют `y` литров воды. Концентрация поддерживается равномерной посредством помешивания. Сколько сахара будет в растворе через `k` минут?

Этапы выполнения задания

I. Исходные данные: числа `m`, `s`, `x`, `k`.

II. Результат: `s` (новое значение)

V. Программа:

```
m = float(input('раствор (m): '))
s = float(input('сахар (s): '))
x = float(input('расход (x): '))
y = float(input('приход (y): '))
k = int(input('минут (k): '))

c = s / m
for i in range(k):
    m -= x
    if m <= 0:
        print('Сахара не осталось')
        break
    s = c * m

    m += y
    c = s / m
```

III. Алгоритм решения задачи.

1. Ввод исходных данных.
2. В переменной `c` будем хранить значение текущей концентрации раствора, в переменной `v` — количество воды.
3. В цикле, от 1 до `k`:

   3.1 Уменьшаем количество раствора на `x`.

   3.2 Добавляем воду.

   3.3 Пересчитываем концентрацию раствора.

   3.4 Если в какой-то момент весь раствор вылили, то прерываем цикл.
4. Вывод значения переменной `s`. Если количество раствора меньше либо равно нулю, то выводим сообщение «сахара не осталось», иначе выводим количество оставшегося сахара.

IV. Описание переменных: `n`, `s`, `r`, `z` – `int`.

```
else:
    print(f'Сахара осталось {s :.2f}')
```

VI. Тестирование.

```
раствор (m): 20
сахар (s): 4
расход (x): 4
приход (y): 5
минут (k): 10
Сахара осталось 0.65
```

```
раствор (m): 10
сахар (s): 1.3
расход (x): 2.4
приход (y): 1.8
минут (k): 50
Сахара не осталось
```

Индивидуальные задания

1. В 1626 г. индейцы продали остров Манхетен за 20 долларов. Если бы эти деньги были помещены в банк на текущий счет и ежегодный прирост составлял *x*%, какова была бы стоимость капитала в этом году?
2. Сумма в *R* рублей положена в банк. При этом прирост составляет *P*% ежегодно. Через какой промежуток времени сумма достигнет *M* рублей. ($M>R$).
3. Сумма в *P* у. е. положена в банк. При этом прирост составляет *m*% в год и считается непрерывным. Через какой срок сумма вклада увеличится в *x* раз.

4. Мяч упал с высоты H и, ударяясь о землю, отскакивает вновь каждый раз на 2/3 высоты, с которой он упал. Через сколько ударов мяч поднимется на высоту $P(P<H)$.
5. Население города ежегодно увеличивается на $1/k$ наличного состава жителей. Через сколько лет население города увеличится в x раз.
6. Численность населения области в 1990 году составляла Nтыс. человек. В течение p лет численность населения снижалась на x% ежегодно, а затем увеличивалась на y% ежегодно. Какова будет численность населения области в S году?
7. Жители островов Чунга и Чанга один раз в год по праздникам обмениваются драгоценностями. Жители острова Чунга привозят половину своих драгоценностей на остров Чанга, а жители острова Чанга привозят треть своих драгоценностей на остров Чунга. Какая часть драгоценностей будет находиться на острове Чунга через M лет.
8. Имеются два сосуда. В первом находится C_1 л воды, а во втором C_2 л воды. Из первого сосуда переливают половину воды во второй, а затем из второго переливают половину воды в первый (одно переливание) и так далее. Сколько воды окажется в каждом из сосудов после k переливаний.
9. Имеются два сосуда. В первом находится C_1 л воды, а во втором C_2 л воды. Из первого сосуда выливают m_1литров воды, а затем переливают половину оставшейся воды во второй сосуд. Затем из второго сосуда выливают m_2 литра воды и переливают половину остатка воды в первый (одно переливание). После чего действия повторяют. Сколько воды окажется в каждом из сосудов после k переливаний.
10. Резервуар наполнен m литрами водного раствора, содержащего 5 кг сахара. Приток воды в сосуд составляет x литров в минуту, а расход смеси из сосуда x-1 литров. Концентрация поддерживается равномерной посредством помешивания. Сколько сахара будет через k минут?
11. Через сосуд емкостью m л, наполненный 50% раствором соли, непрерывно протекает вода, причем в 1 минуту вливается x л воды и такое же количество

раствора выливается. Через какой промежуток времени раствор станет k-процентным.

12. Резервуар наполнен m литрами водного раствора, содержащего s кг сахара. Приток воды в сосуд составляет x литров в минуту, а расход смеси из сосуда y литров. Концентрация поддерживается равномерной посредством помешивания. Каждые h минут в резервуар засыпают f кг сахара ($f<s$). Какую концентрацию будет иметь раствор через k минут?