

**Демонстрационный вариант заданий практического этапа Конкурса
предпрофессиональных умений «Предпрофессиональная мастерская
инженерного и информационно-технологического профилей»
на площадке ФГБОУ ВО «НИУ «МЭИ»**

Номинация: «Инженерный класс»

Направление практического этапа: «Программирование»

Направление подготовки: «Прикладные технологии»

Задание

Артём решил написать свой поисковик. У него есть очень много данных, сгруппированных по ключам. Артём хочет понять, достаточно ли имеющейся информации, или нужно её дополнить. Для этого полезно получить случайную подвыборку из результатов поискового запроса. Артём нашёл такой алгоритм в интернете:

1. Рассматриваем массив результатов ответа на поисковый запрос (выдачи) размера N по одному элементу слева направо.

Для очередного элемента под номером i выберем случайное число p на отрезке от 0 до i включительно.

Поменяем местами i -ый и p -ый элементы.

2. Обработав таким образом массив, получим новый массив, отсортированный случайным образом.

Артём даже написал программу, но N настолько большое, что данные не помещаются в оперативную память.

Но Артём в любом случае не собирается просматривать все результаты, ему достаточно небольшой подвыборки, так что можно было бы выбрать только первые K элементов из массива. Тогда Артём придумал следующую модификацию, чтобы сделать это:

1. Выделить массив из K элементов.

2. Рассматривая очередное входное значение по алгоритму Артёма, если $p < K$, записать входное значение в p -ый элемент массива.

В конце будут получены первые K элементов массива результатов исходного алгоритма Артёма, а оперативная память не переполнится.

Реализуйте алгоритм Артёма с модификацией.

Для генерации случайных чисел используйте формулу $f(x) = x^m + b$, где $m = N$, $b = N - K$; m, b, N, K, x – беззнаковые 32-битные целые числа; N – размер поисковой выдачи; K – размер искомой выборки.

Пример решения

Выполнил: Иванов И. И.

Вариант N: демо.

Метод решения:

1. Ввести n, k .

2. Ввести первые k элементов в массив *random_shuffle*.

3. Для каждого элемента с индексом i [$k, n-1$]:

Ввести элемент.

Рассчитать его индекс после сортировки *shuffled_index* по формуле $(i^n + (n - k)) \% (i + 1)$.

Если *shuffled_index* $< k$, то заменить элемент с индексом *shuffled_index* в *random_shuffle*.

4) Вывести массив *random_shuffle*.

Таблица данных

Имя	Смысл	Тип данных	Структура данных
Входные			
n	количество результатов поисковой выдачи	целый	простая переменная
k	размер подвыборки, которую интересно получить	целый	простая переменная
next_element	очередной результат поисковой выдачи	строковый	простая переменная
Промежуточные			
i	счётчик цикла	целый	простая переменная
shuffled_index	случайный индекс в диапазоне от 0 до i	целый	простая переменная
Выходные			
random_shuffle	случайная подвыборка	строковый	одномерный массив

Тесты

Ввод	Вывод
-1	Некорректное количество результатов поисковой выдачи
2 -1	Некорректный размер выборки
2 3	Некорректный размер выборки
6 2 осьминог крокодил осел суслик паук кошка	Случайная подвыборка: паук, суслик

Код

```
n = int(input("Введите общее количество результатов поисковой выдачи: "))
if n < 0:
    print("Некорректное количество результатов поисковой выдачи")
    exit(1)

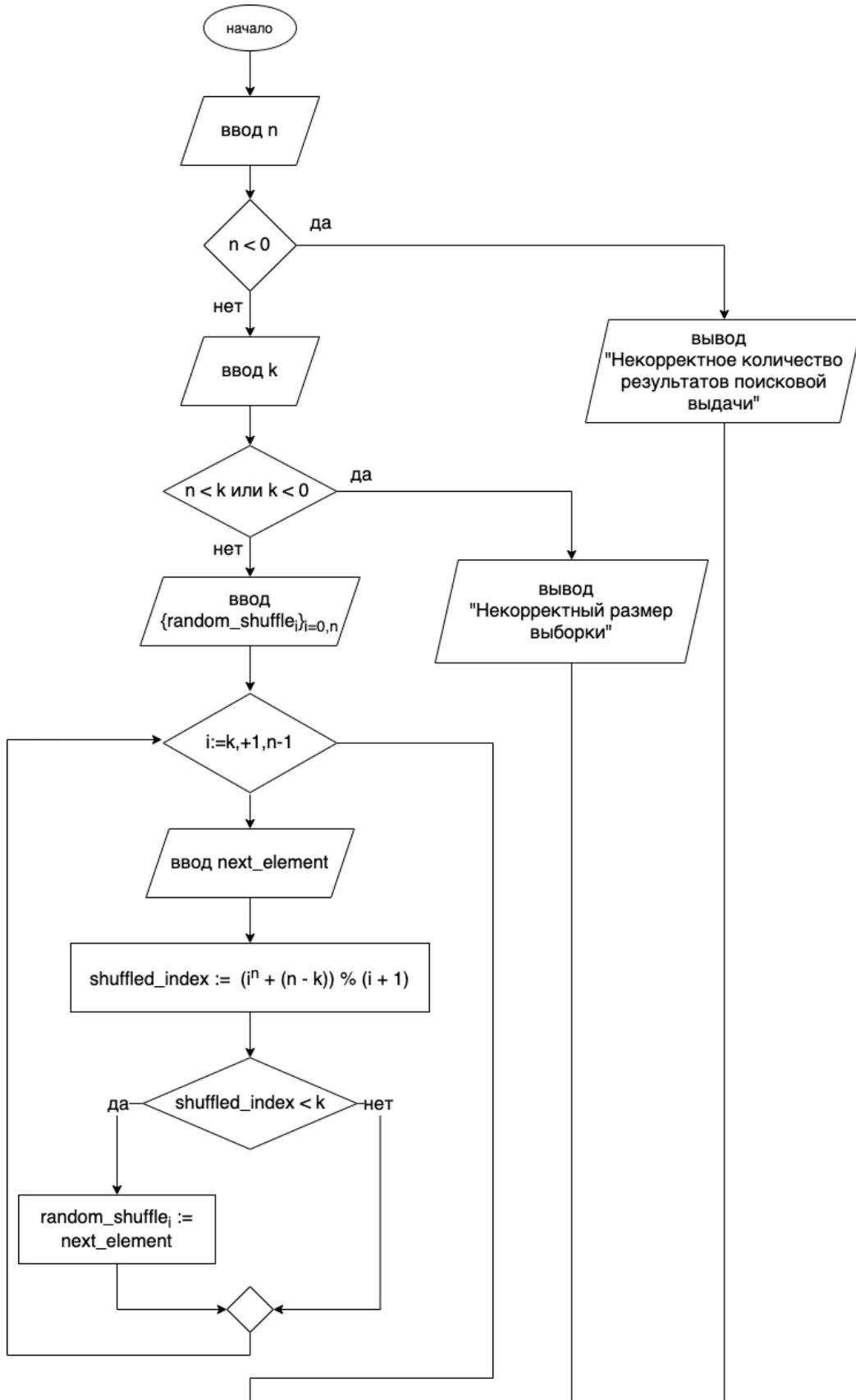
k = int(input("Введите размер подвыборки, которую вы хотите получить: "))
if k > n or k < 0:
    print("Некорректный размер выборки")
    exit(2)

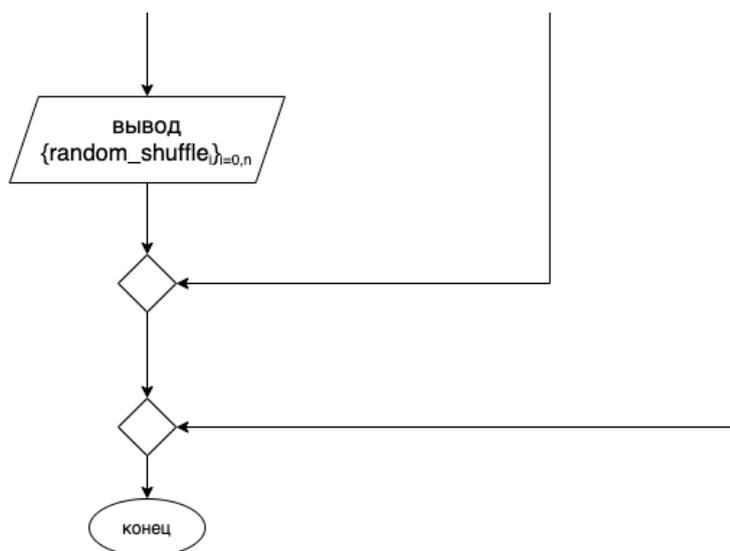
random_shuffle = []
for i in range(k):
    random_shuffle.append(input("Введите результат выдачи номер {}: ".format(i
+ 1)))

for i in range(k, n, 1):
    next_element = input("Введите результат выдачи номер {}: ".format(i + 1))
    shuffled_index = (pow(i, n) + (n - k)) % (i + 1)
    if shuffled_index < k:
        random_shuffle[shuffled_index] = next_element

print("Случайная подвыборка: {}".format(random_shuffle))
```

Блок-схема





Критерии оценки

№	Критерии	Максимальные баллы
1.	Корректность разработанной программы	8 баллов
2.	Эффективность разработанной программы	8 баллов
3.	Надёжность разработанной программы	8 баллов
4.	Качество документации к программе	7 баллов
5.	Полнота тестирования программы	7 баллов
6.	Читабельность программного кода	6 баллов
7.	Самостоятельность выполнения работы	4 балла
8.	Умение аргументировать принятые решения	4 балла
9.	Умение отвечать на вопросы	4 балла
10.	Полнота ответов на дополнительные вопросы	4 балла
Максимально возможное количество баллов:		60 баллов

**Тематический рубрикатор
практического этапа Конкурса предпрофессиональных умений
«Предпрофессиональная мастерская инженерного и информационно-
технологического профилей». Направление «Программирование»,
профиль «Прикладные технологии»**

1. Информация и информационные процессы.
2. Математические основы информатики.
3. Системы счисления.
4. Элементы комбинаторики, теории множеств и математической логики. 5. Дискретные объекты.
6. Типы и структуры данных. Одномерные и многомерные массивы (двумерные массивы — матрицы).
7. Алгоритмические конструкции. Основы алгоритмизации (базовые управляющие структуры). Нисходящий («сверху вниз», top-down) способ проектирования алгоритмов (декомпозиция задач на подзадачи). Кодирование базовых алгоритмических конструкций на одном из языков программирования.
8. Составление алгоритмов и их программная реализация. Алгоритмы решения типовых задач базового уровня из различных предметных областей, алгоритмы обработки одномерных массивов и матриц (поиск экстремума (наибольшего или наименьшего элемента) в одномерном/двумерном массиве, сортировка одномерных массивов и др.).
9. Анализ алгоритмов.
10. Основы программирования. Разработка программ на языках высокого уровня. Парадигма структурного программирования. Интегрированные среды разработки программ. Приёмы отладки программ.
11. Математическое моделирование.
12. Аппаратное и программное обеспечение компьютера.
13. Подготовка и выполнение исследовательского проекта.