

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Донской государственный технический университет»

ОЛИМПИАДЫ «Я – БАКАЛАВР»
ДЛЯ ОБУЧАЮЩИХСЯ 9 КЛАССОВ
2025/2026 учебный год

ЗАКЛЮЧИТЕЛЬНЫЙ ЭТАП

ПО ИНФОРМАТИКЕ

КЛАСС 11

ВАРИАНТ 1

Задание 1 (10 баллов)

Составьте графический алгоритм – блок-схему для представленного стихотворения Генриха Гейна.

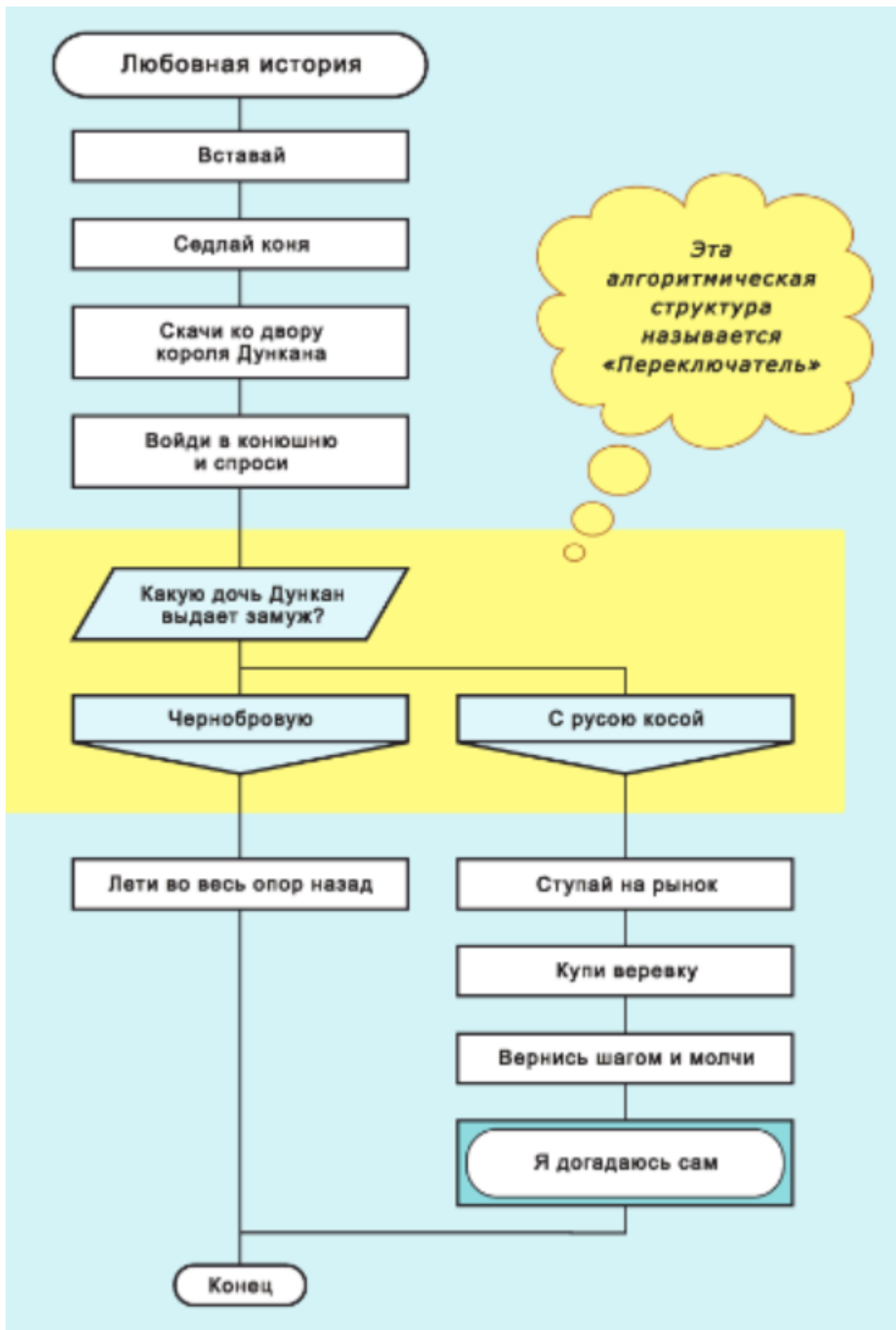
Вставай, слуга! Коня седлай!
Через рощи и поля
Скачи скорее ко двору
Дункана-Короля.

Зайди в конюшню там и жди.
И, если кто войдёт,
Спроси: «Которую Дункан
Дочь замуж выдаёт?»

Коль чернобровую – лети
Во весь опор назад.
Коль ту, что с русою косой,
Спешить не надо, брат.

Тогда ступай на рынок ты
Купи верёвку там,
Вернися шагом и молчи –
Я догадаюсь сам.

Решение:



Задание 2 (10 баллов)

Выполните **по действиям** расчёт значения логического выражения. Результат выполнения каждого действия представьте в бланк ответа в виде решения каждого действия «**столбиком**».

NOT (NOT 16 IMP 51 OR 1E) IMP 10

Операциям исключающее ИЛИ, эквивалентность и импликация соответствуют операторы XOR, EQV и IMP. Для обозначения целочисленных операндов, представленных в восьмеричной и шестнадцатеричной системах счисления, используются префиксы &O и &H соответственно. Если перед числовым значением нет префиксов восьмеричной и шестнадцатеричной систем, значит это число в выражении представлено в десятичной системе счисления. Каждый операнд в расчётах представлять в виде шестнадцатеричной последовательности, результат расчёта каждого действия, естественно, также должен быть шестнадцатеричной последовательностью. Все расчёты и переводы из разных систем счисления представить в бланке ответа. Окончательный результат решения выражения представить в трёх системах счисления: двоичной, восьмеричной, шестнадцатеричной.

Решение:

NOT (NOT 16 IMP 51 OR 1E) IMP 10

Переводим все операнды в двоичную систему счисления:

$$16_{(8)} = 001110_{(2)}$$

$$51_{(8)} = 101001_{(2)}$$

$$1E_{(16)} = 00011110_{(2)}$$

$$10_{(16)} = 00010000_{(2)}$$

Определяем результат выполнения каждой операции побитно, используя для представления операндов шестнадцатиразрядный дополнительный код:

1) (NOT&O16)

$$\begin{array}{r} 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 0 \\ \hline 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 1 \end{array}$$

2) &O51 OR &H1E

$$\begin{array}{r} 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1 \\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 0 \\ \hline 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1 \end{array}$$

3) (NOT&O16) IMP &O51 OR &H1E

$$\begin{array}{r} 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 1 \\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1 \\ \hline 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1 \end{array}$$

4) NOT ((NOT&O16) IMP &O51 OR &H1E)

$$\begin{array}{r} 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1 \\ \hline 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0 \end{array}$$

5) NOT ((NOT&O16) IMP &O51 OR &H1E) IMP &H10

```
1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
-----
0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
```

Ответ: в 2сс 111111, в 8сс 77, в 16сс 3F

Задание 3 (10 баллов)

Расшифруйте четверостишие Омара Хайяма:

РЛЗЬ ЁМЭЙЗ АВБЖУ ИЙЗАВЛУ, БЖЦЛУ ЖЦЭЗЬЖЗ ЖЮЁЩЕЗ,

ЭЫЩ ЫЩАЖФО ИЙЩЫВЕЩ БЦИЗЁЖВ ЭЕШ ЖЦРЩЕЩ:

ЛФ ЕМРСЮ ЪЗЕЗЭЦГ, РЮЁ РЛЗ ИЗИЩЕЗ ЮКЛУ

В ЕМРСЮ ЪМЭУ ЗЭВЖ, РЮЁ ЫЁЮКЛЮ К ДЮЁ ИЗИЩЕЗ.

Конкретный вариант шифрования методом Цезаря при этом неизвестен.

Ответом на это задание будет являться расшифрованное четверостишие и размер ключа для его расшифровки

Решение:

Чтоб мудро жизнь прожить, знать надобно немало,

Два важных принципа запомни для начала:

Ты лучше голодай, чем что попало есть,

И лучше будь один, чем вместе с кем попало.

Задание 4 (20 баллов)

Компании требуется организовать собственную информационную систему, одним из элементов которой является телефонный справочник сотрудников с необычным запросом.

Ваша задача — реализовать телефонный справочник с поддержкой двух команд:

1) ADD name phone — добавить или обновить телефон для заданного имени.

2) QUERY name prefix — вывести все записи, у которых первые символы имени совпадают с введённым prefix.

Входные данные.

Первая строка: количество команд N ($1 \leq N \leq 105$).

Далее N строк — команды двух видов:

1) ADD name phone — имя и номер (имя — латинские строчные буквы, номер — цифры).

2) QUERY name prefix — имя (строка).

Выходные данные.

Для каждой команды QUERY вывести записи (имя и номер), у которых первые символы имени совпадают с введённым prefix. Если таких записей нет, вывести «Номеров нет».

Пример.

Введите количество команд (N): 5
Введите команду 1: ADD ivan 12345
Добавлено/обновлено: ivan -> 12345
Введите команду 2: ADD igor 67890
Добавлено/обновлено: igor -> 67890
Введите команду 3: QUERY x i
Результаты для префикса 'i':
igor 67890
ivan 12345
Введите команду 4: QUERY x iv
Результаты для префикса 'iv':
ivan 12345
Введите команду 5: QUERY x f
Номеров нет

Решение:

Один из вариантов исполнения

Вариант решения задачи на Python.

```
def validate_name(name):
    # Имя должно содержать только строчные латинские буквы
    return name.isalpha() and name.islower()

def validate_phone(phone):
    # Телефон должен содержать только цифры
    return phone.isdigit()

def validate_prefix(prefix):
    # Префикс - строка из строчных латинских букв
    return prefix.isalpha() and prefix.islower()

def main():
    try:
        n = int(input("Введите количество команд (N): "))
        if not (1 <= n <= 10**5):
            print("Ошибка: N должно быть от 1 до 10^5")
            return
    except ValueError:
        print("Ошибка: N должно быть целым числом")
        return

    phonebook = {}

    for i in range(n):
        line = input(f"Введите команду {i+1}: ").strip().split()
        if not line:
```

```

        print("Ошибка: пустая команда")
        continue

    cmd = line[0]
    if cmd == "ADD":
        if len(line) != 3:
            print("Ошибка: команда ADD должна содержать 3 аргумента (ADD name
phone)")
            continue

        name, phone = line[1], line[2]
        if not validate_name(name):
            print(f"Ошибка: имя '{name}' должно содержать только строчные
латинские буквы")
            continue
        if not validate_phone(phone):
            print(f"Ошибка: номер '{phone}' должен содержать только цифры")
            continue

        phonebook[name] = phone
        print(f"Добавлено/обновлено: {name} -> {phone}")

    elif cmd == "QUERY":
        if len(line) != 3:
            print("Ошибка: команда QUERY должна содержать 3 аргумента (QUERY
name prefix)")
            continue

        prefix = line[2]
        if not validate_prefix(prefix):
            print(f"Ошибка: префикс '{prefix}' должен содержать только
строчные латинские буквы")
            continue

        matched = [(name, phone) for name, phone in phonebook.items() if
name.startswith(prefix)]
        if not matched:
            print("Номеров нет")
        else:
            matched.sort(key=lambda x: x[0])
            print(f"Результаты для префикса '{prefix}':")
            for name, phone in matched:
                print(name, phone)

    else:
        print(f"Ошибка: неизвестная команда '{cmd}'")

if __name__ == "__main__":
    main()

```

Задание 5 (25 баллов)

Вам дан текст, состоящий из русских букв, пробелов и знаков препинания (точка, запятая, восклицательный знак, вопросительный знак).

Ваша задача – создать программу для помощи лингвисту. Программа должна производить по выбору пользователя шифрование или расшифровку сообщений, следуя определенным правилам редактирования.

Правила редактирования при шифровании:

1. Удалить все знаки препинания. Не заменять их пробелами, а именно удалить.

2. Заменить все последовательности из одного или более пробелов на ровно один пробел. Если пробелы были в начале или конце текста, их тоже нужно удалить.

3. Удалить все слова, длина которых меньше заданного числа K . Словом считается последовательность непробельных символов, ограниченная пробелами или началом/концом текста.

4. Для каждого оставшегося слова сдвинуть буквы в слове на одну позицию вправо (циклически). То есть, последняя буква слова становится первой, предпоследняя – второй и так далее. Например, слово "привет" после сдвига станет "тприве".

В программе предусмотреть проверку входных данных. Программа должна, корректно работать при допустимых входных данных или выводить предупреждение о необходимости их корректировки.

Входные данные при шифровании

Первая строка содержит целое число K ($1 \leq K \leq 10$) — минимальную длину слова.

Вторая строка содержит текст S ($1 \leq \text{длина } S \leq 100$ символов), состоящий из русских букв, пробелов и знаков препинания (.,!,?).

Выходные данные

Выведите расшифрованный текст, полученный после применения указанных операций в заданном порядке.

При расшифровке необходимо выполнить обратные действия. При этом удалённая при шифровании информация (знаки препинания и слова) не восстанавливается. Производится только сдвиг в обратном порядке.

Входные данные

Зашифрованная строка.

Выходные данные

Расшифрованная строка.

Пример шифрования

Входные данные

Выберите режим (encrypt/decrypt): encrypt

Введите минимальную длину слова (K): 4

Введите текст для шифрования: Привет, мир! Как дела? Все хорошо...

Выходные данные

тПриве адел охорош

Пример расшифровка

Выберите режим (encrypt/decrypt): decrypt

Введите текст для расшифровки: тПриве адел охорош
Результат: Привет дела хорошо

Решение:

Вариант решения задачи для Python.

```
def encrypt_decrypt(text, k, mode="encrypt"):

    if mode == "encrypt":
        # 1. Удаление знаков препинания
        punctuation = ['.', ',', '!', '?']
        text = ''.join(c for c in text if c not in punctuation)

        # 2. Замена последовательностей пробелов на один пробел
        text = ' '.join(text.split())

        # 3. Удаление коротких слов
        words = text.split()
        filtered_words = [word for word in words if len(word) >= k]

        # 4. Сдвиг букв в словах
        shifted_words = [word[-1] + word[:-1] for word in filtered_words]

        return ' '.join(shifted_words)

    elif mode == "decrypt":
        words = text.split()
        unshifted_words = [word[1:] + word[0] for word in words] # Обратный
сдвиг
        return ' '.join(unshifted_words)

    else:
        return "Ошибка: Неверный режим. Используйте 'encrypt' или 'decrypt'."

# Выбор режима
mode = input("Выберите режим (encrypt/decrypt): ").lower()
if mode not in ["encrypt", "decrypt"]:
    print("Ошибка: Некорректный режим. Используйте 'encrypt' или 'decrypt'.")
    exit()

# Ввод данных в зависимости от выбранного режима
if mode == "encrypt":
    try:
        k = int(input("Введите минимальную длину слова (K): "))
        if k < 1:
            raise ValueError("Минимальная длина слова должна быть не меньше 1.")
    except ValueError:
        print("Ошибка: Некорректный ввод для K. Пожалуйста, введите целое число
        больше 0.")
        exit()
    text = input("Введите текст для шифрования: ")
else: # mode == "decrypt"
    k = 0 #k не используется при расшифровке
    text = input("Введите текст для расшифровки: ")

# Выполнение операции и вывод результата
result = encrypt_decrypt(text, k, mode=mode)
```

```
print(f"Результат: {result}")
```

Задание 6 (25 баллов).

Создан новый процессор, который имеет ограничение. Он может обрабатывать числа только в системах счисления с основанием от 2 до 36 включительно. Цифры в системах счисления больше 10 представлены латинскими буквами: A = 10, B = 11, C = 12 и так далее до Z = 35.

Ваша задача – написать программу, которая принимает на вход последовательность чисел, каждое из которых представлено в своей системе счисления, и преобразует их все в десятичную систему счисления. Затем, программа должна определить и вывести минимальное, максимальное, а также среднее значение в полученной десятичной последовательности.

Входные данные

1) N – целое число, количество чисел в последовательности ($1 \leq N \leq 100$).
2) Последовательность из N строк: Каждая строка содержит информацию об одном числе в следующей форме:

- S это строка длиной до 20 символов. Представляет число в некоторой системе счисления. Число может состоять из цифр ('0'-'9') и/или латинских букв ('A'-'Z');
- B это целое число в диапазоне от 2 до 36 включительно. Представляет основание системы счисления, в которой записано число S;
- S и B разделены одним пробелом.

В программе предусмотреть проверку входных данных. Программа должна, корректно работать при допустимых входных данных или выводить предупреждение о необходимости их корректировки.

Выходные данные

Вывести три числа через пробел: минимальное, максимальное и среднее значение, полученные после преобразования всех N чисел в десятичную систему счисления.

Пример

Ввод

Введите количество чисел (1-100): 3

Введите число и основание через пробел (число 1): 10 2

Введите число и основание через пробел (число 2): 10 10

Введите число и основание через пробел (число 3): F 16

Выходные данные

Результаты работы программы:

Минимальное значение: 2

Максимальное значение: 15

Среднее значение: 9.0

Решение:

Один из вариантов исполнения

Вариант решения задачи для Python.

```
def is_valid_num_in_base(s, base):
    allowed_chars = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"[:base]
    s = s.upper()
    return all(c in allowed_chars for c in s)

# Читаем количество чисел с проверкой
while True:
    try:
        N = int(input("Введите количество чисел (1-100): "))
        if 1 <= N <= 100:
            break
        else:
            print("Ошибка: N должно быть от 1 до 100.")
    except ValueError:
        print("Ошибка: введите целое число.")

values = []

for i in range(N):
    while True:
        line = input(f"Введите число и основание системы счисления через пробел
(число {i+1}): ")
        parts = line.split()
        if len(parts) != 2:
            print("Ошибка: введите два значения через пробел.")
            continue
        S, B_str = parts
        try:
            B = int(B_str)
            if not (2 <= B <= 36):
                print("Ошибка: основание системы счисления должно быть от 2 до
36.")
                continue
        except ValueError:
            print("Ошибка: основание системы счисления должно быть целым
числом.")
            continue

        if not is_valid_num_in_base(S, B):
            print(f"Ошибка: число '{S}' содержит недопустимые символы для
основания {B}.")
            continue

        try:
            val = int(S, B)
            values.append(val)
            break
        except ValueError:
            print(f"Ошибка: не удалось преобразовать число '{S}' из системы с
основанием {B}.")
            continue

minimum = min(values)
```

```
maximum = max(values)
average = sum(values) / N
```

```
# Поясняющий вывод результатов
```

```
print("\nРезультаты работы программы:")
print(f"Минимальное значение: {minimum}")
print(f"Максимальное значение: {maximum}")
print(f"Среднее значение: {average}")
```