

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

ЗАКЛЮЧИТЕЛЬНЫЙ ЭТАП

ПО ИНФОРМАТИКЕ

КЛАСС 11

ВАРИАНТ 2

Задание 1 (10 баллов)

Составьте графический алгоритм – блок-схему для описания жизненной ситуации космических пришельцев. Пришельцу Кулеру надо провести проверку летающей тарелки.

Для полёта летающей тарелки на другую планету необходимо проверить пришельцу Кулеру левый и правый двигатели, в случае их исправности сразу необходимо отправиться в полёт запустив плазменный реактор.

Но, если вдруг левый двигатель не сработал, то надо выполнить проверку на работоспособность правого двигателя. Если правый двигатель не запускается, то через 3 секунды динамика запуска должна повторяться бесконечно.

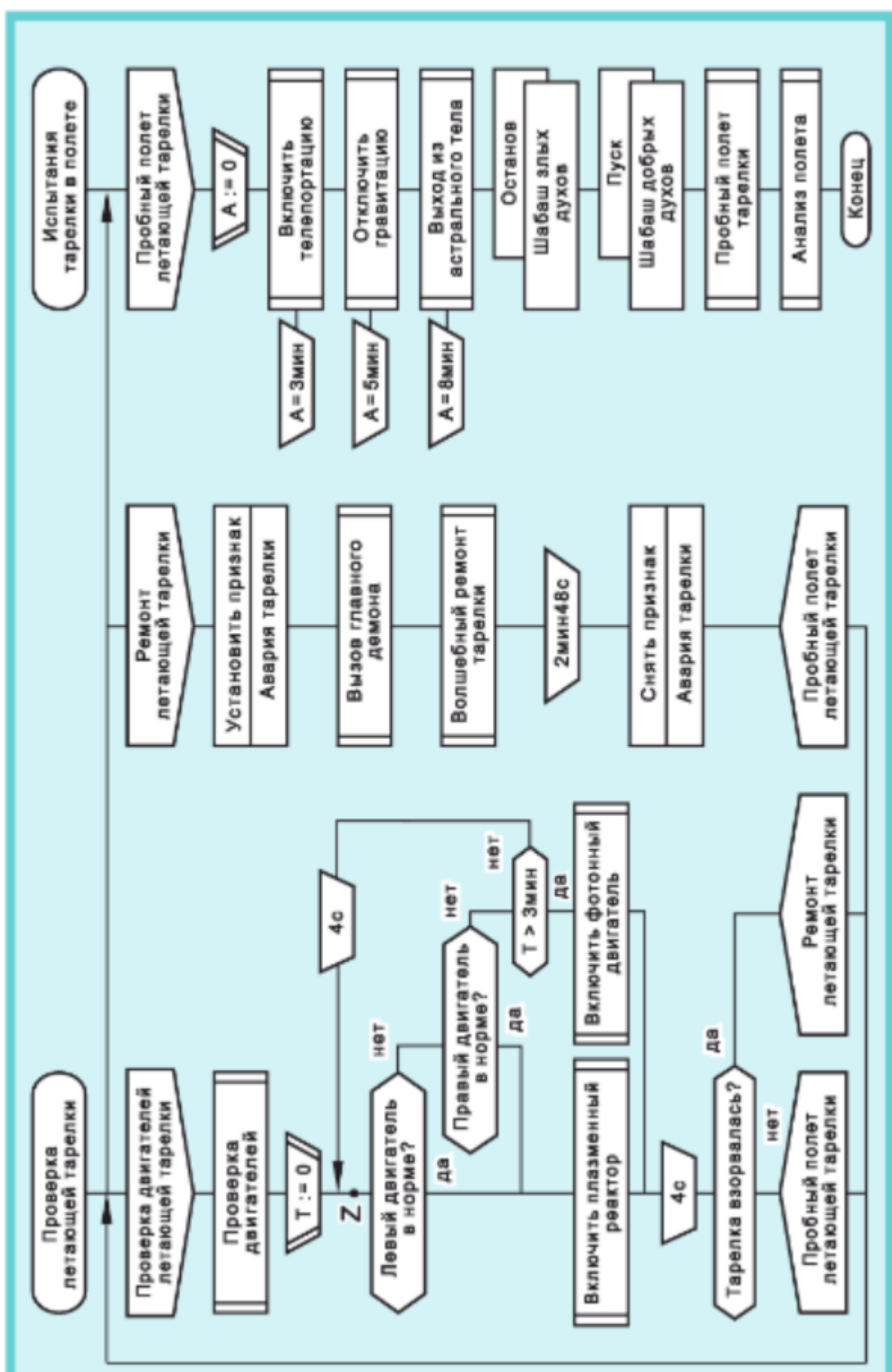
Если летающая тарелка не взорвалась, то пробный полёт выполнен, иначе надо выполнить ремонт.

Для ремонта летающей тарелки надо установить признак «Авария тарелки» и после этого вызвать главного демона – он выполняет волшебный ремонт летающей тарелки, который вызывает паузу в размере 2 минуты 48 секунд. После окончания ремонта демон должен убрать с летающей тарелки признак «Авария тарелки».

После ремонта демон должен осуществить пробный полёт летающей тарелки для испытания её неисправности. После полёта надо включить телепортацию, отключить гравитацию, выйти из астрального тела, остановить шабаш злых духов, запустить шабаш добрых духов.

Завершить стыковку на космическом порту планеты Юпитер и завершить космические сутки анализом полёта.

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 (012 OR (NOT 07 XOR H24)) AND 31

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 (012 OR (NOT 07 XOR H24)) AND 31

Переводим все операнды в двоичную систему счисления:

$$12_{(8)} = 1010_{(2)}$$

$$7_{(8)} = 111_{(2)}$$

$$31_{(10)} = 11111_{(2)}$$

$$24_{(16)} = 00100100_{(2)}$$

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

1) NOT 07

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

2) NOT 07 XOR H24

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

3) 012 OR (NOT 07 XOR H24)

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

1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0

4) NOT NOT (&O12 OR (NOT &O7 XOR &H24))

1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0

5) NOT NOT (&O12 OR (NOT &O7 XOR &H24)) AND 31

1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0

0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1

0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0

Ответ: в 2сс 11110, в 8сс 36, в 16сс 1E

Задание 3м(10 баллов)

Буратино обнаружил начальную строку зашифрованного послания, написанного неизвестным ему шифром:

ШЫР-ПИР Ю ПАПУЖГЫ

ЗЭЛЭМЪЫЙ ГЁСРЫГ...

Помогите Буратино расшифровать послание.

Решение:

ЖИЛ-БЫЛУБАБУШКИ СЕРЕНЬКИЙ КОЗЛИК

Задание 4 (20 баллов)

Компания желает создать информационную систему для учёта книг в библиотеке. Система должна поддерживать команды для добавления и поиска записей.

Ваша задача – написать программу, которая содержит каталог книг с возможностью:

1. Добавлять новую книгу или обновлять данные по уже существующей с помощью команды ADD title author, в которой title — название книги (строка из букв русского алфавита и нижнего подчёркивания. Пробелов быть не должно), author — автор книги (строка из букв русского алфавита и нижнего подчёркивания. Пробелов быть не должно).

Если книга с таким названием уже есть, обновить автора.

2. Производить поиск книг по префиксу названия с помощью команды QUERY prefix:

— вывести все книги, названия которых начинаются с prefix, в формате title author;

— если книг с таким префиксом нет, вывести «Книг нет».

Входные данные:

Первая строка — это количество команд N (целое число, $1 \leq N \leq 100$).

Далее N строк, каждая с командой ADD или QUERY согласно описанию выше.

Выходные данные:

Для каждой команды QUERY вывести все подходящие записи (одна на строку), либо «Книг нет», если совпадений не найдено.

Пример.

Введите количество команд (от 1 до 100): 7

Введите команду 1:

ADD Война_и_мир Лев_Толстой

Введите команду 2:

ADD Преступление_и_наказание Фёдор_Достоевский

Введите команду 3:

QUERY Во: Война_и_мир Лев_Толстой

Введите команду 4:

QUERY Лев: Книг нет

Решение:

Вариант решения задачи для Python (структура исполнения сохраняется, словесные формулировки должны быть заменены на описываемую предметную область)

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

Ваша задача – создать программу для помощи шеф-повару, который использует хитрую систему шифрования рецептов только на русском языке. Программа должна производить по выбору пользователя шифрование или расшифровку рецептов, 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 ингредиенты (слова), длина которых меньше заданного числа K . Ингредиентом считается последовательность непробельных символов, ограниченная пробелами или началом/концом текста.

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

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 ингредиента.

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

Выходные данные

Выведите зашифрованный рецепт, полученный после применения указанных операций в заданном порядке.

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

Входные данные

Зашифрованный рецепт.

Выходные данные

Расшифрованный рецепт.

Пример шифрования

Входные данные

Выберите режим (encrypt/decrypt): encrypt

Введите минимальную длину ингредиента K : 5

Введите рецепт: Соль, укроп, кабачки, лук, морковь!

Выходные данные

Зашифрованный рецепт: пукро икабачк ьморков

Пример расшифровки

Введите зашифрованный рецепт: пукро икабачк ьморков

Расшифрованный рецепт: укроп кабачки морковь

Решение:

Вариант решения задачи для Python (структура исполнения сохраняется, словесные формулировки должны быть заменены на описываемую предметную область)

Один из вариантов исполнения

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)

Вы – криптоархеолог, занимающийся расшифровкой древних посланий, найденных на артефактах давно исчезнувших цивилизаций. Каждая цивилизация использовала свою уникальную систему счисления для записи цифровых значений – от примитивных двоичных кодов до сложных систем с основанием 36, где символы '0'-'9' и 'A'-'Z' (A=10, B=11, ..., Z=35) представляли отдельные разряды.

Ваша задача – разработать программу которая должна принимать на вход последовательность шифров (цифр), представленных в их родных системах счисления, преобразовывать их все в единую, универсальную десятичную систему, а затем определять и выводить два минимальных и два максимальных значения в порядке их возрастания.

Входные данные

- 1) M – целое число, количество шифров для анализа ($1 \leq M \leq 10$).
- 2) Последовательность из M строк. Каждая строка описывает один шифр в следующем формате:

- S это строка длиной до 20 символов. Представляет зашифрованное сообщение в некоторой системе счисления. Состоит 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 записан шифр S (то есть, "алфавит" цивилизации, создавшей этот шифр);
- S и B разделены одним пробелом.

Не забудьте о безопасности! Программа должна тщательно проверять входные данные на корректность и выдавать информативные сообщения об ошибках в случае некорректного ввода.

Выходные данные

Вывести четыре числа через пробел: два минимальных, два максимальных, рассчитанные после преобразования всех M чисел в десятичную систему счисления.

Пример

Ввод

Введите количество шифров (1-10): 5

Введите шифр и основание через пробел (шифр 1): 100 2

Введите шифр и основание через пробел (шифр 2): AF 16

Введите шифр и основание через пробел (шифр 3): 120 10

Введите шифр и основание через пробел (шифр 4): 1001 2

Введите шифр и основание через пробел (шифр 5): 2F 16

Выходные данные

4 9 120 175

Решение:

Вариант решения задачи для Python (структура исполнения сохраняется, словесные формулировки должны быть заменены на описываемую предметную область)

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}")
```