

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

ЗАКЛЮЧИТЕЛЬНЫЙ ЭТАП

ИНФОРМАТИКА

КЛАСС 10

ВАРИАНТ 2

Задание 1 (10 баллов)

Перед сотрудником торговой компании стоит задача составить графическую комбинацию из черных линий разной ширины, пробелов и цифр, в которой будет закодирована информация о производителе и товаре в общей базе товаров организации. Для этого ему предоставили шаблоны, указанные на Рисунке 1.

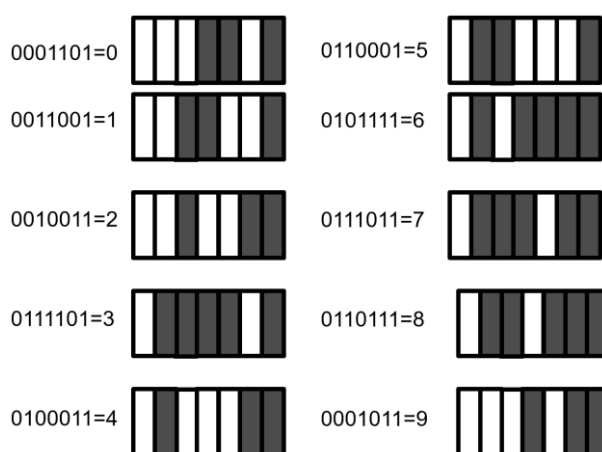
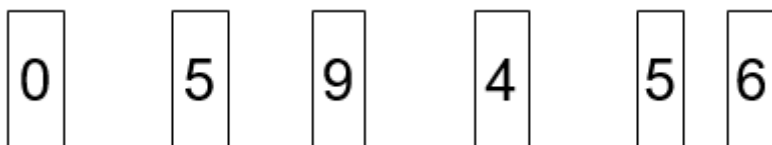


Рисунок 1 – Кодирование чисел с помощью штрих-кода

Изобразите штрих-код для продукции с кодом



И обязательно добавьте первые три бита в нём и последние —101.

Они нужны только для того, чтобы сканер смог определить ширину полосы, соответствующей одному биту (ведь размеры штрих-кода на разных

упаковках могут быть разными) и настроиться на распознавание. Размеры первых и последних трёх кодов должны быть в полтора раза длиннее, чем код самого товара

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**».

36 AND NOT NOT (&H2A XOR 31 XOR &H1E)

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:

36 AND NOT NOT (&H2A XOR 31 XOR &H1E)

Переводим все операнды в двоичную систему счисления:

$$36_{(10)} = 100100_{(2)}$$

$$2A_{(16)} = 00101010_{(2)}$$

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

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

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

1) (&H2A XOR 31)

0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
<hr/>																
0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1

2) &H2A XOR 31 XOR &H1E

0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
<hr/>																
0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1

3-4) NOT NOT &H2A XOR 31 XOR &H1E

0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

5) 35 AND NOT NOT &H2A XOR 31 XOR &H1E

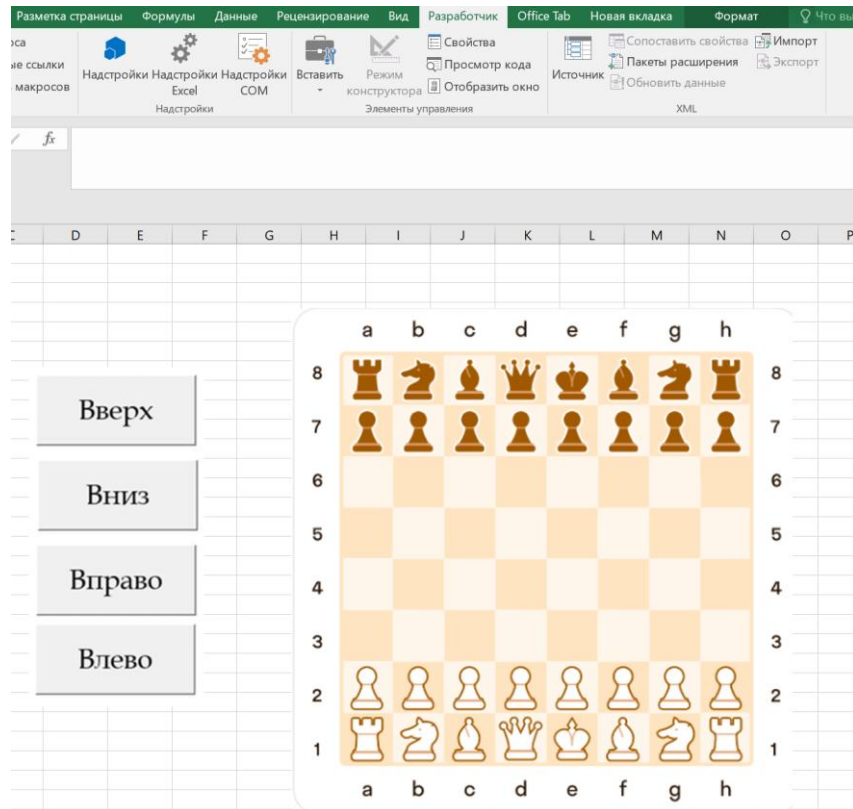
0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1
<hr/>																
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Ответ: в 2сс 100000, в 8сс 40, в 16сс 20

Задание 3 (10 баллов)

Перед Вами игра в шахматы. В роли Фигуры для игры может выступать любая фигура, выбранная пользователем. Создайте 4 макроса на 4 действия Фигуры (перемещение «Вверх», «Вниз», «Вправо», «Влево», представим, что именно такие действия может выполнять рассматриваемая фигура). Герой должен нажатием пользователя на кнопку «Вверх» перемещаться по лабиринту вверх; нажатием на кнопку «Вниз» перемещаться по лабиринту вниз и т.д. и так до тех пор, пока не достигнет Финиша. Картинка лабиринта предоставляется.

Для выполнения поставленного задания с помощью макрокоманд табличного редактора MS Excel составьте 4 макроса на перемещение Героя в 4 направления Лабиринта. Полный текст программы на перемещение Героя от Страта до Финиша не требуется. Ответ на поставленное задание представить в виде записи в бланк ответа 4 программ на 4 действия Героя («Вверх», «Вниз», «Вправо», «Влево»). Программный код каждого макроса надо переписать в бланк ответа.



Решение:

Sub BВверх()

,

' BВверх Макрос

,

Selection.ShapeRange.IncrementLeft 2.5

Selection.ShapeRange.IncrementTop -29.5

End Sub

Sub Влево()

,

' Влево Макрос

,

Selection.ShapeRange.IncrementLeft -50

Selection.ShapeRange.IncrementTop -2

End Sub

Sub Вправо()

,

' Вправо Макрос

,

Selection.ShapeRange.IncrementLeft 83.5

```
Selection.ShapeRange.IncrementTop 1.5
End Sub
```

```
Sub Вниз()
```

```
,
```

```
' Вниз Макрос
```

```
,
```

```
,
```

```
Selection.ShapeRange.IncrementLeft -5
```

```
Selection.ShapeRange.IncrementTop 76
```

```
End Sub
```

Задание 4 (20 баллов)

Дана квадратная сетка размером $N \times N$ (где $3 \leq N \leq 50$), представляющая пиксели черно-белого изображения или элементы узора. Каждый элемент сетки — целое число, обозначающее значение пикселя или характеристики узора.

Ваша задача — составить программу которая определит, сколько элементов сетки обладают уникальным свойством симметрии для элемента на позиции (i, j) значения в четырёх зеркальных позициях относительно центра сетки одинаковы. Конкретно, значение элемента должно совпадать со значениями элементов на позициях $(N-1-i, j)$, $(i, N-1-j)$ и $(N-1-i, N-1-j)$.

Иными словами, элемент считается симметричным, если выполняется равенство:

$$A[i][j] = A[N-1-i][j] = A[i][N-1-j] = A[N-1-i][N-1-j]$$

Необходимо подсчитать и вывести количество таких симметричных элементов в сетке.

Входные данные

Первой строкой вводится целое число N — размер сетки. Далее следуют N строк по N целых чисел, описывающих значения элементов сетки.

Выходные данные

Одно целое число — количество симметричных элементов по описанному критерию.

Пояснение к задаче

Пример зеркальных элементов на матрице 5x5. Пусть есть матрица:

7	3	2	3	7
2	8	1	8	2
5	4	9	4	4
7	8	3	7	2
7	3	4	5	7

7	3	2	3	7
2	8	1	8	2
5	4	9	4	4
7	8	3	7	2
7	3	4	5	7

7	3	2	3	7
2	8	1	8	2
5	4	9	4	4
7	8	3	7	2
7	3	4	5	7

В этой матрице показаны различные симметричные элементы.

Если матрица имеет размер $N \times N$, то индексы столбцов идут от 0 до $N-1$.

Зеркальными относительно центра матрицы (вертикально, горизонтально и по диагонали) будут элементы:

- $A(0, 0) = 7$
- $A(N-1-0, 0) = A(4, 0) = 7$
- $A(0, N-1-0) = A(0, 4) = 7$
- $A(N-1-0, N-1-0) = A(4, 4) = 7$

Все эти четыре значения равны. Значит элемент 7 в $(0, 0)$ — зеркальный.

В этой матрице это не единственный зеркальный элемент.

Например, центральный элемент квадратной матрицы с нечётным размером всегда является зеркальным элементом по определению, так как его зеркальные «аналоги» — это он сам.

Решение:

Один из вариантов исполнения

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

Вариант реализации программы на Python

```
def count_mirror_elements():
    """
    Считывает размер матрицы и саму матрицу из стандартного ввода,
    а затем подсчитывает количество элементов, зеркальных относительно центра
    матрицы.
    Избегает повторного подсчета одних и тех же зеркальных четверок.
    Выводит результат в стандартный вывод.
    """

    # Считываем размер матрицы из стандартного потока ввода.
    n = int(input("Введите размер матрицы N (3 <= N <= 100): "))

    # Считываем матрицу из стандартного потока ввода.
    print("Вводите элементы матрицы построчно:")
    matrix = []
```

```

for i in range(n):
    row = list(map(int, input(f"Введите {n} числа строки №{i+1}: ").split()))
    matrix.append(row)

# Подсчитываем количество зеркальных элементов.
count = 0
visited = set() # Чтобы не считать одни и те же четверки несколько раз

for i in range(n):
    for j in range(n):
        # Сортируем индексы, чтобы избежать дубликатов.
        indices = tuple(sorted([(i, j), (n - 1 - i, j), (i, n - 1 - j), (n -
1 - i, n - 1 - j)]))

        if indices not in visited:
            if (matrix[i][j] == matrix[n - 1 - i][j] ==
                matrix[i][n - 1 - j] == matrix[n - 1 - i][n - 1 - j]):
                count += 1
                visited.add(indices) # отметка о том, что эта комбинация уже
проверена

# Выводим результат в стандартный поток вывода.
print(count)

# Запускаем функцию.
count_mirror_elements()

```

Задание 5 (25 баллов)

Дана строка — описание спортивного результата участника в соревновании.

Ваша задача — составить программу для определения настроения результата по ряду признаков.

Если в тексте есть хотя бы одно слово из списка положительных результатов: "победа", "чемпион", "золото", "рекорд", — результат считается "Положительным".

Если в тексте есть хотя бы одно слово из списка отрицательных результатов "проигрыш", "ошибка", "дисквалификация", "несчастье", — результат считается "Отрицательным".

Если встречаются оба типа слов, подсчитать количество положительных и отрицательных слов и:

- если положительных больше — вывести "Положительный";
- если отрицательных больше — вывести "Отрицательный";
- если равны — вывести "Нейтральный".

Если ни одного из этих слов нет, — результат "Нейтральный".

Также, необходимо проверять наличие отрицательной частицы "не" перед ключевыми словами из списков дружелюбных и грубых слов. Если эта частица есть, то считать это слово несёт противоположный смысл.

При выполнении задания игнорировать регистр и знаки препинания. Считать каждое уникальное слово только один раз.

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

Входные данные

Одна строка — описание спортивного результата.

Выходные данные

Одна строка — "Положительный", "Отрицательный" или "Нейтральный".

Пример

Входные данные

Несмотря на ошибку, спортсмен установил рекорд и победил!

Выходные данные

Положительный.

Решение:

Один из вариантов исполнения

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

Вариант реализации программы на Python

```
def find_longest_palindrome(s):
    # Проверяем строку на валидность
    if len(s) > 100 or any(not (c.islower() and ('a' <= c <= 'я') or c == '?'))
for c in s):
    return None

def expand_around_center(left, right):
    while left >= 0 and right < len(s):
        # Совпадение обычных символов или наличие специального символа ?
        if s[left] == s[right] or s[left] == '?' or s[right] == '?':
            left -= 1
            right += 1
        else:
            break
    return s[left + 1 : right]

max_palindrome = ""
n = len(s)
for i in range(n):
    # Проверяем нечётные палиндромы (центр одиночный символ)
    palindrome_odd = expand_around_center(i, i)

    # Проверяем чётные палиндромы (центр пара символов)
```

```

palindrome_even = expand_around_center(i, i + 1)

current_max = max(palindrome_odd, palindrome_even, key=len)
if len(current_max) > len(max_palindrome):
    max_palindrome = current_max

return max_palindrome

# Основной цикл программы
if __name__ == "__main__":
    input_string = input("Введите строку: ")
    result = find_longest_palindrome(input_string)
    if result is None:
        print("Ошибка ввода!")
    else:
        print(f"Палиндром максимальной длины: {result}")

```

Задание 6 (25 баллов)

Небольшой город получает электроэнергию из двух основных источников, традиционной электростанции и солнечной электростанции. Население города растет, что увеличивает потребление электроэнергии. Производительность традиционной электростанции постепенно снижается из-за старения оборудования и других факторов. Городские власти планируют увеличить долю возобновляемой энергии, устанавливая солнечные панели на крышах зданий.

Ваша задача – разработать программу, которая по заданным параметрам оценит, будет ли достаточно электроэнергии для обеспечения потребностей города через заданное количество лет.

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

Входные данные

N_0 – начальное население города (целое число).

r – годовой процент роста населения (вещественное число, %).

E_0 – начальная годовая выработка электроэнергии традиционной электростанцией (вещественное число, кВт*ч).

q – годовой процент снижения выработки электроэнергии традиционной электростанцией (вещественное число, %).

S – общая площадь крыш, доступная для установки солнечных панелей (вещественное число, м²).

R – среднегодовая выработка электроэнергии с 1 м² солнечной панели (вещественное число, кВт*ч).

n – число лет, на которое делается прогноз (целое число).

consumption_per_person - суточное потребление одним человеком (кВт*ч)

Выходные данные

Численность населения через n лет.

Выработка электроэнергии традиционной электростанцией через n лет (кВт*ч).

Объём электроэнергии, произведённой солнечными панелями на крышах (кВт*ч).

Суммарный объём доступной электроэнергии (кВт*ч).

Общее годовое потребление электроэнергии населением (кВт*ч).

Сообщение: «Электрoэнергии достаточно» или «Электрoэнергии недостаточно».

Пример

Начальное население (человек): 10000

Годовой прирост населения (%): 2.5

Начальная выработка традиционной электростанции (кВт·ч): 5000000

Годовое снижение выработки традиционной станции (%): 1

Площадь крыш для солнечных панелей (м²): 5000

Выработка с 1 м² солнечной панели (кВт·ч/год): 300

Количество лет для прогноза: 10

Суточное потребление на человека (кВт·ч): 10

Через 10 лет:

Численность населения: 12800 человек

Выработка традиционной электростанции: 4521910.38 кВт·ч/год

Выработка солнечных панелей: 1500000.0 кВт·ч/год

Суммарная доступная электроэнергия: 6021910.38 кВт·ч/год

Годовое потребление населения: 46720000.0 кВт·ч/год

Электрoэнергии недостаточно — требуется дополнительная генерация.

Решение:

Один из вариантов решения

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

Вариант реализации программы на Python

```
def calculate_water_resources(N0, p, V0, q, S, R, n):  
    # Проверка правильности значений
```

```

if not all(isinstance(x, (int, float)) for x in [N0, p, V0, q, S, R, n]) or \
    any(x <= 0 for x in [N0, p, V0, q, S, R, n]):
    raise ValueError("Все вводимые значения должны быть положительными
числами.")

# Расчёт числа жителей через n лет
population_after_n_years = int(N0 * ((1 + p / 100)**n))

# Расчёт объёма природного источника воды через n лет
water_source_volume_after_n_years = V0 * ((1 - q / 100)**n)

# Рассчитываем объём дождевой воды, собираемый с крыш
rainwater_collected = S * R * 0.7

# Общие доступные запасы воды
total_available_water = water_source_volume_after_n_years +
rainwater_collected

# Потребление воды всеми жителями за год
annual_consumption = population_after_n_years * 20 * 365

# Сообщаем о состоянии запасов воды
if total_available_water >= annual_consumption:
    message = 'Воды достаточно'
else:
    message = 'Воды недостаточно'

return {
    'population': population_after_n_years,
    'natural_water_volume': round(water_source_volume_after_n_years),
    'collected_rainwater': round(rainwater_collected),
    'total_water': round(total_available_water),
    'annual_consumption': round(annual_consumption),
    'message': message
}

# Входные данные
while True:
    try:
        N0 = int(input("Начальная численность населения: "))
        p = float(input("Годовой прирост населения (%): "))
        V0 = float(input("Начальный объём природного источника воды (литры): "))
        q = float(input("Годовой процент снижения объёма источника (%): "))
        S = float(input("Площадь крыш для сбора дождя (кв.м): "))
        R = float(input("Среднее годовое количество осадков на 1 кв.м (литры):
"))
        n = int(input("Количество лет для прогнозирования: "))

        # Выполняем проверку корректности данных
        if any(x <= 0 for x in [N0, p, V0, q, S, R, n]):
            print("Все вводимые значения должны быть положительными числами!")

```

```
        continue

    break
except ValueError:
    print("Некорректный ввод. Используйте только числа.")

# Вычисляем ресурс воды
result = calculate_water_resources(N0, p, V0, q, S, R, n)
print(f'Через {n} лет:')
print(f'- Численность населения составит: {result["population"]}')
print(f'- Объем природного источника воды составит:
{result["natural_water_volume"]} литров')
print(f'- Объем дождевой воды, собранной с крыш: {result["collected_rainwater"]}
литров')
print(f'- Суммарный доступный объем воды: {result["total_water"]} литров')
print(f'- Годовой расход воды: {result["annual_consumption"]} литров')
print(result['message'])
```