

# Анализ космических снимков и геопространственных данных

## Заключительный этап

### Предметный тур

#### Информатика. 8–11 класс

##### *Задача VI.1.1.1. Странный классификатор (10 баллов)*

###### *Условие*

Странный классификатор анализирует пары слов одинаковой длины, определяя их бинарное сходство при помощи расстояния Хэмминга ( $S_X$ ).

Расстояние Хэмминга — это количество различающихся позиций для строк (в нашем случае — двоичных строк) с одинаковой длиной.

Например,  $S_X(101, 000) = 2$  (первые и последние символы в обеих двоичных строках разные, а средние совпадают).

Классификатор работает следующим образом.

Слова являются совпадающими, если  $S_X = 0$ .

Слова являются очень похожими, если  $0 < S_X \leq 10$ .

Слова являются похожими, если  $10 < S_X \leq 20$ .

Слова являются непохожими, если  $S_X > 20$ .

Классификатор вычисляет  $S_X$  между двоичными строками, полученными путем замены исходных символов строк на двоичное представление их кодов по кодировке Unicode.

По результатам вычислений классификатор относит конкретную пару строк к одному из классов: совпадающие, очень похожие, похожие, непохожие.

По окончании ввода всех пар строк классификатор по специальному запросу выдает все пары слов, относящихся к требуемому классу.

Требуется написать программу, которая выполняет работу описанного классификатора.

###### *Формат входных данных*

На вход программы подаются пары слов одинаковой длины. Каждая пара подается на отдельной строке и разделяется пробелом. Суммарная длина пары слов, включая пробел, не превышает 255 символов. В состав каждого слова могут входить любые символы, кроме пробела. Количество вводимых пар заранее неизвестно. Признаком конца ввода пар является 0 (ноль), вводимый вместо очередной пары. После этого программе подается на вход название одного из классов слов на английском языке: `coinciding` (совпадающие), `very similar` (очень похожие), `similar` (похо-

---

жие), `dissimilar` (непохожие).

При этом гарантируется, что данные, не отвечающие описанному здесь формату (например, строки, состоящие из трех слов, или названия несуществующих классов), на вход программы не подаются.

### *Формат выходных данных*

Программа должна вывести множество пар слов, отвечающих требуемому классу. Каждая пара выводится на отдельной строке, слова в паре разделяются пробелом. Если пар, отвечающих запрошенному классу, не оказалось, то требуется вывести слово «no».

### *Методика проверки и пояснение к тесту*

В процессе тестирования программе подается на вход 10 различных вариантов комбинаций входных данных, за каждый пройденный тест назначается по 1 баллу. Входные данные и ожидаемые выходные данные строго соответствуют описанным выше форматам. Примеры входных данных, показанные ниже, при тестировании не используются.

### *Примеры*

#### *Пример №1*

|   |
|---|
| <b>Стандартный ввод</b>   |
| aaaa aaax<br>aaaaa aaaxx<br>aaaaaa aaaxxx<br>aaaaaaa aaaxxxx<br>0<br>very similar |
| <b>Стандартный вывод</b>  |
| aaaa aaax<br>aaaaa aaaxx<br>aaaaaa aaaxxx   |

#### *Пример №2*

|  |
|--|
| <b>Стандартный ввод</b>  |
| aaaa aaax<br>aaaaa aaaxx<br>aaaaaa aaaxxx<br>aaaaaaa aaaxxxx<br>0<br>similar |
| <b>Стандартный вывод</b>   |
| aaaaaaa aaaxxxx  |

### Пример №3

| Стандартный ввод |
|------------------|
| aaaa aaax        |
| aaaaa aaaxx      |
| aaaaaa aaaxxx    |
| aaaaaaa aaaxxxx  |
| 0                |
| coinciding       |

| Стандартный вывод |
|-------------------|
| no                |

### Пример №4

| Стандартный ввод |
|------------------|
| aaaa aaax        |
| aaaaa aaaxx      |
| aaaaaa aaaxxx    |
| aaaaaaa aaaxxxx  |
| 0                |
| dissimilar       |

| Стандартный вывод |
|-------------------|
| no                |

### Решение

Задача посвящена работе со строками, кодами символов, динамическими массивами (списками), словарями.

Сначала организуется потенциально бесконечный цикл по вводу данных, который прерывается при получении на вход строки '0'. Внутри цикла вводимые строки (не равные '0') разбиваются на два слова (разделителем служит пробел).

Выделенные слова передаются в функцию вычисления расстояния Хэмминга, которая подсчитывает количество различий в двоичных векторах, образованных путем замены символов обеих строк на двоичное представление кодов символов по кодировке Unicode. Позиции, в которых двоичные вектора различаются, определяются при помощи побитовой операции «исключающее или».

Для хранения вводимых пар слов в соответствии с их классом удобно использовать словарь списков. В данном словаре ключом является название класса.

Далее, в зависимости от вычисленного расстояния Хэмминга, пары слов добавляются в тот или иной список по соответствующему ключу словаря.

Когда ввод пар слов завершен, программа ожидает ввода имени класса, получив которое, выводит пары слов, принадлежащие данному классу, в столбик — каждую пару на новой строке.

### Пример программы-решения

Ниже представлено решение на языке Python 3.

---

```

1  # -*- coding: utf-8 -*-
2  def hamming(s1,s2):
3      sx = 0
4      for i in range(len(s1)):
5          sx += bin(ord(s1[i]) ^ ord(s2[i]))[2:].count('1')
6      return sx
7
8  d = {'coinciding':[], 'very similar':[], 'similar':[], 'dissimilar':[]}
9  while True:
10     data = input()
11     if data == '0':
12         break
13     L = data.split()
14     sx = hamming(L[0],L[1])
15     if sx == 0:
16         d['coinciding'].append(data)
17     if 0 < sx <= 10:
18         d['very similar'].append(data)
19     if 10 < sx <= 20:
20         d['similar'].append(data)
21     if sx > 20:
22         d['dissimilar'].append(data)
23 M = d[input()]
24 if M == []:
25     print('no')
26 else:
27     for s in M:
28         print(s)

```

### *Задача VI.1.1.2. В мире животных (20 баллов)*

#### *Условие*

Имеется питательная среда, в которой живут некоторые существа одного вида. В начальный момент времени в среду было помещено  $N$  только что родившихся существ. Про существ известно следующее:

- время жизни одной особи составляет  $T_3 = 40$  условных единиц времени;
- в течение жизни существа размножаются почкованием в моменты времени  $T_1 = 10$  и  $T_2 = 20$ : от каждой родительской особи происходит ровно один потомок, а после почкования родительская особь продолжает жить до наступления момента времени  $T_3$ .

Требуется написать программу, которая будет определять количество живых существ в питательной среде к моменту времени  $T_4$ .

В программе требуется предусмотреть контроль входных данных на допустимость, т. е. должны проверяться следующие условия:

- $N, T_4$  — натуральные числа, при этом;
- $0 < N \leq 50$ ;
- $40 \leq T_4 \leq 500$  и  $T_4$  кратно 10-ти.

---

### **Формат входных данных**

На вход программы в одной строке через пробел подаются значения переменных  $N$ ,  $T_4$ . При этом гарантируется, что значений переменных будет ровно два.

### **Формат выходных данных**

Если введенные значения корректны, то программа должна вывести одно целое число, являющееся результатом вычислений.

Если введенные значения некорректны, то программа должна вывести строку «error» (без кавычек).

### **Методика проверки и пояснение к тесту**

Программе на вход подаются как корректные, так и некорректные для проведения расчетов варианты входных данных. Во всех случаях результат работы программы должен соответствовать описанному выше формату выходных данных. Всего в процессе тестирования программе подается на вход 20 разных вариантов входных данных, за каждый пройденный тест назначается по 1-му баллу. Примеры входных данных, показанные ниже, при тестировании не используются.

### **Примеры**

#### *Пример №1*

|                          |
|--------------------------|
| <b>Стандартный ввод</b>  |
| 2 40                     |
| <b>Стандартный вывод</b> |
| 22                       |

#### *Пример №2*

|                          |
|--------------------------|
| <b>Стандартный ввод</b>  |
| 50 500                   |
| <b>Стандартный вывод</b> |
| 2276877456200            |

#### *Пример №3*

|                          |
|--------------------------|
| <b>Стандартный ввод</b>  |
| 55 100                   |
| <b>Стандартный вывод</b> |
| error                    |

## Пример №2

|                          |
|--------------------------|
| <b>Стандартный ввод</b>  |
| 20 qwe                   |
| <b>Стандартный вывод</b> |
| error                    |

Примечание: обратите внимание, пример № 2 показывает максимально возможный результат. Подберите подходящий тип данных для хранения целочисленного результата вычислений!

### Решение

Задача посвящена моделированию прироста абстрактной популяции живых организмов. Она решается при помощи одного цикла (или при помощи рекурсии). Решение задачи немного отягощено требованиями по контролю входных данных. Логика решения нетрудно понять, если взять небольшую начальную популяцию и проследить за ее приростом в течение некоторого времени с соблюдением правил, описанных в условии. Предположим, что в начальный момент в популяции всего 2 особи ( $N = 2$ ), а  $T_4 = 40$ . Тогда развитие популяции будет выглядеть так, как показано в таблице.

| Текущее время | Количество существ по возрасту |    |    |    |    |
|---------------|--------------------------------|----|----|----|----|
|               | 0                              | 10 | 20 | 30 | 40 |
| 0             | 2                              |    |    |    |    |
| 10            | 2                              | 2  |    |    |    |
| 20            | 4                              | 2  | 2  |    |    |
| 30            | 6                              | 4  | 2  | 2  |    |
| 40            | 10                             | 6  | 4  | 2  | 2  |

В момент  $T_4 = 40$  погибнут 2 особи, от которых в начальный момент произошла вся популяция. Поэтому суммарное количество особей в популяции по всем возрастам будет равно 22.

### Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1  # -*- coding: utf-8 -*-
2  def growPop(time, T4, old0, old10, old20, old30):
3      if time == T4:
4          return old0 + old10 + old20 + old30
5      else:
6          return growPop(time+10, T4, old0+old10, old0, old10, old20)
7
8  L = input().split()
9  if L[0].isdigit() and L[1].isdigit():
10     N, T4 = map(int, L)
11     if (0 < N <= 50) and (40 <= T4 <= 500) and (T4%10 == 0):
12         print(growPop(0,T4,N,0,0,0))
13     else:
14         print("error")
15 else:
16     print("error")
```

---

### **Задача VI.1.1.3. Елочная гирлянда (30 баллов)**

Время работы — не более 2-х секунд.

#### **Условие**

К новому году Вася смастерил устройство управления светодиодной елочной гирляндой, состоящей из лампочек восьми разных цветов:  $a_7, a_6, \dots, a_0$ . Устройство включает лампочки соответствующего цвета, подавая на них напряжение, равное логической единице. Например, если на выходе устройства сформировалась комбинация 10010011, то это означает, что горят лампочки  $a_7, a_4, a_1, a_0$ . Начальная цветовая комбинация задается на пульте управления устройством путем ввода десятичного числа, лежащего в диапазоне от 0 до 255. Далее устройство выполняет свои функции, пока его не выключат, повторяя следующую последовательность шагов:

- текущее состояние преобразуется при помощи циклического сдвига вправо на 2 разряда;
- текущее состояние преобразуется при помощи инверсии разрядов с нулевого по третий включительно.

**Обратите внимание**, что «сдвиг вправо на 2 разряда» и «инверсия разрядов с нулевого по третий» **это два шага, а не один**.

Например, если начальная цветовая комбинация задана числом 55, которое в 8-разрядном двоичном коде равно 00110111, то в результате первого шага будет получена двоичная комбинация 11001101, а в результате второго — комбинация 11000010.

Напишите программу, которая вычислит, какая будет гореть комбинация лампочек через  $K$  шагов.

#### **Формат входных данных**

На вход программы на первой строке подается начальное состояние в виде десятичного числа  $N$ , лежащего в диапазоне  $0 \leq N \leq 255$ .

Следующей строкой подается число  $K$ , лежащее в диапазоне  $0 \leq K \leq 10000000$ , и задающее требуемое количество шагов работы гирлянды.

Гарантируется, что на вход программы подаются только корректные значения.

#### **Формат выходных данных**

На выходе программы должна формироваться двоичная строка, соответствующая состоянию лампочек через требуемое количество шагов. Состояние лампочек должно отображаться в порядке, заданным условием задачи:  $a_7, a_6, \dots, a_0$ .

#### **Методика проверки и пояснение к тесту**

В процессе тестирования программе подается на вход 10 различных вариантов комбинаций входных данных, за каждый пройденный тест назначается по 3 балла. Входные данные и ожидаемые выходные данные строго соответствуют описанным выше форматам. Примеры входных данных, показанные ниже, при тестировании не используются.

---

## Примеры

### Пример №1

|                          |
|--------------------------|
| <b>Стандартный ввод</b>  |
| 123<br>1000              |
| <b>Стандартный вывод</b> |
| 01111011                 |

### Пример №2

|                          |
|--------------------------|
| <b>Стандартный ввод</b>  |
| 55<br>2                  |
| <b>Стандартный вывод</b> |
| 11000010                 |

### Пример №3

|                          |
|--------------------------|
| <b>Стандартный ввод</b>  |
| 234<br>10000000          |
| <b>Стандартный вывод</b> |
| 11101010                 |

## Решение

Ключевым моментом для успешного решения данной задачи является соблюдение временных ограничений, накладываемых на работу программы.

При большом количестве шагов в указанные временные границы легко не уложиться. Такое происходит, если пытаться решить задачу «в лоб», кодируя состояние гирлянды при помощи строкового типа данных и выполняя все требуемые манипуляции как преобразования строки.

Поэтому нужно сообразить, как выполнить необходимые операции, кодируя состояние гирлянды однобайтовым целым числом.

Кроме того, здесь желательно отказаться от оформления операций сдвига и инверсии как отдельных пользовательских функций. При использовании интерпретируемых языков типа Python за счет этого отказа тоже можно сэкономить немного времени работы программы.

## Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n, k = int(input()), int(input())  
2
```



```
3 fullCycles = k//2
4 remains = k%2
5
6 for _ in range(fullCycles):
7     n = ((n & 3)*64) | (n>>2) ^ 15
8
9 if remains==1:
10     n = (n & 3)*64 | (n>>2)
11
12 print(bin(n)[2:].rjust(8, '0'))
```

## Задача VI.1.1.4. Лесорубы (40 баллов)

### Условие

Механизированной бригаде лесорубов выделена территория для работы. Но лес на этой территории растет не сплошняком, а достаточно прорежено. Конкретное расположение деревьев задается картой следующего вида. Вся территория на карте занимает площадь 8 на 8 квадратных секторов. Сектора обозначаются двумерными координатами. Сектор с координатами  $(0, 0)$  расположен в левом верхнем углу карты, поэтому ось  $X$  на карте проходит слева направо, а ось  $Y$  — сверху вниз.

В секторе  $(0, 0)$  также находится база лесорубов, откуда выезжает и куда возвращается каждая их смена. Если в секторе лесные ресурсы существенны для добычи, то данный сектор будет отмечен на карте единицей (будем считать, что в каждом таком секторе количество деревьев одинаковое). В противном случае — отмечен нулем.

Лесорубы работают несколько смен. За одну смену лесорубы могут вырубить фрагмент территории размером 3 на 3 сектора, получая с него столько древесины, сколько было единиц в пределах данного фрагмента. В результате вырубки фрагмент опустошается, после чего другая смена должна выбирать новый фрагмент для работы. Чтобы обозначить на карте конкретный фрагмент территории, достаточно указать координаты его верхнего левого сектора. Будем считать, что лесорубы достигли выбранного фрагмента, если они добрались до его верхнего левого сектора.

Выбор каждого очередного фрагмента для вырубki должен осуществляться рационально, исходя из следующих соображений.

1. Чтобы добыть древесину, лесорубы должны вместе с техникой переместиться в выбранный фрагмент, а в конце работы снова вернуться на базу. Стоимость перемещения между базой и фрагментом зависит от расстояния между ними и от стоимости перемещения на один сектор. Будем считать, что лесорубы могут перемещаться по секторам карты строго по вертикали или горизонтали.

**Внимание!** Стоимость перемещения лесорубов в пределах фрагмента в процессе работы не учитывается.

2. Каждая добытая единица древесины может быть продана по определенной цене.

Таким образом, некоторый фрагмент приносит какую-то прибыль, если выручка с продажи древесины, вырубленной на этом фрагменте, больше, чем расходы на перемещение туда и обратно.

Напишите программу, которая поможет лесорубам выбрать  $N$  фрагментов территории так, чтобы каждая следующая смена могла получить максимально возможную прибыль. Если для некоторой смены получение прибыли становится невозможным,

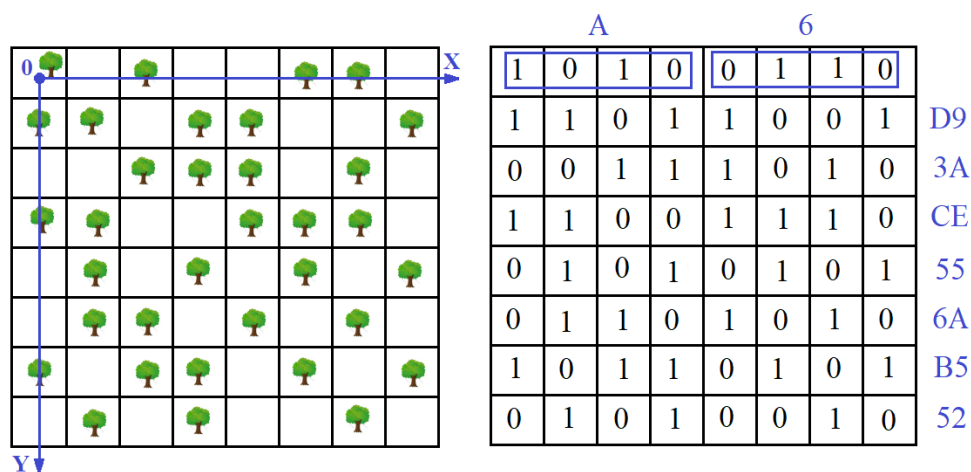
то программа должна вывести соответствующее сообщение.

Для упрощения задачи будем считать, что очередная смена приступает к планированию после окончания работы предыдущей.

**Внимание!** Анализ карты территории рекомендуется вести согласно движению слева направо и сверху вниз, т. е. начинать от базы лесорубов и двигаться направо до конца карты, после чего увеличивать координату  $Y$ .

### Формат входных данных

Сначала на вход программы подается шестнадцатеричная последовательность, кодирующая карту территории, на которой произрастает лес. В этой последовательности каждые два шестнадцатеричных разряда описывают состояние следующих 8-ми секторов на карте, если просматривать карту слева направо и далее сверху вниз. Всего в последовательности 16 шестнадцатеричных цифр. Для лучшего понимания рассмотрите пример на следующем рисунке.



В левой части рисунка деревьями отмечены сектора, представляющие интерес для добычи. В правой части рисунка эти же сектора отмечены единицами, а нулями показаны пустые сектора. Далее каждая строка карты для краткости свернута к шестнадцатеричному виду, а вся карта представляется как шестнадцатеричная последовательность A6D93ACE556AB552. Эта последовательность является примером данных, подаваемых первой строкой на вход программы.

Также обратите внимание на расположение координатных осей в левой части рисунка.

Далее второй строкой вводится  $N$  — количество смен, в течение которых лесорубы планируют работать.

Следом на новой строке вводится цена единицы древесины (натуральное число).

И наконец, следующей строкой вводится стоимость перемещения лесорубов на один сектор по горизонтали или по вертикали (натуральное число).

Гарантируется, что входные данные отвечают описанному формату, поэтому нет необходимости делать проверку данных на корректность.

---

### **Формат выходных данных**

Результатом работы программы являются  $N$  пар чисел, представленных в формате  $(x, y)$ . Каждая пара выводится на новой строке и соответствует координатам левого верхнего угла фрагмента территории размером 3 на 3 сектора, в котором будет работать соответствующая смена.

Если для некоторой смены не нашлось прибыльного фрагмента территории, то вместо координат следует вывести строку «no profitable work» (без кавычек).

### **Методика проверки и пояснение к тесту**

В процессе тестирования программе подается на вход 10 различных вариантов комбинаций входных данных, за каждый пройденный тест назначается по 4 балла. Входные данные и ожидаемые выходные данные строго соответствуют описанным выше форматам. Примеры входных данных, показанные ниже, при тестировании не используются.

### **Примеры**

#### *Пример №1*

|                                      |
|--------------------------------------|
| <b>Стандартный ввод</b>              |
| 0050275707380808<br>3<br>5<br>3      |
| <b>Стандартный вывод</b>             |
| (1,1)<br>(5,2)<br>no profitable work |

#### *Пример №2*

|  |
|--|
| <b>Стандартный ввод</b>                                    |
| C0C00000E3F3F27<br>4<br>4<br>2                             |
| <b>Стандартный вывод</b>                                   |
| (0,0)<br>(2,4)<br>no profitable work<br>no profitable work |

Пример №3

|                          |
|--------------------------|
| <b>Стандартный ввод</b>  |
| A6D93ACE556AB552         |
| 5                        |
| 4                        |
| 2                        |
| <b>Стандартный вывод</b> |
| (0,0)                    |
| (2,0)                    |
| (0,3)                    |
| no profitable work       |
| no profitable work       |

Пример №4

|                          |
|--------------------------|
| <b>Стандартный ввод</b>  |
| A6D93ACE556AB552         |
| 5                        |
| 5                        |
| 1                        |
| <b>Стандартный вывод</b> |
| (2,0)                    |
| (1,4)                    |
| (0,1)                    |
| (4,3)                    |
| (5,0)                    |

**Решение**

Первый существенный момент в решении данной задачи — это корректное восстановление карты территории из шестнадцатеричной свертки в матрицу, заполненную числами 0 и 1.

Далее эта карта анализируется в цикле  $N$  раз, где  $N$  — это количество смен, которые планируют работать лесорубы.

На каждой из  $N$  итераций вся карта разбивается на все возможные фрагменты размера 3 на 3 и для каждого фрагмента собирается следующая информация:

- ожидаемая с фрагмента прибыль;
- координаты фрагмента (как координаты его верхнего левого сектора).

Ожидаемая прибыль  $P$  рассчитывается по следующей формуле:

$$P = W \cdot C - 2 \cdot (i + j) \cdot E,$$

где  $W$  — количество единиц древесины, собираемых с фрагмента;

$C$  — цена единицы древесины;

$i, j$  — координаты фрагмента;

$E$  — стоимость перехода лесорубов между секторами по горизонтали или вертикали.

---

Обратите внимание: коэффициент 2 в формуле возникает, поскольку необходимо учесть стоимость пути с базы до фрагмента и обратно.

Указанная информация по всем фрагментам собирается в специальный двумерный массив.

Далее необходимо найти в этом массиве строку, относящуюся ко фрагменту с максимальной прибылью, и вывести в консоль координаты соответствующего фрагмента.

В конце итерации цикла по планированию работы текущей смены происходит имитация вырубки выбранного фрагмента, т. е. соответствующая область на исходной карте территории обнуляется, чтобы планирование работы новой смены происходило с учетом результатов работы предыдущей смены.

Если какая-то смена не находит для себя прибыльного фрагмента территории, то работа лесорубов останавливается.

### **Пример программы-решения**

Ниже представлено решение рассматриваемой задачи на языке Python 3 с использованием библиотеки Numpy, благодаря которой код получился максимально компактным.

Код снабжен подробными комментариями, поясняющими детали предлагаемой реализации.

```
1  # -*- coding: utf-8 -*-
2  import numpy as np
3
4  # получение зашифрованной карты и преобразование ее к двоичному виду
5  # с учетом возможных старших нулей в исходной последовательности
6  s2=bin(int(input()),16)[2:].rjust(64,'0')
7  # преобразование битовой строки s2 в Numpy-матрицу цифр размером 8 на 8
8  M = np.array([int(x) for x in s2]).reshape(8,8)
9  # получение количества смен, стоимости 1цы дерева,
10 # величины затрат на перемещение лесорубов на одну клетку
11 n, cost, expenses = [int(input()) for _ in range(3)]
12 # у каждой смены свой цикл принятия решений
13 for _ in range(n):
14     # список для сбора статистики по фрагментам размера 3 на 3
15     fragmentStat = list()
16     # перебор всех возможных фрагментов 3 на 3, существующих на карте
17     for j in range(0,len(M)-2):
18         for i in range(0,len(M[0])-2):
19             # объем работы на фрагменте (количество деревьев)
20             work = np.sum(M[j:j+3,i:i+3])
21             # подсчет ожидаемой прибыли (или убытков) при вырубке фрагмента
22             profit = work*cost - (i+j)*2*expenses
23             # добавление статистики по фрагменту в список
24             # [прибыль(убытки), X, Y]
25             fragmentsStat.append([profit,i,j])
26     # преобразование списка со статистикой в Numpy-матрицу
27     # для удобства ее последующей обработки
28     fragmentsStat = np.array(fragmentsStat)
29     # определение максимально возможной прибыли по всем фрагментам
30     mx = max(fragmentsStat[:,0])
31     if mx <= 0:
32         # если остались только бесприбыльные варианты, то у текущей смены
```

---

```
33     # уже нет выгодной работы
34     print('no profitable work')
35 else:
36     # выборка всех фрагментов, у которых прибыль равна
37     # максимально возможному значению, найденному ранее.
38     # Все эти фрагменты равноценны, поэтому берем нулевой.
39     maxProfit = fragmentsStat[fragmentsStat[:,0] == mx][0]
40     # считываем его координаты и выводим ответ в консоль
41     _, x, y = maxProfit
42     print(f'({x},{y})')
43     # имитируем вырубку выбранного фрагмента
44     M[y:y+3,x:x+3] = 0
```

Тестовые наборы для задач представлены по ссылке — [https://disk.yandex.ru/d/vn4-p3lJzv\\_d2Q](https://disk.yandex.ru/d/vn4-p3lJzv_d2Q).

## Анализ космических снимков и

### Заключительный этап

## География. 8–11 класс

### *Задача VI.1.2.1. Определение положения территории (5 баллов)*

#### *Условие*

Определить субъект Российской Федерации по его описанию.

Описание территории: территория повышенной сейсмичности на юге Восточной Сибири в таёжной зоне со значительными запасами лесных и гидроэнергоресурсов, металлических руд, золота, угля, нефти и газа. Эксплуатируются ряд крупных ГЭС, в том числе и Братская.

Варианты ответа:

1. Красноярский край;
2. Алтайский край;
3. Иркутская область;
4. Хабаровский край.

**Ответ:** 3.

### *Задача VI.1.2.2. Причины ночных заморозков (8 баллов)*

#### *Условие*

При каком типе погоды (1) и какова причина весенне-летних *ночных заморозков* (2) в умеренных широтах при дневной температуре воздуха даже более +10 °С.

**Ответ:** заморозки возможны:

1. при антициклональном типе погоды;
2. при отсутствии облачности в качестве экрана теплового излучения земли, когда ночное снижение температур достигает +15–18 °С, что превышает положительные дневные температуры.

#### *Критерии оценивания*

Общая оценка 8 баллов — по 4 балла за полные ответы по каждому из них.

### *Задача VI.1.2.3. Динамика природных зон (8 баллов)*

#### *Условие*

Основные закономерности смены лесной зоны на степную по климату (1) и водному режиму почв (2). Каковы значения и динамика показателей коэффициента увлажнения (3) и индекса сухости климата (4) при переходе от лесной к степной зонам.

---

**Ответ:**

1. снижение соотношения выпадающих осадков и величины испарения;
2. смена промывного режима почв на непромывной;
3. значение коэффициента увлажнения (отношение годовой суммы осадков к испаряемости) уменьшается в среднем с 1,2 до 0,6;
4. значение индекса сухости климата Будыко (отношение среднегодового радиационного баланса земной поверхности к количеству тепла, необходимому для испарения среднегодовой суммы осадков) увеличивается в среднем с 0,9 до 1,7.

***Критерии оценивания***

Общая оценка 8 баллов, по 2 балла за каждый полный ответ. При правильной оценке тенденции показателей учитывается их процентное отклонение от реальных значений, что и определяет снижение баллов по п.п. 3 и 4.

***Задача VI.1.2.4. Факторы формирования почв (8 баллов)***

***Условие***

Почва с максимальным содержанием гумуса (1), факторы его образования (2) и сохранения (3.1 и 3.2).

**Ответ:**

1. максимальное содержание гумуса в чернозёме;
2. его образование возможно на выровненном рельефе, где произрастает густая травяная растительность — необходимое условие образования гумуса и плодородия черноземов в целом;
3. сохраняется:
  - 3.1. при непромывном водном режиме;
  - 3.2. при температурах выше +5 °С при постоянном и достаточном количестве осадков (300–450 мм).

***Критерии оценивания***

Общая оценка 8 баллов, по 2 балла за каждый полный ответ.

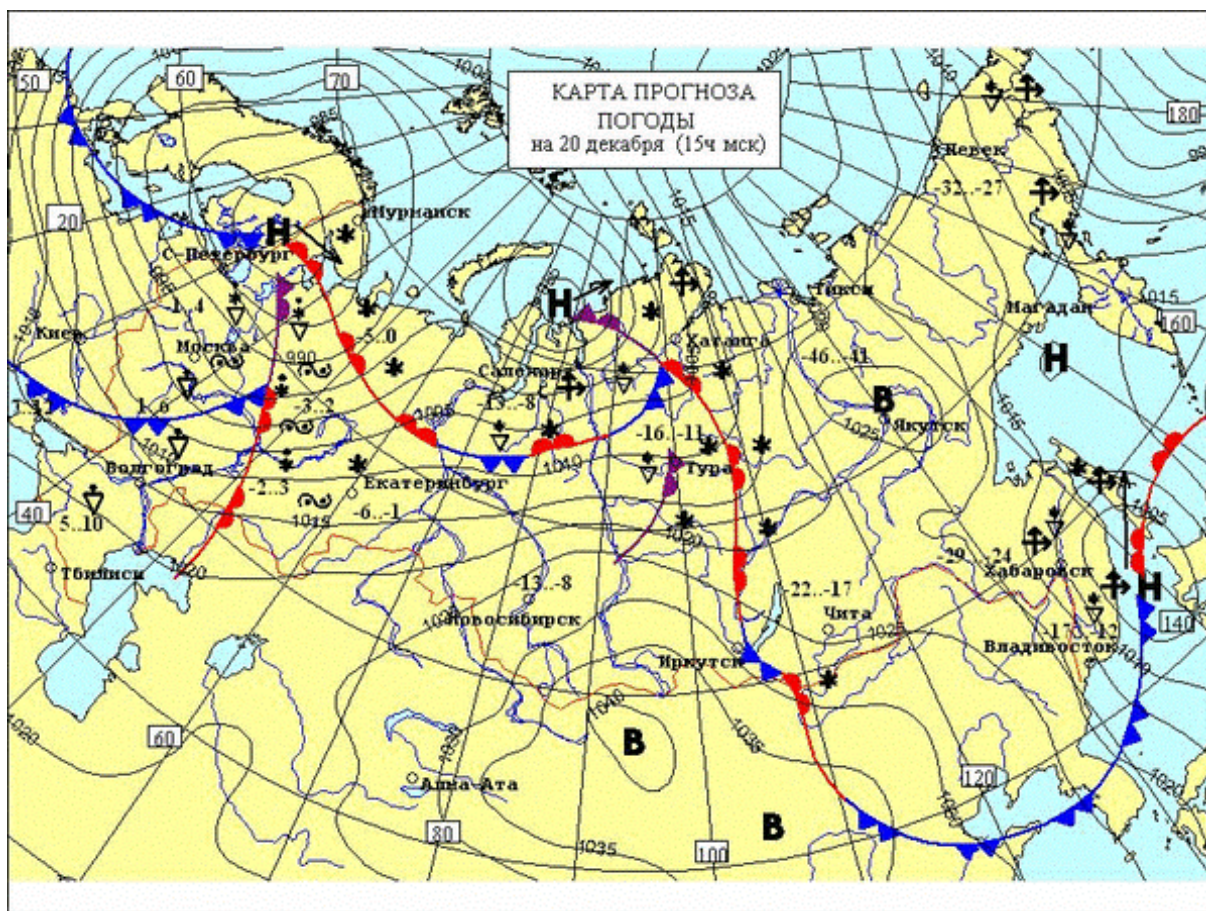
***Задача VI.1.2.5. Погода атмосферных фронтов (12 баллов)***

***Условие***

Атмосферные фронты их приуроченность и особенности погоды.

1. С каким атмосферным образованием связаны фронты?
2. Какие фронты выделены на карте?
3. Какая погода ожидается в городах Волгограде, Москве и Санкт-Петербурге?
4. Под влиянием какого фронта и чем сопровождается погода полуострова Таймыр и г. Хатанге?





Ответ:

1. Атмосферные фронты связаны в основном с циклонами.
2. На карте выделены:
  - 2.1. теплые фронты (линии и знаки красного цвета);
  - 2.2. холодные фронты (синего цвета);
  - 2.3. фронты окклюзии (сиреневого цвета).
3. В Волгограде, Москве и Санкт-Петербурге будет:
  - 3.1. понижение температуры, связанное с приходом холодных фронтов;
  - 3.2. выпадение осадков в виде снежной крупы.
4. Полуостров Таймыр и г. Хатанге находятся:
  - 4.1. под влиянием фронта окклюзии с обложным снегопадом;
  - 4.2. с метелью.

### *Критерии оценивания*

Общая оценка 12 баллов, по 1,5 балла за каждый полный ответ.

---

**Задача VI.1.2.6. Топливо-энергетические ресурсы России (10 баллов)**

**Условие**

Назвать основной состав (5 видов) топливо-энергетических ресурсов России и указать какова их часть в мировых запасах.

**Ответ:** основу составляют следующие виды:

1. природный газ (1/3 часть запасов в мире);
2. нефть (1/7 часть);
3. каменный и бурый уголь (1/8 часть);
4. энергия рек (1/20 часть);
5. энергия урановых руд (1/16 мировой добычи).

**Критерии оценивания**

Общая оценка в 10 баллов, по 1 баллу за каждую часть полного ответа.

**Задача VI.1.2.7. Нефтяные ресурсы России (10 баллов)**

**Условие**

Выделить 5 основных районов и указать их долю в российской добыче.

**Ответ:**

1. Ханты-Мансийский автономный округ составляет 42% от российской добычи;
2. Ямало-Ненецкий автономный округ — 6%;
3. Республика Татарстан — около 6%.
4. Оренбургская область — более 4%.
5. Сахалин — более 2%.

**Критерии оценивания**

Общая оценка в 10 баллов, по 1 баллу за каждую часть полного ответа.

**Задача VI.1.2.8. Локальные аномалии климата (10 баллов)**

**Условие**

Основная причина (1) и уровень различий в увлажнении (2) Грузии и Армении.

**Ответ:**

1. территория Грузии является наветренной по отношению к влагонесущим воздушным потокам, тогда как Армения остаётся подветренной с преобладанием

---

феновых (иссушающих) эффектов в межгорных районах, а также под влиянием сухих тропических воздушных масс из Аравии и Ирана;

2. количество выпадающих осадков в Грузии в 3–4 раза превышает таковые в Армении.

### ***Критерии оценивания***

Общая оценка в 10 баллов, по 5 баллов за каждую часть полного ответа.

### ***Задача VI.1.2.9. Определение положения территории (5 баллов)***

#### ***Условие***

Определить местоположения территории. Берега страны омываются Атлантическим и Тихим океанами. Население около 100 млн человек. В южной части страны проживают основатели древней культуры — майя.

Варианты ответа:

1. Аргентина;
2. Мексика;
3. Канада;
4. Бразилия.

**Ответ:** 2.

### ***Задача VI.1.2.10. Агроспециализация района (10 баллов)***

#### ***Условие***

Назвать специализацию (1.1, 1.2), основные центры (2) и виды общего распространения агропромышленного комплекса (3.1 — север, 3.2 — юг) на территории Уральского экономического района.

**Ответ:**

1. Основная специализация района:
  - 1.1. выращивание яровой пшеницы, овса и ржи;
  - 1.2. производство продукции животноводства (мясо, молоко, шерсть).
2. Сельское хозяйство наиболее развито на Южном Урале, где основные посевы зерновых.
3. Общее распространение по территории района:
  - 3.1. на севере — птицеводство, молочное скотоводство;
  - 3.2. на юге — овцеводство, мясомолочное и мясное животноводство.

### ***Критерии оценивания***

Общая оценка в 10 баллов, по 2 балла за каждую часть полного ответа.

---

**Задача VI.1.2.11. Использование космической информации (10 баллов)**

**Условие**

Какая территория, какие водные объекты и атмосферные образования представлены на космическом снимке?



**Ответ:**

1. На снимке представлена территория Восточной Европы и России, а также части Балтийского, Чёрного и Азовского морей.
2. Атмосферное образование — циклон.
3. Тёплый фронт изображен полосой сплошной облачности в правой верхней части снимка до слияния с холодным фронтом.
4. Холодный фронт изображен закруглённой полосой с более разреженной облачностью в правой нижней части.
5. Фронт окклюзии (спиралевидная часть центра циклона после слияния теплого и холодного фронтов).

**Критерии оценивания**

Общая оценка в 10 баллов, по 5 баллов за каждую часть полного ответа.

---

### **Задача VI.1.2.12. Численность населения городов (4 балла)**

#### **Условие**

Какие четыре из перечисленных городов имеют максимальную численность населения (на 2022 г. в порядке убывания и без указания численности)?

- Пермь.
- Нижний Новгород.
- Екатеринбург.
- Челябинск.
- Новосибирск.
- Красноярск.
- Хабаровск.
- Владивосток.

#### **Ответ:**

1. Новосибирск — 1, 6 млн человек;
2. Екатеринбург — 1, 5 млн человек;
3. Нижний Новгород — 1, 25 млн человек;
4. Челябинск — 1, 2 млн человек.

#### **Критерии оценивания**

Общая оценка в 4 балла, по 1 баллу за каждое правильное положение городов в итоговом списке по убыванию их численности.