

**Всероссийская проверочная работа**  
**по профильному учебному предмету «ИНФОРМАТИКА»**  
**для обучающихся по программам среднего профессионального образования,**  
**завершивших в предыдущем учебном году освоение общеобразовательных предметов,**  
**проходящих обучение по очной форме на базе основного общего образования.**

**Вариант 20274**

**Инструкция по выполнению работы**

Проверочная работа включает в себя 15 заданий. На выполнение работы по информатике отводится 2 часа (120 минут).

Записывайте ответы на задания в отведённом для этого месте в работе. В случае записи неверного ответа зачеркните его и запишите рядом новый.

При выполнении работы не разрешается пользоваться учебниками, рабочими тетрадями, справочниками, калькулятором.

При выполнении заданий Вы можете использовать черновик. Записи в черновике проверяться и оцениваться не будут.

Советуем выполнять задания в том порядке, в котором они даны. Для экономии времени пропускайте задание, которое не удаётся выполнить сразу, и переходите к следующему. Если после выполнения всей работы у Вас останется время, Вы сможете вернуться к пропущенным заданиям.

Баллы, полученные Вами за выполненные задания, суммируются. Постарайтесь выполнить как можно больше заданий и набрать наибольшее количество баллов.

***Желаем успеха!***

*Таблица для внесения баллов участника*

| Номер задания | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | Сумма баллов | Отметка за работу |
|---------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|--------------|-------------------|
| Баллы         |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |              |                   |

В заданиях используются следующие соглашения.

1. Обозначения для логических связок (операций):

- а) *отрицание* (инверсия, логическое НЕ) обозначается  $\neg$  (например,  $\neg A$ );
- б) *конъюнкция* (логическое умножение, логическое И) обозначается  $\wedge$  (например,  $A \wedge B$ ) либо  $\&$  (например,  $A \& B$ );
- в) *дизъюнкция* (логическое сложение, логическое ИЛИ) обозначается  $\vee$  (например,  $A \vee B$ ) либо  $|$  (например,  $A | B$ );
- г) *следование* (импликация) обозначается  $\rightarrow$  (например,  $A \rightarrow B$ );
- е) *тождество* обозначается  $\equiv$  (например,  $A \equiv B$ ). Выражение  $A \equiv B$  истинно тогда и только тогда, когда значения  $A$  и  $B$  совпадают (либо они оба истинны, либо они оба ложны);
- ф) символ 1 используется для обозначения истины (истинного высказывания); символ 0 – для обозначения лжи (ложного высказывания).

2. Два логических выражения, содержащих переменные, называются *равносильными* (эквивалентными), если значения этих выражений совпадают при любых значениях переменных. Так, выражения  $A \rightarrow B$  и  $(\neg A) \vee B$  равносильны, а  $A \vee B$  и  $A \wedge B$  неравносильны (значения выражений разные, например, при  $A = 1, B = 0$ ).

3. Приоритеты логических операций: инверсия (отрицание), конъюнкция (логическое умножение), дизъюнкция (логическое сложение), импликация (следование), тождество. Таким образом,  $\neg A \wedge B \vee C \wedge D$  означает то же, что и  $((\neg A) \wedge B) \vee (C \wedge D)$ . Возможна запись  $A \wedge B \wedge C$  вместо  $(A \wedge B) \wedge C$ . То же относится и к дизъюнкции: возможна запись  $A \vee B \vee C$  вместо  $(A \vee B) \vee C$ .

4. Обозначения Мбайт и Кбайт используются в традиционном для информатики смысле – как обозначения единиц измерения, чьё соотношение с единицей «байт» выражается степенью двойки.

1 Сколько единиц в двоичной записи шестнадцатеричного числа  $12F3_{16}$ ?

Ответ: \_\_\_\_\_.

2 Миша заполнял таблицу истинности функции  $(\neg x \vee \neg y) \wedge \neg(x \equiv z) \wedge w$ , но успел заполнить лишь фрагмент из трёх **различных** её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных  $w, x, y, z$ .

|   |   |   |   | $(\neg x \vee \neg y) \wedge \neg(x \equiv z) \wedge w$ |
|---|---|---|---|---------------------------------------------------------|
| 0 | 0 |   |   | 1                                                       |
|   | 0 | 1 | 1 | 1                                                       |
| 0 |   | 1 | 0 | 1                                                       |

Определите, какому столбцу таблицы соответствует каждая из переменных  $w, x, y, z$ .

В ответе напишите буквы  $w, x, y, z$  в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу, и т.д.). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

*Пример.* Функция задана выражением  $\neg x \vee y$ , зависящим от двух переменных, а фрагмент таблицы имеет следующий вид.

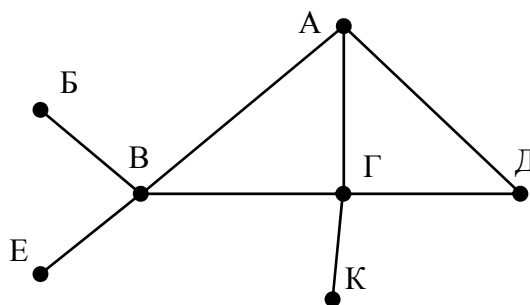
|   |   | $\neg x \vee y$ |
|---|---|-----------------|
| 0 | 1 | 0               |

В этом случае первому столбцу соответствует переменная  $y$ , а второму столбцу – переменная  $x$ . В ответе следует написать  $yx$ .

Ответ: \_\_\_\_\_.

3 На рисунке справа схема дорог  $N$ -ского района изображена в виде графа, в таблице содержатся сведения о протяжённости каждой из этих дорог (в километрах).

|    | П1 | П2 | П3 | П4 | П5 | П6 | П7 |
|----|----|----|----|----|----|----|----|
| П1 |    | 6  | 7  | 5  |    |    | 3  |
| П2 | 6  |    |    |    |    |    |    |
| П3 | 7  |    |    | 11 |    |    | 12 |
| П4 | 5  |    | 11 |    | 2  | 4  |    |
| П5 |    |    |    | 2  |    |    |    |
| П6 |    |    |    | 4  |    |    |    |
| П7 | 3  |    | 12 |    |    |    |    |



Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова протяжённость дороги из пункта А в пункт В. В ответе запишите целое число – так, как оно указано в таблице.

Ответ: \_\_\_\_\_.

4

Ниже представлены два фрагмента таблиц из базы данных о жителях микрорайона. Каждая строка таблицы 2 содержит информацию о ребёнке и об одном из его родителей. Информация представлена значением поля ID в соответствующей строке таблицы 1. Определите на основании приведённых данных, для скольких жителей выполнено условие, что она (он) родилась(-лся) в том же городе, что и её (его) бабушка или дедушка. При вычислении ответа учитывайте только информацию из приведённых фрагментов таблиц.

| Таблица 1 |                |     |                |
|-----------|----------------|-----|----------------|
| ID        | Фамилия И.О.   | Пол | Место рождения |
| 137       | Бондарчук О.О. | М   | Бийск          |
| 138       | Бондарчук С.И. | Ж   | Барнаул        |
| 139       | Митта Г.О.     | Ж   | Бийск          |
| 143       | Митта К.О.     | М   | Абакан         |
| 144       | Митта О.А.     | М   | Иркутск        |
| 146       | Митта С.К.     | Ж   | Абакан         |
| 147       | Ромм Г.О.      | Ж   | Иркутск        |
| 148       | Ромм Н.К.      | Ж   | Омск           |
| 157       | Ромм О.К.      | Ж   | Иркутск        |
| 158       | Ромм С.К.      | Ж   | Омск           |
| 159       | Хитрук Г.И.    | Ж   | Иркутск        |
| 162       | Хитрук Е.О.    | Ж   | Бийск          |
| 163       | Хитрук О.И.    | Ж   | Иркутск        |
| 165       | Хитрук Т.И.    | М   | Иркутск        |
| ...       | ...            | ... | ...            |

| Таблица 2   |            |
|-------------|------------|
| ID Родителя | ID Ребёнка |
| 137         | 139        |
| 138         | 139        |
| 144         | 143        |
| 139         | 146        |
| 143         | 146        |
| 144         | 147        |
| 147         | 148        |
| 147         | 157        |
| 147         | 158        |
| 162         | 159        |
| 137         | 162        |
| 138         | 162        |
| 162         | 163        |
| 162         | 165        |
| ...         | ...        |

Ответ: \_\_\_\_\_.

5

Для кодирования растрового рисунка, напечатанного с использованием шести красок, применили неравномерный двоичный код. Для кодирования цветов используются кодовые слова.

| Цвет    | Кодовое слово |
|---------|---------------|
| Белый   | 0             |
| Зелёный | 1111          |
| Красный | 1110          |

| Цвет       | Кодовое слово |
|------------|---------------|
| Синий      |               |
| Фиолетовый | 11110         |
| Чёрный     | 10            |

Укажите кратчайшее кодовое слово для кодирования синего цвета, при котором код будет удовлетворять условию Фано. Если таких кодов несколько, укажите код с **наименьшим** числовым значением.

*Примечание.* Условие Фано означает, что никакое кодовое слово не является началом другого кодового слова. Это обеспечивает возможность однозначной расшифровки закодированных сообщений.

Ответ: \_\_\_\_\_.

6

На вход алгоритма подаётся натуральное число  $N$ . Алгоритм строит по нему новое число  $R$  следующим образом.

1) Строится двоичная запись числа  $N$ .

2) К этой записи дописываются справа ещё два разряда по следующему правилу:

а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;

б) над этой записью производятся те же действия – справа дописывается остаток от деления суммы цифр на 2.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа  $N$ ) является двоичной записью искомого числа  $R$ .

Укажите такое **наименьшее** число  $N$ , для которого результат работы алгоритма больше 45. В ответе это число запишите в десятичной системе счисления.

Ответ: \_\_\_\_\_.

7

Дан фрагмент электронной таблицы. Из ячейки A2 в ячейку B1 была скопирована формула. При копировании адреса ячеек в формуле автоматически изменились. Каким стало числовое значение формулы в ячейке B1?

|   | A             | B | C   | D    | E     |
|---|---------------|---|-----|------|-------|
| 1 | 40            |   | 400 | 4000 | 40000 |
| 2 | = C\$2 + \$D3 | 3 | 300 | 3000 | 30000 |
| 3 | 20            | 2 | 200 | 2000 | 20000 |
| 4 | 10            | 1 | 100 | 1000 | 10000 |

*Примечание.* Знак \$ обозначает абсолютную адресацию.

Ответ: \_\_\_\_\_.

8

Запишите число, которое будет напечатано в результате выполнения следующей программы. Для Вашего удобства программа представлена на пяти языках программирования.

| Бейсик                                                                                                                                                                                                                         | Python                                                                                                                                                                                      |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> DIM S, N AS INTEGER S = 0 N = 90 WHILE S + N &lt; 135     S = S + 20     N = N - 10 WEND PRINT S </pre>                                                                                                                  | <pre> s = 0 n = 90 while s + n &lt; 135:     s = s + 20     n = n - 10 print(s) </pre>                                                                                                      |
| Алгоритмический язык                                                                                                                                                                                                           | Паскаль                                                                                                                                                                                     |
| <pre> алг нач     цел n, s     s := 0     n := 90     нц пока s + n &lt; 135         s := s + 20         n := n - 10     кц     вывод s кон </pre>                                                                             | <pre> var s, n: integer; begin     s := 0;     n := 90;     while s + n &lt; 135 do         begin             s := s + 20;             n := n - 10;         end;     writeln(s) end. </pre> |
| C++                                                                                                                                                                                                                            |                                                                                                                                                                                             |
| <pre> #include &lt;iostream&gt; using namespace std;  int main() {     int s = 0, n = 90;     while (s + n &lt; 135) {         s = s + 20;         n = n - 10;     }     cout &lt;&lt; s &lt;&lt; endl;     return 0; } </pre> |                                                                                                                                                                                             |

Ответ: \_\_\_\_\_.

9

В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес, – в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда – нули. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.

Для узла с IP-адресом 111.81.208.27 адрес сети равен 111.81.192.0. Чему равно наименьшее возможное значение третьего слева байта маски? Ответ запишите в виде десятичного числа.

Ответ: \_\_\_\_\_.

10

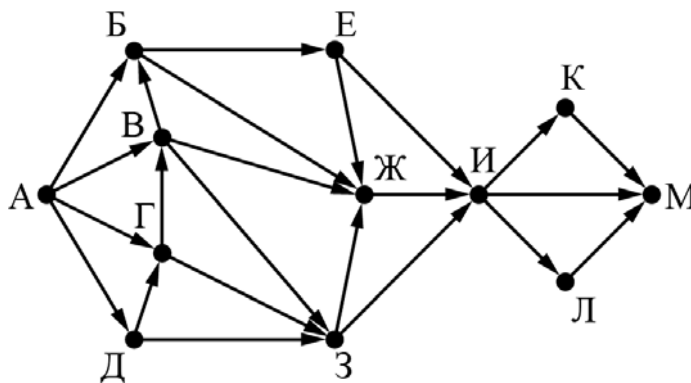
При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 7 символов и содержащий только символы из 26-символьного набора прописных латинских букв. В базе данных для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым и минимально возможным количеством бит. Кроме собственно пароля, для каждого пользователя в системе хранятся дополнительные сведения, для чего выделено целое число байт; это число одно и то же для всех пользователей.

Для хранения сведений о 15 пользователях потребовалось 300 байт. Сколько байт выделено для хранения дополнительных сведений об одном пользователе? В ответе запишите только целое число – количество байт.

Ответ: \_\_\_\_\_.

11

На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К, Л, М. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город М, проходящих через город Ж?



Ответ: \_\_\_\_\_.



12

В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для обозначения логической операции «И» – символ «&». В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

| Запрос                     | Найдено страниц<br>(в сотнях тысяч) |
|----------------------------|-------------------------------------|
| <i>Горло</i>               | 35                                  |
| <i>Корабль</i>             | 35                                  |
| <i>Нос</i>                 | 40                                  |
| <i>Корабль &amp; Нос</i>   | 20                                  |
| <i>Горло &amp; Нос</i>     | 13                                  |
| <i>Горло &amp; Корабль</i> | 0                                   |

Какое количество страниц (в сотнях тысяч) будет найдено по запросу

*Горло | Корабль | Нос?*

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ: \_\_\_\_\_.

13

Исполнитель К17 преобразует число, записанное на экране.

У исполнителя есть три команды, которым присвоены номера:

**1. Прибавить 1**

**2. Прибавить 2**

**3. Умножить на 2**

Первая из них увеличивает число на экране на 1, вторая увеличивает его на 2, третья умножает на 2.

Программа для исполнителя К17 – это последовательность команд.

Сколько существует таких программ, которые преобразуют исходное **число 3** в **число 13** и при этом траектория вычислений программы содержит **числа 9** и **11**? Траектория должна содержать оба указанных числа.

Траектория вычислений программы – это последовательность результатов выполнения всех команд программы. Например, для программы **132** при исходном числе 7 траектория будет состоять из чисел 8, 16, 18.

Ответ: \_\_\_\_\_.

14

На обработку поступает последовательность из четырёх неотрицательных целых чисел (некоторые числа могут быть одинаковыми). Нужно написать программу, которая выводит на экран количество чётных чисел в исходной последовательности и максимальное чётное число. Если чётных чисел нет, требуется на экран вывести «NO». Известно, что вводимые числа не превышают 1000. Программист написал программу неправильно. Ниже написанная им программа для Вашего удобства приведена на пяти языках программирования.

| Бейсик                                                                                                                                                                                                                                                                                                                                                                                                                  | Python                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> CONST n = 4 count = 0 maximum = 1000 FOR I = 1 TO n     INPUT x     IF x mod 2 = 0 THEN         count = count + 1         IF x &gt; maximum THEN             maximum = I         END IF     END IF NEXT I IF count &gt; 0 THEN     PRINT count     PRINT maximum ELSE     PRINT "NO" END IF </pre>                                                                                                                | <pre> n = 4 count = 0 maximum = 1000 for i in range(1, n+1):     x = int(input())     if x % 2 == 0:         count += 1         if x &gt; maximum:             maximum = x if count &gt; 0:     print(count)     print(maximum) else:     print("NO") </pre>                                                                                                                                                                                                                                                                                                                                                       |
| Алгоритмический язык                                                                                                                                                                                                                                                                                                                                                                                                    | Паскаль                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <pre> алг нач     цел n = 4     цел i, x     цел maximum, count     count := 0     maximum := 1000     нц для i от 1 до n         ввод x         если mod(x, 2) = 0 то             count := count + 1             если x &gt; maximum то                 maximum := x         все     все     кц     если count &gt; 0 то         вывод count, нс         вывод maximum     иначе         вывод "NO"     все кон </pre> | <pre> const n = 4; var i, x: integer; var maximum, count: integer; begin     count := 0;     maximum := 1000;     for i := 1 to n do         begin             read(x);             if x mod 2 = 0 then                 begin                     count := count + 1;                     if x &gt; maximum then                         maximum := x                     end                 end             end;         if count &gt; 0 then             begin                 writeln(count);                 writeln(maximum)             end         else             writeln('NO')         end. end. </pre> |

C++

```
#include <iostream>
using namespace std;

int main() {
    const int n = 4;
    int x, maximum, count;
    count = 0;
    maximum = 1000;
    for (int i = 1; i <= n; i++) {
        cin >> x;
        if (x % 2 == 0) {
            count++;
            if (x > maximum)
                maximum = i;
        }
    }
    if (count > 0) {
        cout << count << endl;
        cout << maximum << endl;
    }
    else
        cout << "NO" << endl;
    return 0;
}
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе последовательности:

2 15 44 15.

2. Приведите пример такой последовательности, содержащей хотя бы одно чётное число, что при её вводе приведённая программа, несмотря на ошибки, выведет правильный ответ.

3. Найдите допущенные программистом ошибки и исправьте их. Исправление ошибки должно затрагивать только строку, в которой находится ошибка. Для каждой ошибки:

1) выпишите строку, в которой сделана ошибка;

2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Известно, что в тексте программы нужно исправить не более двух строк так, чтобы она стала работать правильно.

Достаточно указать ошибки и способ их исправления для одного языка программирования.

Обратите внимание на то, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения.

*Примечание.* 0 делится на любое натуральное число.

[illegible]

15

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в одну из куч (по своему выбору) **один** камень или увеличить количество камней в куче в **два раза**. Например, пусть в одной куче 10 камней, а в другой 7 камней; такую позицию в игре будем обозначать  $(10, 7)$ . Тогда за один ход можно получить любую из четырёх позиций:  $(11, 7)$ ,  $(20, 7)$ ,  $(10, 8)$ ,  $(10, 14)$ . Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 75. Победителем считается игрок, сделавший последний ход, т.е. первым получивший такую позицию, при которой в кучах будет 75 или больше камней.

В начальный момент в первой куче было пять камней, во второй куче –  $S$  камней;  $1 \leq S \leq 69$ .

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. В описание выигрышной стратегии **не следует** включать ходы играющего по этой стратегии игрока, не являющиеся для него безусловно выигрышными, т.е. не являющиеся выигрышными независимо от игры противника.

Выполните следующие задания

#### Задание 1

- а) Укажите все такие значения числа  $S$ , при которых Петя может выиграть за один ход.
- б) Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Укажите минимальное значение  $S$ , когда такая ситуация возможна.

#### Задание 2

Укажите такое значение  $S$ , при котором у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

- Петя не может выиграть за один ход;
- Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

Для указанного значения  $S$  опишите выигрышную стратегию Пети.

#### Задание 3

Укажите значение  $S$ , при котором одновременно выполняются два условия:

- у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;
- у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения  $S$  опишите выигрышную стратегию Вани.

Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы).

В узлах дерева указывайте позиции, на рёбрах рекомендуется указывать ходы. Дерево не должно содержать партии, невозможные при реализации выигрывающим игроком своей выигрышной стратегии. Например, полное дерево игры не является верным ответом на это задание.

[illegible]