

«Мир информатики»

Журнал для тех, у кого информатика – любимый школьный предмет

Выпуск № 56, май 2021 г.

ЕГЭ

Задание 17 из демонстрационного варианта ЕГЭ по информатике 2021 года [1]

В Спецификации [2] применительно к заданию 17 в качестве проверяемого элемента содержания указано: «Умение создавать собственные программы (20–40 строк) для обработки целочисленной информации».

В общем виде задание формулируется так: «Рассматривается множество целых чисел, принадлежащих отрезку [...; ...], которые *условие, связанное со значением чисел*. Определить¹:

- количество таких чисел и максимальное из них. В ответе записать два числа через пробел: сначала количество, затем максимальное число» (обозначим такой тип задач – “тип 1”);
- количество таких чисел и минимальное из них. В ответе записать два числа через пробел: сначала количество, затем минимальное число» (“тип 2”);
- сумму таких чисел и максимальное из них. В ответе записать два числа через пробел: сначала сумму, затем максимальное число. Гарантируется, что искомая сумма не превосходит ...» (“тип 3”);
- среднее арифметическое таких чисел. Ответ записать с ... значащими цифрами после запятой» (“тип 4”);
- разность между максимальным и минимальным из них» (“тип 5”);
- медиану² таких чисел» (“тип 6”)).

Пример задачи типа 1. Рассматривается множество целых чисел, принадлежащих отрезку [792; 6737], которые делятся на 5 и не делятся на 11, 17, 19 и 23. Найти количество таких чисел и максимальное из них. В ответе записать два числа через пробел: сначала количество, затем максимальное число.

Пример задачи типа 2. Рассматривается множество целых чисел, принадлежащих отрезку [1067; 8537], которые делятся на 3 и не делятся на 13, 21, 29 и 31. Найти количество таких чисел и минимальное из них. В ответе записать два числа через пробел: сначала количество, затем минимальное число.

¹ Указывается один или несколько перечисленных вариантов.

² Медианой ряда чисел называется число, стоящее посередине упорядоченного по возрастанию ряда (в случае, если количество чисел нечетное). Если же количество чисел в ряду четно, то медианой ряда является полусумма двух стоящих посередине чисел упорядоченного по возрастанию ряда. Например, для чисел 5, 2, 18, 8, 3 медиана равна 5, для чисел 5, 17, 3, 9, 14, 2 – 7.

Общая схема программы решения задач двух указанных типов следующая:

1) при поиске максимального числа:

1. Присваивание переменной-счетчику начального значения
2. **цикл** для каждого числа из заданного диапазона в порядке возрастания
 - если** указанное в задаче условие соблюдается
 - то**
 - Увеличить значение переменной-счетчика на 1
 - Принять в качестве максимального значения текущее число
 - все**
- конец цикла**
3. Вывод ответов

где переменная-счетчик – переменная, соответствующая искомому количеству чисел.

Обратим внимание на то, что находить максимальное значение так, как находится максимальный элемент массива или максимальное числа последовательности:

1. Присваивание переменной-счетчику начального значения
2. Присваивание начального значения переменной, соответствующей искомому максимальному значению
3. **цикл** для каждого числа из заданного диапазона в порядке возрастания
 - если** указанное в задаче условие соблюдается
 - то**
 - Увеличить значение переменной-счетчика на 1
 - Сравнить текущее число с максимальным значением
 - если** оно больше
 - то**
 - Принять в качестве максимального значения текущее число
 - все**
- все**
- конец цикла**
4. Вывод ответов

– необходимости нет. Поскольку числа рассматриваются в порядке их возрастания, то последнее найденное число и будет искомым результатом.

2) при поиске минимального числа:

1. Присваивание переменной-счетчику начального значения
2. **цикл** для каждого числа из заданного диапазона в порядке убывания
 - если** указанное в задаче условие соблюдается
 - то**
 - Увеличить значение переменной-счетчика на 1
 - Принять в качестве минимального значения текущее число
 - все**
- конец цикла**
3. Вывод ответов

Здесь также использована особая методика нахождения минимального значения.

В программе решения задачи примера задачи типа 1 используем следующие переменные величин:

- n – исследуемые натуральные числа;
- кол – количество чисел, удовлетворяющих условию;
- макс – максимальное из чисел, удовлетворяющих условию.

Прежде чем приводить программу, напомним, что проверить кратность одного целого числа другому можно с помощью функции mod (в программе на школьном алгоритмическом языке) или с использованием операции определения остатка (в программах на других языках).

Вся программа имеет вид:

```
алг Пример_задачи_типа_1
нач цел n, кол, макс
  кол := 0 | Начальное значение переменной-счетчика
  нц для n от 792 до 6737 | Рассматриваем все числа диапазона
                        | в порядке возрастания
    | Проверяем условие
    если mod(n, 5) = 0 и mod(n, 11) > 0 и mod(n, 17) > 0
      и mod(n, 19) > 0 и mod(n, 23) > 0
    то
      | Увеличиваем значение переменной-счетчика
      кол := кол + 1
      | Принимаем текущее значение n
      | в качестве максимального числа
      макс := n
    все
  кц
  | Выводим ответы
  вывод нс, кол, " ", макс
кон
```

Обратим внимание на то, что попытка сократить условие для подсчета подходящих чисел:

```
алг Пример_задачи_типа_1
нач цел n, кол, макс
  кол := 0 | Начальное значение переменной-счетчика
  нц для n от 792 до 6737 | Рассматриваем все числа диапазона
                        | в порядке возрастания
    | Проверяем условие
    если mod(n, 5) = 0 и mod(n, 11 * 17 * 19 * 23) > 0
    то
      | Увеличиваем значение переменной-счетчика
      кол := кол + 1
      | Принимаем текущее значение n
      | в качестве максимального числа
      макс := n
    все
  кц
  | Выводим ответы
  вывод нс, кол, " ", макс
кон
```

– приведет к неправильному результату (почему – подумайте).

Программа решения примера задачи типа 2 аналогична; отличие только в том, что все числа диапазона рассматриваются от большего к меньшему:

```
алг Пример_задачи_типа_2
нач цел n, кол, мин
  кол := 0 | Начальное значение переменной-счетчика
  нц для n от 8537 до 1067 шаг -1 | Рассматриваем все числа
                                | диапазона в порядке убывания
    | Проверяем условие
    если mod(n, 3) = 0 и mod(n, 13) > 0 и mod(n, 21) > 0
      и mod(n, 29) > 0 и mod(n, 31) > 0
    то
      | Увеличиваем значение переменной-счетчика
      кол := кол + 1
      | Принимаем текущее значение n в качестве минимального числа
```

```

        мин := n
    все
кц
| Выводим ответы
вывод нс, кол, " ", мин
кон

```

Пример задачи типа 3. Рассматривается множество целых чисел, принадлежащих числовому отрезку [3294; 8199], которые делятся на 5 и не делятся на 11, 17, 19 и 23. Найдите наибольшее из таких чисел и их сумму. Гарантируется, что искомая сумма не превосходит 10^{10} .

Решение

Программа решения данной задачи отличается от программы решения задачи типа 1 тем, что вместо переменной-счетчика используется переменная-сумматор. Разработайте ее самостоятельно.

Пример задачи типа 4. Рассматривается множество целых чисел, принадлежащих отрезку [792; 6737], которые делятся на 5 и не делятся на 11, 17, 19 и 23. Определить среднее арифметическое таких чисел. Ответ записать с двумя значащими цифрами после запятой.

Решение

Так как для расчета искомого значения нужно знать не только количество указанных в условии чисел, но и их сумму, то общая схема программы решения задачи такая:

1. Присваивание переменной-счетчику и переменной-сумматору начальных значений
2. **цикл** для каждого числа из заданного диапазона в порядке возрастания
если указанное в задаче условие соблюдается
то
 Увеличить значение переменной-счетчика на 1
 Учесть текущее число в значении переменной-сумматора
все
конец цикла
3. Расчет среднего арифметического
4. Вывод ответа

где переменная-сумматор – переменная, соответствующая сумме учитываемых чисел.

Прежде чем представлять программы решения задачи, заметим, что в школьном алгоритмическом языке нет возможности так называемого «форматированного вывода» вещественных значений. Поэтому для вывода результата с двумя цифрами в дробной части можно применить такой прием: умножить его на 100, взять целую часть произведения (использовать функцию `int()`), а ее разделить на 100:

```

алг Пример_задачи_типа_4
нач цел n, кол, сумма, вещ сред_арифм
    кол := 0
    сумма := 0
нц для n от 792 до 6737
    | Проверяем условие
    если mod(n, 5) = 0 и mod(n, 11) > 0 и mod(n, 17) > 0
        и mod(n, 19) > 0 и mod(n, 23) > 0
    то
        | Увеличиваем значение переменной-счетчика
        кол := кол + 1
        | Учитываем значение n в общей сумме
        сумма := сумма + n
    все

```

```

кц
| Рассчитываем и выводим ответ
сред_арифм := сумма/кол
вывод int(сред_арифм * 100)/100
кон

```

Пример задачи типа 5. Рассматривается множество целых чисел, принадлежащих отрезку [792; 6737], которые делятся на 5 и не делятся на 11, 17, 19 и 23. Определить разность между максимальным и минимальным из них.

Решение

Задачу можно рассматривать как «сумму» задач типа 1 и типа 2, то есть решить ее так:

- 1) рассмотрев все числа отрезка в порядке возрастания, найти максимальное из соответствующих условию;
- 2) рассмотрев все числа в порядке убывания, найти минимальное;
- 3) определить разность найденных экстремумов.

Можно также найти искомое значение, так сказать, «за один проход» (при однократном рассмотрении всех чисел заданного отрезка) на основе следующих рассуждений:

1. Рассмотреть все числа отрезка в порядке возрастания.
2. Первое число, соответствующее условию, принять в качестве минимального.
3. Максимальное число найти, как в задаче типа 1 (это будет последнее из подходящих чисел отрезка).
4. Определить разность найденных экстремумов.

Зафиксировать факт нахождения первого числа, соответствующего условию, можно, используя переменную логического типа. В приведенной ниже программе ее имя – мин_найден. Остальные особенности программы описаны в комментариях.

```

алг Пример_задачи_типа_5
нач цел n, макс, мин, лог мин_найден
    мин_найден := нет
    нц для n от 792 до 6737
        | Рассматриваем все числа диапазона
        | в порядке возрастания
        | Проверяем условие
        если mod(n, 5) = 0 и mod(n, 11) > 0 и mod(n, 17) > 0
            и mod(n, 19) > 0 и mod(n, 23) > 0
        то | Число подходит
            если не мин_найден | Если это первое подходящее число
                то
                    | Принимаем его в качестве минимального числа
                    мин := n
                    | Фиксируем факт нахождения минимального числа
                    мин_найден := да
            иначе | Это одно из остальных подходящих чисел
                | Принимаем его в качестве максимального числа
                макс := n
    все
все
кц
| Выводим ответ
вывод макс - мин
кон

```

В программе на языке Python удобно все подходящие по условию числа записывать в список (используя метод `append`), а затем определить искомую разность с помощью функций `max` и `min`:

```
mas = []
for n in range(792, 6738):
    if n % 5 == 0 and n % 11 > 0 and n % 17 > 0 and n % 19 > 0
        and n % 23 > 0:
        mas.append(n)
print(max(mas) - min(mas))
```

Пример задачи типа 6. Рассматривается множество целых чисел, принадлежащих отрезку $[792; 6737]$, которые делятся на 5 и не делятся на 11, 17, 19 и 23. Определить медиану чисел этого множества.

Решение

Так как понадобятся все значения чисел, указанных в условии, то здесь без массива не обойтись. Поскольку количество чисел, удовлетворяющих условию, неизвестно, размер этого массива должен быть принят «с запасом». Можно принять размер массива равным, например, трети разности значений на границах отрезка. Прежде чем представлять программу решения задачи, обратим внимание на то, что числа в массив записываются в порядке возрастания, что облегчает поиск медианы.

```
алг Пример_задачи_типа_5
нач цел n, кол, вещ медиана, цел таб мас[1 : div(6737 - 792, 3)]
кол := 0
нц для n от 792 до 6737
    | Проверяем условие
    если mod(n, 5) = 0 и mod(n, 11) > 0 и mod(n, 17) > 0
        и mod(n, 19) > 0 и mod(n, 23) > 0
    то | Встретилось очередное подходящее число
        кол := кол + 1 | Его номер
        мас[кол] := n | Записываем число в массив
все
кц
если mod(кол, 2) = 1 | Размер массива - нечетный
то
    медиана := мас[div(кол, 2) + 1]
иначе
    медиана := (мас[div(кол, 2)] + мас[div(кол, 2) + 1])/2
все
| Выводим ответ
вывод медиана
кон
```

Язык Python

```
mas = []
for n in range(792, 6738):
    if n % 5 == 0 and n % 11 > 0 and n % 17 > 0 and n % 19 > 0
        and n % 23 > 0:
        mas.append(n)
if len(mas) % 2 == 1:
    mediana = mas[len(mas)//2]
else:
    mediana = (mas[len(mas)//2 - 1] + mas[len(mas)//2])/2
print(mediana)
```

Задачи для самостоятельной работы

См. [3], файл kege4.doc

Разработайте также программы, в которых применительно к условиям приведенных в [3] задач определяется:

- среднее арифметическое таких чисел (ответ должен быть записан с тремя значащими цифрами после запятой);
- разность между максимальным и минимальным из них;
- медиана указанного в условии множества чисел.

В заключение обратим внимание на то, что во всех перечисленных выше задачах в условии фигурировала кратности или «некратность» чисел. Возможны также более сложные условия. Приведем примеры и комментарии к решению соответствующих задач.

1. «... сумма цифр которых больше (или меньше) *[указывается значение]*».

Здесь возникает задача расчета суммы всех цифр числа:

```
сум := 0      | Начальное значение суммы цифр
нц пока n > 0
    посл := mod(n, 10)
    сум := сум + посл
    n := div(n, 10)
кц
```

и сравнение ее с заданным в условии значением.

2. «... сумма цифр которых кратна *[указывается значение]*».

После нахождения суммы всех цифр числа проверяется остаток от ее деления на значение, заданное в условии.

3. «... произведение цифр которых больше (или меньше) *[указывается значение]*».

4. «... произведение цифр которых кратно *[указывается значение]*».

Отличие программ решения двух последних задач от задач с условием в примерах 1 и 2 в том, что рассчитывается не сумма, а произведение цифр числа:

```
произв := 1    | Начальное значение
нц пока n > 0
    посл := mod(n, 10)
    произв := произв * посл
    n := div(n, 10)
кц
```

5. «... одна из цифр которых равна *[указывается значение]*».

Здесь для каждого числа при выделении его цифр следует проверять каждую цифру на равенство заданному значению. Факт встречи нужной цифры можно зафиксировать с помощью величины логического типа (или принимающей значение 0 и 1), и в зависимости от окончательного значения этой величины учитывать текущее число в количестве всех таких чисел, их сумме и т. п.

6. «... двоичная запись которых оканчивается (или не оканчивается) на 0 (или 00, или 000, или т. п.)» [12]³.

При решении таких задач нет необходимости переводить числа в двоичную систему счисления. Можно учесть, что для двоичная запись десятичного числа оканчивается на 0, если это число четное, на 00 – если оно кратно четырем и т. д. Это значит, что достаточно проверять кратность или «некратность» проверяемых чисел.

³ Автор задач, аналогичных задачам примеров 5–10, – А.Куканова.

7. «... запись которых в двоичной системе оканчивается на 11».

Анализ двоичных чисел, оканчивающихся на 11 (11, 111, 1011, 1111 и т. д.) показывает, что это десятичные числа 3, 7, 11, 15 и т. д., общее свойство которых такое: «остаток от их деления на 4 равен 3». Это свойство и следует использовать в программе в условии для учета исследуемых целых чисел.

8. «... запись которых в девятеричной (или другой) системе оканчивается на [указывается цифра]».

Здесь также нет необходимости переводить числа в девятеричную (или другую) систему счисления. Нужная последняя цифра равна остатку от деления проверяемого числа на 9 (или на основание другой системы) – вспомните метод перевода десятичных чисел в другие системы счисления методом деления числа на основание «новой» системы.

9. «... для которых запись в двоичной и четверичной системах счисления оканчивается одинаковой цифрой».

Для данного условия свойства соответствующих десятичных чисел можно получить на основе следующих рассуждений.

Возможная одинаковая цифра в условии – 0 или 1.

Двоичная запись десятичного числа оканчивается на 0, если это число четное, четверичная запись – если оно кратно четырем. Общий признак, дающий цифру 0, – «число кратно четырем» (значения 6, 10 и т. п. – не подходят)

И двоичная, и четверичная записи десятичного числа оканчивается на 1, если это число нечетное (убедитесь в этом!).

Это значит, что условие для учета соответствующих десятичных чисел можно сформулировать так: «число кратно четырем или нечетное».

10. «... запись которых в пятеричной системе имеет не менее 6 цифр и оканчивается на 21 или 23» (или с другими значениями).

Здесь рассмотрим два признака.

Во-первых, можно определить, какое минимальное десятичное число в пятеричной системе являются шестизначным. Максимальное пятеричное пятизначное число – 44444. Соответствующее десятичное – 3124. Значит, уже число 3125 будет в пятеричной системе шестизначным. Именно с него и должен начинаться поиск подходящих чисел.

Далее, какие десятичные числа в пятеричной системе оканчиваются на 21? Ответ – у которых остаток от деления на 25 равен 11 (убедитесь в этом!). Аналогично для числа 23 – у которых остаток от деления на 25 равен 13. Эти два признака должны быть объединены с помощью дизъюнкции.

Возможны и комбинации перечисленных условий и условия кратности или «некратности» чисел из обрабатываемого диапазона.

Дополнение

Возможны также задачи, в которых требуется исследовать все цифры десятичных чисел. Обсудим методику их решения.

В общем виде задача смысла задачи можно сформулировать так⁴: «Определить количество ...-значных натуральных чисел, у которых <условие, связанное со значениями цифр>».

Пример 1. Найти количество шестизначных натуральных чисел, у которых суммы трех их первых и трех последних цифр равны.

⁴ В условии могут фигурировать также «счастливые числа», «простые числа», «совершенные числа» и т. п.

Пример 2. Найти количество четырехзначных натуральных чисел, у которых суммы двух их первых и двух последних цифр отличаются не более, чем на 2.

Пример 3. Назовем натуральное шестизначное число N «счастливым», если сумма двух его первых цифр больше, чем сумма двух средних цифр, которая, в свою очередь, больше, чем сумма двух последних цифр. Найти количество таких чисел.

Пример 4. Назовем натуральное пятизначное число N «счастливым», если его цифры записаны в порядке неубывания. Найти количество таких чисел.

Общая схема программы решения задач указанного типа следующая:

1. Присваивание переменной-счетчику начального значения
2. **цикл** для каждого числа из заданного диапазона
Выделение цифр числа
если указанное в задаче условие соблюдается
Увеличить значение переменной-счетчика на 1
все
конец цикла
3. Вывод ответа

где переменная-счетчик – переменная, соответствующая искомому количеству.

Частная задача выделения отдельных цифр натурального числа была рассмотрена в п. 1.1.2.

В программе решения задачи примера 1 используем следующие переменные величины:

n – отдельное исследуемое шестизначное число;

цифра1, цифра2, ..., цифра6 – отдельные цифры этого числа;

кол – искомое значение количества чисел (переменная-счетчик).

Программа:

```
алг Задача_примера_1
нач цел n, цифра1, цифра2, цифра3, цифра4, цифра5, цифра6, кол
кол := 0
нц для n от 100000 до 999999 | Рассматриваем все 6-значные числа
| Выделяем каждую цифру (см. п 1.1.1)
цифра6 := mod(n, 10); n := div(n, 10)
цифра5 := mod(n, 10); n := div(n, 10)
цифра4 := mod(n, 10); n := div(n, 10)
цифра3 := mod(n, 10); n := div(n, 10)
цифра2 := mod(n, 10); n := div(n, 10)
цифра1 := mod(n, 10)
| Сравниваем суммы
если цифра1 + цифра2 + цифра3 = цифра4 + цифра5 + цифра6
то
кол := кол + 1
все
кц
вывод нс, кол
кон
```

Язык Python

```
kol = 0
for n in range(100000, 1000001):
    tsifra6 = n % 10; n = n//10
    tsifra5 = n % 10; n = n//10
    tsifra4 = n % 10; n = n//10
    tsifra3 = n % 10; n = n//10
```

```

    tsifra2 = n % 10; n = n//10
    tsifra1 = n % 10
    if tsifra1 + tsifra2 + tsifra3 == tsifra4 + tsifra5 + tsifra6:
        kol = kol + 1
print(kol)

```

В программе на языке Паскаль в теле оператора цикла с параметром величину, являющуюся параметром цикла, изменять нельзя, поэтому приходится использовать переменную, хранящую копию значения числа *n* (ее имя в приведенной ниже программе – *kopia_n*).

```

var n, tsifra1, tsifra2, ..., tsifra1, kopia_n, kol: longint;
BEGIN
    kol := 0;
    for n := 100000 to 999999 do
        begin
            kopia_n := n;
            tsifra6 := kopia_n mod 10; kopia_n := kopia_n div 10;
            tsifra5 := kopia_n mod 10; kopia_n := kopia_n div 10;
            ...
            tsifra1 := kopia_n mod 10;
            if tsifra1 + tsifra2 + tsifra3 =
                tsifra4 + tsifra5 + tsifra6 then
                kol := kol + 1
            end;
        writeln(kol)
    END.

```

Когда выделяемых цифр много, целесообразно предварительно записать их в массив. Например, для задачи с восьмизначными числами: «Найти количество восьмизначных натуральных чисел, у которых суммы четырех их первых и четырех последних цифр равны» – программа будет иметь вид:

```

алг
нач цел n, k, кол, цел таб цифры[1 : 8]
    кол := 0
    нц для n от 10000000 до 99999999
        | Записываем цифры числа n в массив
        нц для k от 1 до 8
            цифры[k] := mod(n, 10)
            n := div(n, 10)
        кц
        если цифры[1] + цифры[2] + цифры[3] + цифры[4] =
            цифры[5] + цифры[6] + цифры[7] + цифры[8]
        то
            кол := кол + 1
        все
    кц
    вывод нс, кол
кон

```

В ней использован массив из восьми элементов. Как отмечалось в *п. 1.1.6*, в результате заполнения этого массива в первом элементе массива записана младшая (последняя) цифра. В данной задаче это значения не имеет (сравниваются суммы цифр в двух «половинах» числа).

Язык Python

```

kol = 0
for n in range(10000000, 100000000):
    tsifri = [0 for i in range(8)]
    for k in range(8):

```

```

tsifri[k] = n % 10
n = n//10
if tsifri[0] + tsifri[1] + tsifri[2] + tsifri[3] == _
    tsifri[4] + tsifri[5] + tsifri[6] + tsifri[7]:
    kol = kol + 1
print(kol)

```

Вариант программы с методом append:

```

kol = 0
for n in range(100000000, 1000000000):
    tsifri = [] # Именно здесь!
    for k in range(8):
        tsifri.append(n % 10)
        n = n//10
    if tsifri[0] + tsifri[1] + tsifri[2] + tsifri[3] == _
        tsifri[4] + tsifri[5] + tsifri[6] + tsifri[7]:
        kol = kol + 1
print(kol)

```

Обратим внимание на важное обстоятельство. В задачах, аналогичных примеру 2, следует использовать абсолютную величину разности сумм, так как сумма первых двух цифр может быть как больше, так и меньше, чем сумма двух последних цифр:

```

если iabs((цифра1 + цифра2) - (цифра3 + цифра4)) <= 2
то
    ...

```

где функция школьного алгоритмического языка `iabs` возвращает модуль (абсолютное значение) своего аргумента целого типа.

Заметим также, что в данном случае при использовании функции `iabs` в условии указывается нестрогое неравенство (в других задачах, возможно, это и не так).

Обсудим решение примера 4.

Как определить, что все цифры некоторого числа n записаны в порядке неубывания? Проще всего это сделать, подсчитав количество цифр, не меньше «предыдущих», начиная со второй цифры. Если это количество равно 4, то число n – счастливое. В приведенной ниже программе использованы переменные величины-счетчики:

```

кол_не_меньше – количество цифр числа, не меньше «предыдущих»;
кол_счаст – искомое количество счастливых чисел.

алг Задача_примера_4
нач цел n, кол_счаст, цифра1, цифра2, цифра3, цифра4, цифра5,
    кол_не_меньше
    кол_счаст := 0
нц для n от 10000 до 99999
    | Выделяем цифры
    цифра5 := mod(n, 10); n := div(n, 10)
    цифра4 := mod(n, 10); n := div(n, 10)
    ...
    цифра1 := mod(n, 10)
    | Определяем значение переменной кол_не_меньше
    кол_не_меньше := 0 | Именно здесь!
    если цифра2 >= цифра1
        то
            кол_не_меньше := кол_не_меньше + 1
    все
    если цифра3 >= цифра2
        то

```

```

        кол_не_меньше := кол_не_меньше + 1
    все
    ...
    если цифра5 >= цифра4
    то
        кол_не_меньше := кол_не_меньше + 1
    все
    | Проверяем на "счастливость"
    если кол_не_меньше = 4
    то
        кол_счаст := кол_счаст + 1
    все
кц
вывод нс, кол_счаст
кон

```

Конечно, при записи цифр в массив лучше использовать оператор цикла:

```

алг Задача_примера_4
нач цел n, кол_счаст, кол_не_меньше, инд, k, цел таб цифры[1 : 5]
    кол_счаст := 0
    нц для n от 10000 до 99999
        | Записываем цифры числа n в массив
        нц для k от 5 до 1 шаг -1
            цифры[k] := mod(n, 10)
            n := div(n, 10)
        кц

        | Определяем значение переменной кол_не_меньше
        кол_не_меньше := 0
        нц для инд от 2 до 5
            | Начиная со 2-й цифры,
            | если цифры[инд] >= цифры[инд - 1] | сравниваем с предыдущей
            то
                кол_не_меньше := кол_не_меньше + 1
            все
        кц
        | Проверяем на "счастливость"
        если кол_не_меньше = 4
        то
            кол_счаст := кол_счаст + 1
        все
    кц
    вывод нс, кол_счаст
кон

```

Обратим внимание на то, что в приведенной программе цифры записываются в массив в порядке их следования в числе.

Более сложным вариантом задач рассматриваемого типа являются задачи, в которых используются особенности цифр, представленных в десятичной записи натурального числа.

Пример 5. Назовем натуральное число N ($10000_8 \leq N \leq 77777_8$) счастливым, если суммы двух первых и двух последних цифр его восьмеричной записи различаются не больше, чем на 2. Найти количество таких чисел.

Решение

Отличие решения в том, что следует получить цифры каждого из натуральных чисел в их восьмеричной записи. Такая задача отличается от задачи перевода в двоичную систему тем, что при выделении восьмеричных цифр нужно вместо числа 2 использовать 8 (основание новой системы). Кроме того, здесь важно определить диапазон проверяемых натуральных

чисел. Для рассматриваемой задачи – это числа от 4096 до 32767 ($10000_8 = 4096_{10}$, $77777_8 = 32767_{10}$).

Вся программа имеет вид:

```
алг Задача_примера_5
нач цел n, к, кол, цел таб цифры[1 : 5]
  кол := 0
  нц для n от 4096 до 32767
    | Переводим число в восьмеричную систему,
    | записывая восьмеричные цифры числа в массив
  нц для к от 5 до 1 шаг -1
    цифры[к] := mod(n, 8)
    n := div(n, 8)
  кц
  | Проверяем (абсолютную величину разности сумм цифр)
  если iabs(цифры[1] + цифры[2] - цифры[4] - цифры[5]) <= 2
    то
      кол := кол + 1
  все
кц
вывод нс, кол
кон
```

Язык Паскаль

```
Var n, kol: longint; k: byte; tsifri: array[1..5] of byte;
BEGIN
kol := 0;
for n := 4096 to 32767 do
  begin
    for k := 5 downto 1 do
      begin
        tsifri[k] := n mod 8;
        n := n div 8
      end;
    if abs(tsifri[1] + tsifri[2] - tsifri[4] - tsifri[5]) <= 2 then
      kol = kol + 1;
    end;
  end;
write(kol)
END.
```

Язык Python

```
tsifri = [0 for k in range(5)]
kol = 0
for n in range(4096, 32768):
  for k in range(4, -1, -1):
    tsifri[k] = n % 8
    n = n // 8
  if abs(tsifri[0] + tsifri[1] - tsifri[3] - tsifri[4]) <= 2:
    kol = kol + 1
print(kol)
```

В программе на языке Python можно не переводить восьмеричные числа 10000 и 77777 в десятичную систему «вручную», как это делалось при написании программ выше. Удобно использовать следующую особенность функции `int()`. Эта функция не только возвращает целую часть своего аргумента-числа и переводит строковое представление в целые числа, но и переводит недесятичные числа в их десятичную запись! Для такого перевода надо указать в функции `int()` два аргумента:

- 1) переводимое число (как строку – кавычках);
- 2) основание системы, в которой оно записано.

Например, в результате выполнения инструкции

```
print(int("A1 ", 16))
```

на экран будет выведено число 161 ($A1_{16} = 161_{10}$). Видно, что «цифры», большие 9, записываются латинскими буквами (от «A» до «Z»). Основание (второй аргумент) может быть больше либо равно 2 и меньше либо равно 36. Например, в результате выполнения инструкции

```
print(int("ZF1 ", 36))
```

на экран будет выведено число 45901.

Учет указанной особенности на экзамене позволяет сэкономить время выполнения задания.

Например, в последней программе заголовок инструкции `for` может быть оформлен так:

```
for n in range(int("10000 ", 8), int("100000 ", 8)):
```

...

Обсудим также вариант задачи, в которой используются особенности цифр, представленных в недесятичной записи натурального числа, но количество цифр в этой записи или/и основание недесятичной системы счисления заранее неизвестно.

Пример 6. «Определить количество натуральных чисел из интервала от ... до ..., в записи которых в системе счисления с основанием ..., суммы двух первых и двух последних цифр равны».

Решение

В таких программах нужно:

- 1) использовать оператор цикла с условием;
- 2) массив для хранения недесятичных цифр описать с запасом;
- 3) при записи цифр в массив вести подсчет их количества.

алг Задача_примера_6

нач **цел** n, основание, кол, **цел таб** цифры[1 : 100]

ввод основание

кол := 0

нц для n **от** ... **до** ...

кол_цифр := 0

нц пока n > 0

кол_цифр := кол_цифр + 1

цифры[кол_цифр] := mod(n, основание)

n := div(n, основание)

кц

если цифры[1] + цифры[2] = цифры[кол_цифр] + цифры[кол_цифр - 1]

то

кол := кол + 1

все

кц

вывод кол, кол

кон

Язык Паскаль

```
Var n, kol: longint; k: byte; tsifri: array[1..100] of byte;
```

```
BEGIN
```

```
readln(osnovanie);
```

```
kol := 0;
```

```

for n := ... to ... do
begin
kol_tsifr := 0;
while n > 0: do
begin
kol_tsifr := kol_tsifr + 1;
tsifri[kol_tsifr] := n mod osnovanie;
n := n div osnovanie
end;
if tsifri[1] + tsifri[2] = tsifri[kol_tsifr - 1] +
tsifri[kol_tsifr])
then
kol = kol + 1;
end;
write(kol)
END.

```

Язык Python

```

osnovanie = int(input())
tsifri = [0 for k in range(100)]
kol := 0
for n in range(..., ...):
kol_tsifr = 0
while n > 0:
kol_tsifr = kol_tsifr + 1
tsifri[kol_tsifr - 1] = n % osnovanie
n = n // osnovanie
if tsifri[0] + tsifri[1] ==
tsifri[kol_tsifr - 2] + tsifri[kol_tsifr - 1]:
kol = kol + 1
print(kol)

```

При использовании в программе на языке Python метода `append` обратиться к двум последним элементам списка (индексы которых в данном случае неизвестны) так: `tsifri[-2]` и `tsifri[-1]`.

В заключение заметим, что возможны также варианты всех рассмотренных выше типов задач, в которых требуется определить не количество чисел, удовлетворяющих условию, а другие характеристики их множества, упоминавшиеся выше (медиану и др.).

Задачи для самостоятельной работы

См. [3], файл `kege3.doc`

Литература

1. Демонстрационный вариант контрольных измерительных материалов единого государственного экзамена 2021 года по информатике и ИКТ. <https://fipi.ru/ege/demoversii-specifikacii-kodifikatory#!/tab/151883967-5> (Демоверсии, спецификации, кодификаторы 2021. Информатика и ИКТ).
2. Спецификация контрольных измерительных материалов для проведения в 2021 году единого государственного экзамена по информатике и ИКТ. <https://fipi.ru/ege/demoversii-specifikacii-kodifikatory#!/tab/151883967-5> (Демоверсии, спецификации, кодификаторы 2021. Информатика и ИКТ).
3. <http://kpolyakov.spb.ru/school/ege/retro.htm>

Книга для подготовки к ЕГЭ по информатике

Издательство ДМК Пресс предлагает книгу Д.М. Златопольского «Подготовка к ЕГЭ по информатике в компьютерной форме».

Златопольский Д. М. Подготовка к ЕГЭ по информатике в компьютерной форме 	Книга предназначена для самостоятельной подготовки учащихся к единому государственному экзамену по информатике и ИКТ, который начиная с 2021 года будет проходить в компьютерной форме. Согласно демонстрационному варианту ЕГЭ 2021 года, в содержании экзамена будет существенно увеличено количество заданий, связанных с алгоритмизацией и программированием. Таким задачам в книге уделяется особое внимание. Обсуждаются и методики выполнения ряда других заданий, представленных в демонстрационном варианте. В приложениях приведены вспомогательные материалы, связанные с заданиями ЕГЭ. Кроме учащихся старших классов, готовящихся к сдаче экзамена, книгу могут использовать преподаватели информатики, а также студенты и школьники, желающие углубить общие знания по информатике и ИКТ вне связи с ЕГЭ.
---	---

Подробнее — см. <https://dmkpress.com/catalog/computer/handbooks/978-5-97060-896-8/>

Книга «400 вопросов по информатике на логику и смекалку»

В издательстве ДМК Пресс вышла книга Д.М. Златопольского «400 вопросов по информатике на логику и смекалку» (<https://dmkpress.com/catalog/computer/handbooks/978-5-97060-898-2/>).

Златопольский Д. М. 400 вопросов по информатике на логику и смекалку 	В книге приведены более 400 вопросов по информатике, которые можно предложить учащимся при проведении классных и школьных конкурсов «Что? Где? Когда?» и «Брейн-ринг» и других внеклассных мероприятий. Ряд вопросов может быть использован и на уроках. Вопросы охватывают многие темы школьного курса информатики, имеют разный уровень сложности. В приложениях представлены другие вопросы и задания, связанные с интеллектуальными соревнованиями. Ко всем вопросам даны ответы. Книга предназначена педагогам, но может быть полезна также родителям учащихся и, конечно, самим учащимся, которые могут проверить свои знания, логику и смекалку.
---	---

Ответы, решения, разъяснения
к заданиям, опубликованным в журнале «Мир информатики»,
выпуск № 53 (февраль 2020 г.)

Задача «Шутники и серьезные. Задача 2»

Напомним условие: «Школьники одного из классов для игры разбились на две “партии”: на “серьезных”, отвечающих правильно на любой вопрос, и на “шутников”, дающих на любой вопрос только неправильные ответы.

Однажды учитель спросил каждого из пяти учеников этого класса: “Сколько среди вас «серьезных»?” — и получил следующие ответы: “Ни одного”, “Один”, “Два”, “Три”, “Четыре”.

Сколько же ребят из “партии «серьезных»” было в этой группе?»

Решение

Ученик, давший ответы: “Ни одного”, — “шутник”, так как “серьезный” обязательно назвал бы число, равное или больше 1. Так что ему верить нельзя, и в группе должен быть обязательно по крайней мере один “серьезный”.

Предположим, что ученик, назвавший число 1, — тоже лжец. Тогда в ответах его товарищей должно было появиться два раза число 2 (если “серьезных” — два) или три раза число 3 (если их — три). Но этого не произошло. Следовательно, среди ребят — один “серьезный”, а именно тот, который назвал число 1.

Кроссворд

Приведем ответы так, как их оформила Виктория Голубева, ученица 6-го (!) класса Сапожковской средней школы имени Героя России Тучина Алексея Ивановича, Рязанская обл. (учитель **Задубровский В.Ф.**):

По горизонтали:	По вертикали:
5. Альфа	1. Радио
7. Плата	2. Лье
8. Бит	3. Шаг
9. Архив	4. Хакер
10. Стек	6. Абонент
12. Директива	7. Принтер
15. Тор	11. График
16. Файл	12. Драйвер
17. Граф	13. Материк
20. Стратегия	14. Данные
23. Обь	18. Регистр
24. Игрек	19. Теорема
25. Код	21. Абзац
28. Автор	22. Адрес
29. Объём	26. Стек
30. Каталог	27. Круг

Задача «Трое с одной улицы»

Напомним условие: «Кондратенко, Давыденко и Федоренко живут на одной улице. Один из них — столяр, другой — маляр, третий — сантехник. Недавно Давыденко хотел попросить столяра, который был его старым знакомым, сделать кое-что для своей квартиры, но ему сказали, что столяр работает в доме сантехника. Известно также, что Федоренко никогда не слышал о Давыденко. У кого какая профессия?

Задача решается табличным методом.

Ответ: Кондратенко — столяр, Давыденко — маляр, Федоренко — сантехник.

Обратим внимание на то, что в ряде ответов фигурировали другие фамилии и профессия «водопроводчик». Это говорить о том, что ... ☹.

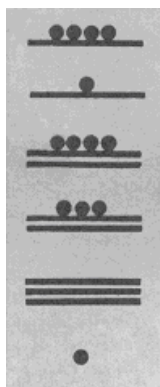
Задания, предложенные для самостоятельной работы в статье «Старинные системы счисления»

1. Записанное по вавилонской системе число



соответствует числу $24 \times 60^3 + 2 \times 60^2 + 32 \times 60 + 42 = 5193162$.

2. Записанное по системе майя число:



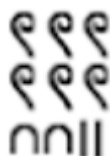
соответствует числу $9 \times 20^3 \times 360 + 6 \times 20^2 \times 360 + 14 \times 20 \times 360 + 13 \times 20 \times 18 + 15 \times 20 + 1 = 26889781$.

3. Записанные по египетской системе числа:



соответствуют числам 365 и 257.

4. Сумма указанных чисел (622), представленная в египетской системе счисления:



5. $999 = \text{CMXCIX}$, $1987 = \text{MCMLXXXVII}$

6. Римские числа CDXLIV и MCMLXXIV соответствуют арабским числам 444 и 1974.

7. Соответствие заданных десятичных чисел и чисел в славянской алфавитной системе следующие:

$$\begin{aligned}\overline{\text{Ѡ}}\overline{\text{Ѣ}} &= 15 \\ \overline{\text{П}}\overline{\text{Г}} &= 83 \\ \overline{\text{Ф}}\overline{\text{С}}\overline{\text{І}} &= 516 \\ \overline{\text{Ф}}\overline{\text{Н}}\overline{\text{З}} &= 557 \\ \overline{\text{Д}}\overline{\text{В}}\overline{\text{Н}}\overline{\text{І}} &= 4818 \\ \overline{\text{Ѧ}}\overline{\text{Х}}\overline{\text{М}}\overline{\text{В}} &= 5642 \\ \textcircled{\text{З}}\overline{\text{С}}\overline{\text{М}}\overline{\text{Л}} &= 70241\end{aligned}$$

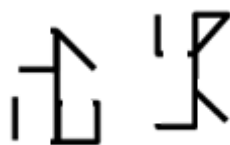
8. Соответствие между заданными десятичными числами и числами, записанными цистерцианскими цифрами, следующие:

а) 43 =

б) 285 =

в) 9999 =

9. Числам, записанным цистерцианскими цифрами:



соответствуют десятичные числа 6723 и 1465.

Задачи с песочными часами. Часть 4

Ответы

1.

Первые часы (7 мин)	1	3	5	н	к
Вторые часы (8 мин)	2	4	6		

2.

Первые часы (7 мин)	1	3	5	н	к
Вторые часы (12 мин)	2	4			

3.

Первые часы (5 мин)	1	3	5	н	к
Вторые часы (6 мин)	2	4	6		

Условное форматирование

Л.Н. Медведев,
Москва

При работе с электронной таблицей Microsoft Excel часто встречаются ситуации, когда необходимо принять какие-то решения или дать оценку выполнению каких-то действий пользователя. Например, при решении квадратного уравнения может оказаться, что корней у данного уравнения нет. Можно выдать результат на экран так, чтобы привлечь к нему внимание и остановить дальнейшие действия пользователя. На этом же принципе может быть основан контроль ввода. Если пользователь должен ввести число, лежащее в заданном интервале, а вводит значение, не соответствующее этому условию, можно вывести замечание в такой форме, чтобы это сразу бросалось в глаза. Вот на последнем примере я и остановлюсь подробнее.

На *рис. 1* показан фрагмент листа таблицы, в которой предлагается ввести в ячейку A2 число и контролируется вводимое значение.

	A	B	C	D	E
1	Введите в ячейку A2 число в интервале от 0 до 1				
2	0				
3					
4					

Рис. 1

Если число введено правильно (0 или 1), то ничего не происходит. А если ввести число, меньшее 0 (см. *рис. 2*)?

	A	B	C	D
1	Введите в ячейку A2 число в интервале от 0 до 1			
2	-1		МАЛО	
3				
4				

Рис. 2

Ячейка A2 изменила формат, кроме того, в ячейке C2 появился комментарий на цветном фоне. При вводе числа больше 1 получим такую реакцию (*рис. 3*):

	A	B	C	D
1	Введите в ячейку A2 число в интервале от 0 до 1			
2	3		МНОГО	
3				
4				

Рис. 3

Снова — изменения. Как же это сделано? На *рис. 4* показана формула, работающая в ячейке C2 и скрытый⁵ текст в ячейках E2 и E3, который выводится в качестве комментария.

⁵ Оформленный белым цветом. — Ред.

	A	B	C	D	E
1	Введите в ячейку A2 число в интервале от 0 до 1				
2			=ЕСЛИ(A2>1;E2;ЕСЛИ(A2<0;E3;""))		МНОГО
3					МАЛО
4					

Рис. 4

Надеюсь, что анализ формулы в ячейке C2 не вызовет у вас затруднений⁶. А что же с форматом ячеек A2 и C2? Там использовано так называемое *условное форматирование*. Чтобы его задать, надо выделить соответствующую ячейку, в главном меню программы выбрать пункт **Формат**, а в нем подпункт **Условное форматирование** и в появившемся диалоговом окне (см. рис. 5) задать нужные форматы. Можно изменять: формат шрифта (тип, размер, начертание, цвет), фон ячейки, виды и цвета рамок со всех четырех сторон ячейки.

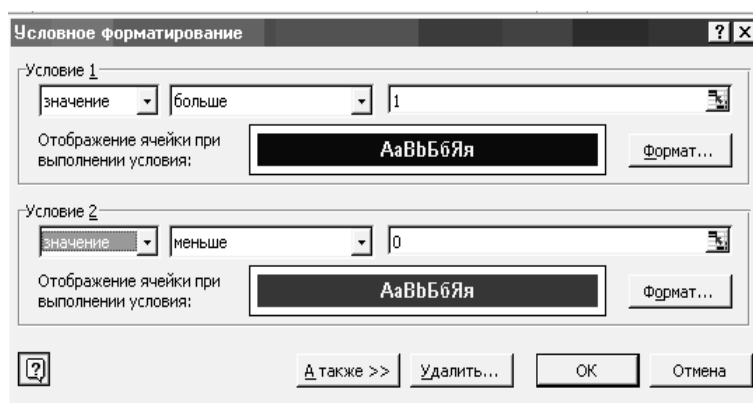


Рис. 5

В окне **Условное форматирование** можно задать до трех условий, на соответствие которым будет проверяться значение, получающееся⁷ в контролируемой ячейке. Каждое из этих условий может выполнять одну из операций сравнения из списка (он находится там, где на рис. 5 стоят слова больше и меньше). Таких операций восемь:

между
 вне
 равно
 не равно
 больше
 меньше
 больше или равно
 меньше или равно

Следует учитывать, что условное форматирование имеет приоритет перед обычным форматированием ячейки, то есть если вы задали для ячейки с помощью меню **Формат | Ячейки** синий фон и желтый шрифт, а согласно условному форматированию должен быть желтый фон и черный шрифт — действовать будет последний вариант.

Задания для самостоятельной работы

1. Оформите лист для решения квадратного уравнения, коэффициенты a , b , c которого вводятся в отдельные ячейки. При вводе нулевого значения коэффициента a в той же ячейке должны быть выведено сообщение о недопустимости такого значения.

⁶ В ней использована так называемая «вложенная функция ЕСЛИ». — Ред.

⁷ Это может быть как значение, вводимое с клавиатуры, так и полученное в результате расчета по формуле. — Ред.

2. Создайте “шахматное поле”, которое при вводе значения 0 в ячейку A1 оказывается пустым (без черно-белых ячеек, как на *рис. 6*), при вводе 1 — таким, как на *рис. 7*, а при вводе 2 — как на *рис. 8*.

	A	B	C	D	E	F	G	H	I	J	K
1	0										
2											
3											
4											
5											
6											
7											
8											
9											
10											

Рис. 6

	A	B	C	D	E	F	G	H	I	J	K
1	1										
2											
3											
4											
5											
6											
7											
8											
9											
10											

Рис. 7

	A	B	C	D	E	F	G	H	I	J	K
1	2										
2											
3											
4											
5											
6											
7											
8											
9											
10											

Рис. 8

3. Подготовьте лист электронной таблицы таким образом, чтобы флаги на *рис. 10–12* соответствовали названию страны, вводимому в ячейку A1 (цвета национальных флагов приведенных стран узнайте в справочной литературе или у учителя географии).

	A	B	C	D	E	F
1						
2						
3						
4						

Рис. 9

	A	B	C	D	E	F
1	Россия					
2						
3						
4						

Рис. 10

	A	B	C	D	E	F
1	Франция					
2						
3						
4						

Рис. 11

	A	B	C	D	E	F
1	Италия					
2						
3						
4						
5						

Рис. 12

4. На рис. 13 и 14 приведен пример того, как можно создать логическую схему с помощью условного форматирования. При этом:

1) текст в ячейках с результатами логических операций должен быть получен по формулам с использованием логических функций;

2) значение ЛОЖЬ в только что упомянутых ячейках должно выводиться синим цветом на желтом фоне, значение ИСТИНА — бирюзовым цветом на синем фоне.

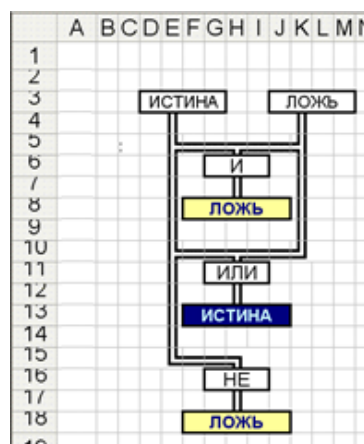


Рис. 13



Рис. 14

Вместо изображенных на рисунках двойных линий используйте утолщенные границы ячеек.

Исходные логические значения и результаты операций (ЛОЖЬ, ИСТИНА) и логические связки И, ИЛИ и НЕ вводятся в так называемые «объединенные» ячейки (после выделения объединяемых ячеек с помощью правой кнопки мыши вызывается контекстное меню, в нем выбирается пункт **Формат ячеек**, вкладка **Выравнивание**). Значения в ячейках с результатами должны определяться законам логики.

Листы с решениями присылайте в редакцию (можно выполнять не все задания). Фамилии всех приславших будут опубликованы, а авторы лучших ответов будут награждены дипломами. Срок представления ответов — 30 сентября 2021 года.

Внимание! Конкурс!

Итоги конкурса № 37 «Старинные системы счисления»

Победителями конкурса признаны и будут награждены дипломами:

— Авраменко Максим и Бородин Виктор, г. Москва, школа № 1530 «Школа Ломоносова», учитель **Каменецкая Т.С.**;

— Барсков Максим, Камышанская Анастасия, Князев Святослав, Комолкин Роман, Лефтеров Владислав, Никитина Ульяна, Печникова Арина и Смирнова Полина, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Волков Александр, Московская обл., г. Мытищи, ЧУОШ «Логос М», учитель **Елистратова А.А.**;

— Киселев Станислав, Владимирская обл., г. Струнино, школа № 11, учитель **Волкова Т.П.**;

— Листопад Глеб, Владимирская обл., г. Александров, школа № 3, учитель **Солодова Е.Н.**;

— Фалин Вячеслав, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Фирсова Ирина, Волгоградская обл., г. Фролово, школа № 1 имени А.М. Горького, учитель **Фирсова М.Н.**

Поздравляем всех!

Книжная полка

Сборник задач по программированию

Если вы хотите стать программистом, то должны знать, что одним из условий этого является большой опыт решение задач по программированию. Решайте задачи, абстрактные и содержательные, на 5 минут и на день работы. Все это обязательно вам пригодится. Чем больше опыт решения задач, тем больше вероятность, что уже решенная задача или аналогичная ей встретится вам в будущем. В приобретении такого опыта вам поможет книга «1400 задач по программированию», выпущенная издательством ДМК Пресс.



В книге приведено более 1400 задач по программированию. Задачи имеют разный уровень сложности и охватывают большой круг тем. Они не привязаны к конкретному языку программирования и могут быть решены в программе на любом языке (Паскаль, Python, Си, школьном алгоритмическом языке и т. д.).

Издание будет полезно как читателям, изучающим программирование самостоятельно, так и преподавателям школ, колледжей и вузов. В последнем случае задачи могут быть использованы при объяснении нового материала, для организации самостоятельной работы учащихся и проведения контрольных мероприятий. Ряд задач предназначен для внутришкольных олимпиад, для использования в кружках по программированию и т. п.

Наше пожелание выпускникам:

