

“Мир информатики”

Журнал для тех, у кого информатика – любимый школьный предмет

Выпуск № 3, август 2016 г.

01100010001010100100111110010001|1001001000101001001001011

Школа программирования

*Когда человек хочет передвинуть гору,
он начинает с того, что убирает маленькие камни.*
Восточная мудрость

Типовые задачи обработки последовательности чисел

В данной статье рассмотрены методы решения 12 типовых задач, которые встречаются при обработке последовательности чисел (нахождение их суммы, максимального значения и др.). При описании будем применять школьный алгоритмический язык (система программирования КуМир). Русский синтаксис этого языка делает его максимально понятным, и вы легко сможете разработать аналогичные программы на языке программирования, который изучаете.

Использованы следующие основные величины:

n — общее количество чисел в последовательности (оно может быть заранее задано в программе или вводиться в ходе её выполнения);

a — очередное обрабатываемое число последовательности.

Смысл остальных величин можно легко определить по их именам.

Так как количество обрабатываемых чисел n известно, в программах использован оператор цикла с параметром (см. [1]). Заметим также, что все рассмотренные задачи указаны в Кодификаторе элементов содержания и требований к уровню подготовки выпускников ЕГЭ по информатике [2] и используются в заданиях экзамена.

1. Суммирование всех чисел последовательности

Пусть требуется найти сумму всех чисел последовательности: 5, 3, 1, 5, 12, 6, 3, 3, 8, 12.
Составим таблицу:

Номер числа	Число a	Сумма уже рассмотренных чисел, сум
1	5	5
2	3	8 (5 + 3)
3	1	9 (8 + 1)
...		
i	a	?
...		
10	10	

Как определить сумму всех уже рассмотренных чисел после ввода некоторого очередного i -го числа a , если для хранения этой суммы использовать величину $сум$? Ответ — надо к “старому” значению суммы прибавить очередное число:

```
сум := 0
нц для  $i$  от 1 до  $n$ 
  |Ввод очередного числа  $a$ 
  ...
  сум := сум +  $a$ 
кц
|Вывод результата или использование его в расчётах
...
```

Пример. Известна масса каждого предмета, загружаемого в автомобиль. Определить общую массу груза.

Решение

```
...
вывод нс, "Укажите количество предметов"
ввод  $n$ 
сум := 0
нц для  $i$  от 1 до  $n$ 
  вывод нс, "Введите массу",  $i$ , "-го предмета"
  ввод  $a$ 
  сум := сум +  $a$ 
кц
вывод нс, "Общая масса всех предметов равна", сум
```

Примечание. В языках программирования Бейсик, Паскаль, Си и ряде других оператор $сум := 0$ не является обязательным. Вместе с тем, “правилом хорошего тона” является начальное присваивание величине $сум$ нулевого значения (a в отдельных задачах и языках программирования это является обязательным).

2. Нахождение произведения всех чисел последовательности

Здесь начальное присваивание $произв := 1$ является обязательным (почему — установите самостоятельно):

```
произв := 1
нц для  $i$  от 1 до  $n$ 
  |Ввод очередного числа  $a$ 
  ...
  произв := произв *  $a$ 
кц
```

Следует иметь в виду, что значение произведения не должно выходить за пределы, допускаемые для использованного для величины $произв$ типа данных.

3. Суммирование тех чисел последовательности, которые удовлетворяют некоторому условию

Отличие данной задачи от задачи 1 в том, что учитывать в сумме $сум$ надо не все числа последовательности. В [1] указывалось, что если в программе некоторые действия выполняются только при определённом условии, то должен быть использован так называемый “неполный условный оператор”:

```
сум := 0
нц для  $i$  от 1 до  $n$ 
  |Ввод очередного числа  $a$ 
  ...
```

```

если <условие> | Только в этом случае
  то
    | происходит учёт значения a в сумме
    сум := сум + a
все
кц
| вывод результата или использование его в расчётах
...

```

Примечание. Здесь условие в условном операторе (команде **если**) может определяться значением числа *a* или его порядковым номером *i*.

4. Подсчёт количества тех чисел последовательности, которые удовлетворяют некоторому условию

Пусть требуется найти количество чётных чисел последовательности: 5, 2, 7, 8, 3, 1, 5, 12, 6, 3, 3, 8, 12.

Составим таблицу:

Номер числа	Число <i>a</i>	Количество чётных чисел среди уже рассмотренных, <i>кол</i>
1	5	0
2	2	1 (0 + 1)
3	7	1
4	8	2 (1 + 1)
...	...	
<i>i</i>	12	?
...		

Как определить количество чётных чисел среди уже рассмотренных чисел после ввода некоторого *i*-го четного числа *a*, если для хранения такого количества использовать величину *кол*? Ответ — надо к “старому” значению количества прибавить 1.

В общем случае соответствующий фрагмент программы решения задачи:

```

кол := 0
нц для i от 1 до n
  | Ввод очередного числа a
  ...
  если <условие>
    то
      кол := кол + 1
  все
кц
| Вывод результата или использование его в расчётах
...

```

Примечание. В данном случае условие в условном операторе (команде **если**) определяется значением числа *a*. Если это условие зависит от порядкового номера *i*, то задача может быть решена без использования оператора цикла (убедитесь в этом!).

5. Определение среднего арифметического тех чисел последовательности, которые удовлетворяют некоторому условию

Для нахождения среднего арифметического чисел последовательности, которые удовлетворяют некоторому условию, надо знать сумму и количество таких чисел. Решать соответствующие задачи мы уже умеем. Поэтому можем так оформить фрагмент программы:

```

сум := 0
кол := 0
нц для i от 1 до n
  |Ввод очередного числа a
  ...
  если <условие>
    то
      сум := сум + a
      кол := кол + 1
    все
кц
|Подсчёт и вывод результата
среднее_арифметическое := сум/кол
вывод нс, "Среднее арифметическое: ", среднее_арифметическое

```

Обращаем внимание на то, что многократно определять значение *среднее_арифметическое* в “теле” условного оператора необходимости нет. Это можно сделать один раз после окончания работы оператора цикла. Обратим также внимание на то, что указанная величина должна быть описана как величина вещественного типа даже в случаях, когда *сум* — целое число.

Однако может оказаться, что чисел, удовлетворяющих заданному условию, в последовательности не окажется. В этом случае при расчёте будет иметь место деление на ноль, что недопустимо. Значит расчёт величины *среднее_арифметическое* должен проводиться не безусловно. При этом заметим, что оформление завершающей части программы в виде:

```

...
|Подсчёт и вывод результата
если кол > 0
  то
    среднее_арифметическое := сум/кол
  все
вывод нс, "Среднее арифметическое: ", среднее_арифметическое

```

в случае, когда чисел, удовлетворяющих заданному условию, в последовательности не окажется, даст неправильный результат — будет выведен ответ, равный 0.

Правильное оформление:

```

...
|Подсчёт и вывод результата
если кол > 0
  то
    среднее_арифметическое := сум/кол
    вывод нс, "Среднее арифметическое: ", среднее_арифметическое
  иначе
    вывод нс, "Чисел, удовлетворяющих условию, нет"
  все

```

6. Определение порядкового номера некоторого значения в заданной последовательности

Фрагмент программы, в котором *значение* — число, порядковый номер которого (*номер_значения*) ищется, следующий:

```

нц для i от 1 до n
  |Ввод очередного числа a
  ...
  |Сравнение его с искомым значением
  если a = значение

```

```

    то
        номер_значения := i
    все
кц

```

Нетрудно заметить, что при таком оформлении, в случае если числа *значение* в массиве нет, результат будет неправильным или появится сообщение об ошибке. Чтобы учесть в программе возможность такого случая, необходимо изменить фрагмент следующим образом:

```

номер_значения = 0
...
| Вывод результата
если номер_значения <> 0
    то
        вывод нс, "Номер этого значения: ", номер_значения
    иначе
        вывод нс, "Такого числа нет"
все

```

Возникает вопрос: какой из порядковых номеров будет найден по приведенному алгоритму, если в последовательности окажутся несколько чисел с искомым значением?

Примечание. Несколько изменив приведенный фрагмент, с его помощью можно получить ответ на вопрос о том, имеется ли в заданной последовательности некоторое число (см. также дополнение в конце статьи).

7. Определение максимального значения в последовательности чисел

Алгоритм решения этой задачи аналогичен алгоритму действий человека, который определяет максимальное значение в некоторой последовательности:

4, 2, 9, 6, 3, 16, 10, 2, 7

— сначала он запоминает первое число, а затем рассматривает второе число. Если оно больше того числа, которое помнил, то запоминает новое число и переходит к следующему, в противном случае просто переходит к следующему, третьему, числу и делает то же самое. Так и в нашей программе сначала следует задать первое число и принять его в качестве максимального (его имя в программе — *максимальное*). Затем задаются остальные числа *a* и каждое из них сравнивается со значением *максимальное*. Если $a > \text{максимальное}$, то в качестве нового значения *максимальное* принимается значение числа *a*.

Соответствующий фрагмент программы:

```

| Ввод первого числа a
...
максимальное := a
нц для i от 2 до n
    | Ввод остальных чисел a последовательности
    ...
    если a > максимальное
        то
            максимальное := a
        все
кц
| Вывод результата или использование его в расчётах
...

```

Примечание. Ясно, что при незначительном изменении приведённого фрагмента можно найти минимальное число последовательности.

Обсудим вопрос — можно ли не вводить отдельно первое число последовательности, а вводить и обрабатывать *все* её числа в теле оператора цикла:

```
нц для i от 1 до n
  |Ввод очередного числа a последовательности
  ...
  если a > максимальное
    то
      максимальное := a
  все
кц
|Вывод результата или использование его в расчётах
...
```

На первый взгляд, — да, можно. Однако это не так, точнее, так можно оформлять программу не во всех случаях. Например, если стоит задача определения максимальной температуры в первой декаде января и вводимые значения равны $-12, -10, -9, -6, -10, -13, -12, -10, -9, -9$, то в приведённом варианте ответом будет ошибочное максимальное значение 0.

Можно сделать вывод о том, что приведённый краткий вариант программы может быть применен, если известно, что в обрабатываемой последовательности чисел не все значения отрицательны.

В общем случае краткий вариант программы может быть использован, если известно минимальное число обрабатываемой последовательности:

```
максимальное := минимальное
нц для i от 1 до n
  |Ввод очередного числа a последовательности
  ...
  если a > максимальное
    то
      максимальное := a
  все
кц
```

где *минимальное* — минимальное число последовательности.

Примечание. Ясно, что последние рассуждения применимы и к задаче нахождения минимального числа последовательности (естественно, с учётом того, что ищется не максимум, а минимум).

8. Определение порядкового номера максимального значения в последовательности чисел

Здесь также алгоритм решения задачи аналогичен алгоритму действий человека, определяющего номер максимального значения в некоторой последовательности:

4, 2, 9, 6, 3, 16, 10, 2, 7

— сначала он запоминает первое число и число 1, а затем рассматривает второе число. Если оно больше того числа, которое помнил, то запоминает новое число и число 2 и переходит к следующему, в противном случае просто переходит к следующему, третьему, числу и делает то же самое.

В программе сначала следует задать первое число и принять его в качестве максимального (максимальное), а искомый порядковый номер (*номер_максимального*) принять равным 1. Затем задаются остальные числа a и каждое из них сравнивается со значением *максимальное*. Если $a > \text{максимальное}$, то в качестве нового значения величины *максимальное* принимается значение числа a , а в качестве нового значения величины *номер_максимального* — номер встреченного числа i .

Соответствующий фрагмент программы:

```
|Ввод первого числа а последовательности
...
максимальное := а
номер_максимального := 1
нц для i от 2 до n
  |Ввод остальных чисел а последовательности
  ...
  если а > максимальное
    то
      максимальное := а
      номер_максимального := i
  все
кц
|Вывод результата или использование его в расчётах
...
```

Если минимальное число обрабатываемой последовательности известно, то, как и в предыдущей задаче, фрагмент может быть оформлен короче:

```
максимальное := минимальное - 1
нц для i от 1 до n
  |Ввод очередного числа а последовательности
  ...
  если а > максимальное
    то
      максимальное := а
      номер_максимального := i
  все
кц
```

где *минимальное* — минимальное число последовательности.

Примечание. Задача определения порядкового номера минимального значения в последовательности чисел решается аналогично.

9. Определение максимального значения тех чисел последовательности, которые удовлетворяют некоторому условию

Здесь возможны два случая:

1) известно, что числа, удовлетворяющие заданному условию, в обрабатываемой последовательности имеются;

2) чисел, удовлетворяющих заданному условию, в обрабатываемой последовательности может не быть.

В первом случае задача решается следующим образом:

```
максимальное := X
нц для i от 1 до n
  |Ввод очередного числа а последовательности
  ...
  если <условие>
    то |Встретилось число, удовлетворяющее заданному условию
      |сравниваем его с величиной максимальное
      если а > максимальное
        то
          максимальное := а
      все
    все
кц
```

где X — число, о котором заведомо известно, что оно не превышает минимальное из чисел, для которых определяется максимальное значение.

Если же такое значение заранее неизвестно¹, задача несколько усложняется. В приведённом фрагменте программы её решения используется величина логического типа *впервые*², определяющая, впервые ли встретилось в последовательности число, удовлетворяющее заданному условию. Остальные особенности работы программы описаны в комментариях.

```
впервые := да | Искомое число будет первым
нц для i от 1 до n
  | Ввод очередного числа a последовательности
  ...
  если <условие>
    то | Встретилось число, удовлетворяющее заданному условию
      если впервые | если это произошло впервые
        то
          | Принимаем его в качестве значения максимальное
          максимальное := a
          впервые := нет | Следующие такие числа уже будут
                          | встречаться не впервые
        иначе | Если не впервые
          | сравниваем число с величиной максимальное
          если a > максимальное
            то
              максимальное := a
          все
        все
      все
    кц
```

Если же допускается, что чисел, удовлетворяющих заданному условию, в обрабатываемой последовательности может не быть, то для решения рассматриваемой задачи может быть использован последний приведённый фрагмент программы:

```
впервые := да
нц для i от 1 до n
  | Ввод очередного числа последовательности a
  ...
кц
```

— но вывод ответа при этом должен быть уточнен:

```
если впервые
  то
    вывод нс, "Чисел, удовлетворяющих условию, "
    вывод "в последовательности нет"
  иначе
    вывод нс, "Искомое максимальное значение: ", максимальное
  все
```

10. Нахождение второго по величине максимального числа последовательности

Данная задача допускает два толкования. Если рассматривать, например, последовательность чисел 5 10 22 6 22 20 6 12, то каким должен быть ответ?

Под “вторым по величине максимальным числом”, или, короче, “вторым максимумом”, можно понимать:

¹ Вопрос о том, можно ли всегда найти такое значение, исследуйте самостоятельно.

² В программах на языке Бейсик и Си вместо величины логического типа следует использовать величину целого типа, принимающую, например, значения 0 и 1.

- 1) число, которое стояло бы на предпоследнем месте, если бы все числа последовательности были отсортированы по неубыванию. При таком толковании — 22;
- 2) число, *больше* которого только максимальное число последовательности. В этом случае ответ — 20.

Если в последовательности только один максимальный элемент (все остальные меньше), то оба толкования совпадают, и искомые значения будут одними и теми же, в противном случае — нет.

Обсудим оба варианта задачи. В каждом из них будем использовать две переменные:

- 1) *максимум1* — максимальный элемент последовательности (первый максимум);
- 2) *максимум2* — второй максимум (искомое значение).

10.1. Поиск элемента, который стоял бы на предпоследнем месте, если бы числа последовательности были отсортированы по неубыванию

Сначала рассмотрим вариант, в котором диапазон значений элементов последовательности известен.

В качестве начальных значений величин *максимум1* и *максимум2* (см. выше) принимаем число, которое заведомо меньше нижней границы диапазона значений элементов чисел последовательности (например, при диапазоне от -1000 до 1000 — число -1001):

```
максимум1 := -1001
максимум2 := -1001
```

Строго говоря, значение *максимум2* можно не задавать, что будет показано далее.

Затем вводим и рассматриваем все числа, сравнивая их сначала со значением *максимум1*, а затем (при необходимости) — и со значением *максимум2*:

```
нц для i от 1 до n
  ввод a
  если a > максимум1
    |Встретилось число, большее максимум1
    то
      |Бывший первый максимум станет вторым
      максимум2 := максимум1
      |Первым максимумом станет встреченный элемент
      максимум1 := a
      |Внимание - именно в таком порядке!
  иначе
    |Очередное число не больше максимум1
    |Сравниваем его со значением максимум2
    если a > максимум2
      |Встретилось число, большее максимум2
      то
        |Принимаем его в качестве нового значения максимум2
        максимум2 := a
        |Значение максимум1 не меняется
  все
кц
```

Нетрудно убедиться, что уже после рассмотрения первого числа произойдет “переприсваивание”:

```
максимум2 := максимум1
```

— то есть начальное значение величины *максимум2*, равное -1001, можно не задавать.

Если же диапазон значений элементов последовательности неизвестен, то предварительно нужно определить начальные значения величин *максимум1* и *максимум2*,

сравнив первый и второй элементы последовательности a_1 и a_2 (естественно, предварительно задав их):

```
если a1 > a2
  то
    максимум1 := a1
    максимум2 := a2
  иначе
    максимум1 := a2
    максимум2 := a1
все
```

или короче:

```
максимум1 := a1
максимум2 := a2
если a2 > a1
  то
    максимум1 := a2
    максимум2 := a1
все
```

Затем рассматриваем остальные числа, как и ранее, сравнивая их сначала со значением *максимум1*, а затем (при необходимости) — и со значением *максимум2*:

```
нц для i от 3 до n
  ввод a
  если a > максимум1
    ...
```

10.2. Нахождения элемента последовательности, больше которого только максимальный

Примем, что диапазон значений элементов последовательности известен.

В качестве начального значения величины *максимум1*³ принимаем число, которое заведомо меньше нижней границы диапазона значений элементов массива (например, при диапазоне от -1000 до 1000 — число -1001):

```
максимум1 := -1001
```

Рассматриваем все числа последовательности и проверяем каждое из них сначала на первый, а затем на второй максимум:

```
нц для i от 1 до n
  ввод a
  если a > максимум1
    |Встретилось число, большее максимум1
    то
      |Бывший первый максимум станет вторым
      максимум2 := максимум1
      |Первым максимумом станет встреченное число
      максимум1 := a
    иначе
      |Очередное число не больше максимум1
      |В этом случае сравниваем его со значением максимум2
      если a < максимум1 и a > максимум2
        |Только в этом случае!
        то
          |Встретилось число, меньшее максимум1 и большее максимум2
          |Принимаем его в качестве нового значения максимум2
          максимум2 := a
```

³ Для величины *максимум2* начальное значение, как и для первого варианта, можно не задавать.

```

        |Значение максимум1 не меняется
    все
кц
Фрагмент программы, относящийся к выводу ответа, оформляется так:
если максимум2 = -1001
    то |Второй максимум не встретился
        вывод нс, "В последовательности нет такого значения"
    иначе
        вывод нс, "Второй максимум равен ", максимум2
все

```

Примечание. Задача нахождения второго минимума решается аналогично.

11. Нахождение количества максимальных элементов последовательности

Понятно, что задача может быть решена, если все числа последовательности вводить дважды:

- 1) первый раз найти максимальное число последовательности (см. задачу 7);
- 2) второй раз подсчитать количество максимумов (см. задачу 4).

Есть более рациональный способ решения, при котором числа последовательности вводятся и обрабатываются только один раз. Идея такая: обрабатывая числа, кроме значения максимума контролировать также количество элементов, равных максимальному (*кол_макс*). Если очередное число оказывается больше текущего максимума — оно принимается в качестве максимального значения, а величина *кол_макс* — равной 1. Если же очередной элемент последовательности не больше максимального числа, то сравниваем его с максимумом. Если они равны, то встретился еще один максимум, и значение *кол_макс* увеличиваем на 1.

Соответствующий фрагмент программы:

```

|Начальное присваивание значений искомым величинам
ввод а |1-е число
максимальное := а
кол_макс := 1
|Рассматриваем остальные числа
нц для i от 2 до n
    ввод а
    если а > максимальное
        то |Встретился новый максимум
            |Принимаем его в качестве значения максимальное
            максимальное := а
            |Пока он — единственный
            кол_макс := 1
        иначе
            |Проверяем, не равно ли очередное значение "старому" максимуму
            если а = максимальное
                то |Встретился еще один максимум
                    |Учитываем это
                    кол_макс := кол_макс + 1
            все
    все
кц
вывод нс, кол_макс

```

Если диапазон возможных значений элементов последовательности известен, то и здесь можно отдельно не рассматривать первое число:

```

|Начальное присваивание значений искомым величинам
максимальное := минимальное - 1
кол_макс := 0
|Рассматриваем все числа
нц для i от 1 до n
  ввод a
  если a > максимальное
...

```

где *минимальное* — нижняя граница диапазона возможных значений.

Примечание. Задача нахождения количества минимальных элементов решается аналогично.

12. Нахождение третьего максимума

“Третьим максимумом” будем называть число, которое стояло бы на третьем справа месте, если бы вся последовательность была отсортирована по неубыванию.

Задача решается аналогично задаче 10 (в первом толковании термина):

```

максимум1 := ...
максимум2 := ...
|Значение максимум3 можно не задавать
нц для i от 1 до n
  ввод a
  если a > максимум1
    |Встретилось число, большее максимум1
    то
      |Бывший второй максимум станет третьим
      максимум3 := максимум2
      |Бывший первый максимум станет вторым
      максимум2 := максимум1
      |Первым максимумом станет встреченное число
      максимум1 := a
      |Внимание - именно в таком порядке!
    иначе
      |Очередное число не больше максимум1
      |Сравниваем его со значением максимум2
      если a > максимум2
        |Встретился элемент, больший максимум2
        то
          |Бывший второй максимум станет третьим,
          максимум3 := максимум2
          |а встреченный элемент принимаем в качестве
          |нового значения максимум2
          максимум2 := a
          |Значение максимум1 не меняется
        иначе
          |Очередное число не больше максимум2
          |Сравниваем его со значением максимум3
          если a > максимум3
            то
              |Меняем только значение максимум3
              максимум3 := a
            все
          все
        все
      все
    кц

```

Примечание. Задача нахождения третьего минимума решается аналогично.

Задачи на разработку программ для самостоятельной работы

1. Известно сопротивление каждого из шести элементов электрической цепи. Все элементы соединены последовательно. Определить общее сопротивление цепи.
2. Известна масса 10 предметов, загружаемых в автомобиль грузоподъемностью Q кг. Определить, сможет ли автомобиль перевезти все 10 предметов.
3. Известны данные о стоимости каждого товара из группы. Найти общую стоимость тех товаров, которые стоят дороже 1000 рублей (количество таких товаров неизвестно).
4. Известен рост 12 юношей. Определить, сколько из них имеют рост менее 165 см.
5. Известны оценки по информатике каждого ученика класса. Определить количество пятерок.
6. Даны 10 целых чисел. Определить сумму тех из них, которые оканчиваются нулём.
7. Известны данные о количестве осадков, выпавших за каждый день некоторого месяца. Определить общее количество осадков, выпавших второго, четвёртого и т.д. числа этого месяца. Оператор цикла с параметром, шаг изменения которого отличается от 1 или от -1 , не использовать.
8. Известен рост каждого ученика класса. Рост мальчиков условно задан отрицательными числами. Определить средний рост мальчиков и средний рост девочек.
9. Даны 12 целых чисел. Определить среднее арифметическое чётных чисел и арифметическое нечётных чисел.
10. Известны расстояния от Москвы до нескольких городов. Найти расстояние от Москвы до самого близкого от неё города из представленных в списке.
11. Известны данные о температуре воздуха в течение месяца. Определить максимальную температуру этого месяца.
12. Дана последовательность из 10 целых чисел. Определить, имеется в ней число 13.
13. Известны данные о количестве осадков, выпавших за каждый день месяца. Какого числа выпало самое большое число осадков? Если таких дней несколько, то должна быть найдена дата последнего из них.
14. Дана последовательность из 20 целых чисел. Определить максимальное чётное число последовательности. При решении учесть, что чётных чисел в последовательности может не быть.
15. Известна информация о максимальной скорости каждой из 12 марок легковых автомобилей. Определить скорость автомобиля, больше которой только максимальное значение в массиве.
16. Известна информация о результатах 22 спортсменов, участвовавших в соревнованиях по бегу на 100 м. Определить результаты спортсменов, занявших первое и второе места. Задачу решить, не вводя известную информацию дважды.
17. Известна информация о баллах, набранных каждым из 25 учеников на ЕГЭ по информатике. Определить, сколько учеников набрали максимальную сумму баллов.
18. Известна информация о названиях 12 команд — участницах чемпионата по футболу и о количестве набранных каждой из них очков. Определить команды, занявшие первое, второе и третье место. Задачу решить при однократном вводе известной информации. Принять, что указанные призёры набрали разное количество очков.
19. Известна информация о среднедневной температуре за каждый день июля. Определить даты двух самых холодных дней.

Программы решения задач (можно не всех) на языке программирования, который вы изучаете, присылайте в редакцию. Фамилии всех приславших будут опубликованы, а авторы лучших результатов будут награждены дипломами. Срок представления ответов не ограничен.

Редакция также ждёт программы, предложенные для самостоятельной работы в предыдущем выпуске журнала “Мир информатики”.

Литература

1. Об операторах цикла. // “Мир информатики”, выпуск № 2 (июнь 2016 г.).
2. Кодификатор элементов содержания и требований к уровню подготовки выпускников образовательных организаций для проведения единого государственного экзамена по информатике и ИКТ. Возможные алгоритмические задачи для подраздела 1.1 перечня требований к уровню подготовки выпускников, достижение которых проверяется на едином государственном экзамене по информатике и ИКТ. 2016. <http://www.fipi.ru/ege-i-gve-11/demoversii-specifikacii-kodifikatory>

Дополнение

При обсуждении задачи 6 упоминалась задача получения ответа на вопрос о том, имеется ли в заданной последовательности некоторое число. Рассмотрим более общую задачу: “Определить, имеется ли в последовательности хотя бы одно число с заданными свойствами”. Пример такой задачи: “Определить, есть ли в последовательности хотя бы одно чётное число”.

Эта задача может быть решена разными способами.

Во-первых, можно использовать решение только что упоминавшейся задачи 6.

Можно также подсчитать количество чётных чисел (см. задачу 4) и по найденному значению получить искомый ответ.

Однако в обоих случаях задача решается путём рассмотрения *всех* чисел последовательности, что нерационально, поскольку искомое число может встретиться уже в начале последовательности. Желательно прекратить обработку последовательности после нахождения искомого числа (если оно есть). Для этого следует использовать оператор цикла с условием (см. [1]).

Поскольку как минимум одно число последовательности должно быть введено, то в таких случаях, как указывалось в [1], целесообразно использовать оператор цикла с постусловием.

Вспомним, что оператор цикла с параметром, который мы использовали во всех задачах ранее:

```
нц для i от 1 до n
  | Ввод и обработка очередного числа a
  ...
кц
```

может быть преобразован в следующий оператор цикла с постусловием:

```
i := 1
нц
  | Ввод и обработка очередного числа a
  ...
  i := i + 1
кц_при i > n
```

Однако при этом по-прежнему будут вводиться и обрабатываться все числа последовательности. Надо изменить условие окончания работы оператора. Можно рассуждать так. Необходимо прекратить вводить и рассматривать числа последовательности, когда встретится число с заданными свойствами или когда вся последовательность будет рассмотрена.

Например, для задачи о чётных числах (см. чуть выше) соответствующее условие:

$\text{mod}(a, 2) = 0$ **или** $i > n$

где *mod* — функция школьного алгоритмического языка, возвращающая остаток от деления своего первого аргумента на второй (в других языках программирования для этого используются не функция, а специальная операция).

Запишем его в программе:

```
i := 1
нц
  | Ввод очередного числа a
```

```

...
i := i + 1
кц_при i > n или mod(a, 2) = 0

```

После этого важным является вопрос о том, как получить ответ на вопрос, имеются ли в последовательности числа с заданными свойствами (в нашем случае — чётные числа).

Попробуем в качестве критерия использовать значение величины i , сравнив его со значением n :

```

| Вывод результата
если i > n
то
    вывод нс, "Нет, таких чисел нет"
иначе
    вывод нс, "Да, имеются"
все

```

Примем для простоты $n = 3$ и рассмотрим различные варианты последовательности из трёх чисел:

№	Числа	Ответ
1	1, 2, 7	"Да, имеются"
2	3, 5, 7	"Нет, таких чисел нет"
3	1, 3, 6	"Нет, таких чисел нет"

Видно, что в третьем случае ответ неверный (почему?).

Попытки изменить условие, связанное с i , правильный результат также не дадут (проверьте!).

А вот использование в условии значения a :

```

если mod(a, 2) = 0
то
    вывод нс, "Да, имеются"
иначе
    вывод нс, "Нет, таких чисел нет"
все

```

даёт правильный ответ во всех трёх случаях (проверьте!).

В заключение укажем на одну типичную ошибку, встречающуюся при решении обсуждаемой задачи, — использование переменной (логического типа или принимающей значения 0 и 1), которая фиксирует факт соблюдения для каждого проверяемого числа последовательности заданных свойств, и вывод ответа в зависимости от значения этой переменной:

```

нц для i от 1 до n
    если <условие>
        то
            имеется := 1
        иначе
            имеется := 0
    все
кц
если имеется = 1
то
    вывод нс, "Да, имеется"
иначе
    вывод нс, "Нет, такого элемента нет"
все

```

Здесь ответ может быть ошибочным — он зависит от того, обладает ли заданными свойствами *только последнее число* последовательности!

Русская семафорная азбука

Как вам, очевидно, известно, передача информации возможна разными способами. Например, на флоте для передачи сообщений между кораблями применялась так называемая “семафорная азбука”. В ней передача информации производилась сигнальщиками с помощью двух флажков. Каждой букве и служебному знаку соответствует определенное положение рук с флажками, а всё сообщение состояло из слов, составленных из букв, изображаемых соответствующим положением флажков.

Русский вариант азбуки составлен в соответствии с русским алфавитом и включает 29 букв (см. *рис. 1*) и ряд служебных знаков (*рис. 2*). Он не содержит цифр и знаков препинания. Их передача производилась словами, по буквам. Например, цифра 7 передавалась словом “семь”, а знак “,” — словом “запятая”.

Буква	Положение флагов	Буква	Положение флагов	Буква	Положение флагов
А		Л		Х	
Б		М		Ц	
В		Н		Ч	
Г		О		Ш	
Д		П		Щ	
Е, Ё, Э		Р		Ъ, Ь	
Ж		С		Ы	

З		Т		Ю	
И, Й		У		Я	
К		Ф			

Рис. 1

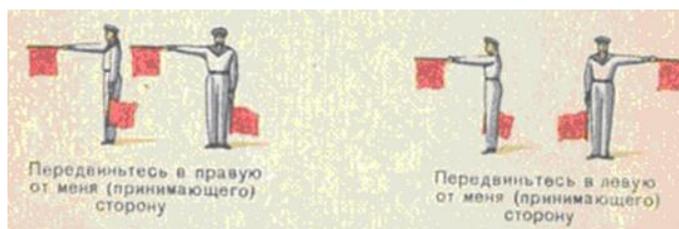
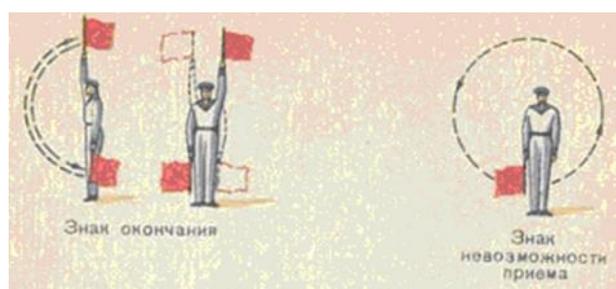
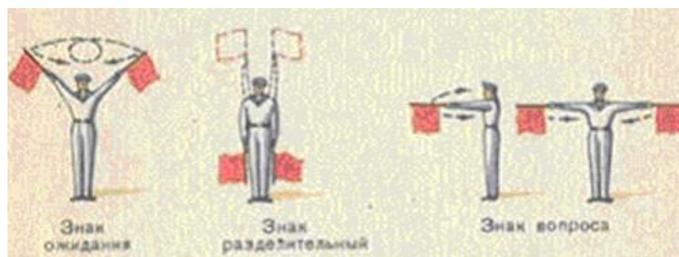
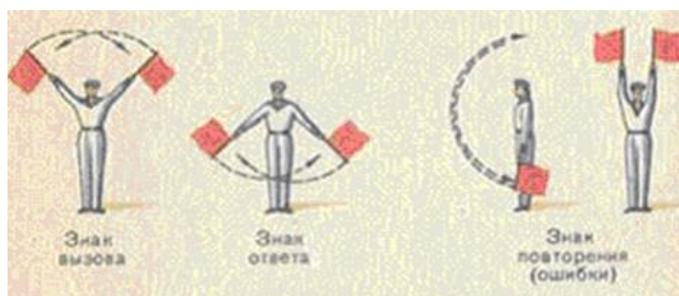




Рис. 2

Размер ткани флажков составлял 30×35 см. Цвет ткани зависел от времени суток, например, в тёмное время суток — использовались флажки с тканью светлого тона (жёлтый, белый); а в светлое время суток — с тканью тёмного тона (красный, чёрный). При отсутствии флажков — семафорили головным убором (бескозырками). Средняя скорость передачи флажным семафором обученным сигнальщиком составляла 60–80 знаков в минуту.

Должность сигнальщика на Российском флоте была введена в 1869 году. Как можно догадаться, специальность это не простая. Научиться стремительно переводить слова в движения флажками и обратно, да ещё во время боя или аварии, — задача не из лёгких. В 2011 году семафорную азбуку исключили из учебного плана будущих флотских сигнальщиков. Причина — недостаточное количество учебного времени для освоения этой системы связи, а также наличие более современной системы световой сигнализации, в которой используются не флаги, а прожектор и азбука Морзе. Тем не менее, хотя бы один матрос, знающий семафорную азбуку, на корабле обязательно должен быть. Современная техника может не сработать (например, прожектор совершенно непригоден под прямыми солнечными лучами, радиостанция или мобильный телефон могут выйти из строя). А вот старинная флажковая сигнализация и простое человеческое мужество никогда не подведут...

Задания для самостоятельной работы

1. Расшифруйте сообщение [2]:



2. Зашифруйте с помощью сигналов семафорной азбуки сообщение:
ПОДОЙДИ БЛИЖЕ

Ответ можете оформить с использованием схем, показывающих только положение флажков.

Ответы присылайте в редакцию.

Литература

1. <https://ru.wikipedia.org>
2. <http://www.yaklass.ru>



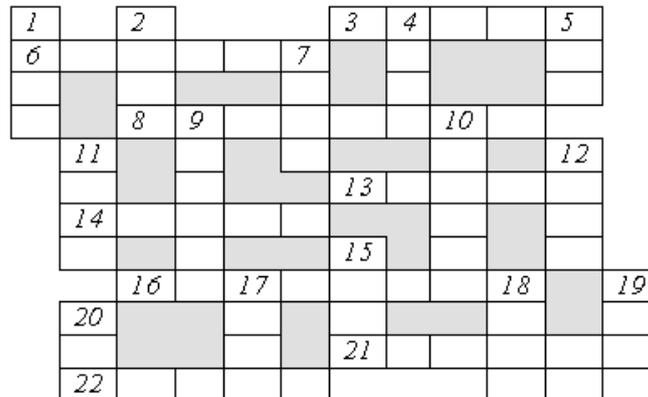
Степан Осипович Макаров (1848–1904)

Русский военно-морской деятель, океанограф, полярный исследователь, кораблестроитель, вице-адмирал.

В 1895 году разработал русскую семафорную азбуку.

Кроссворд

Решите, пожалуйста, кроссворд.



По горизонтали

3. Один из контактов транзистора типа МОП (Металл — Оксид — Полупроводник).
6. Рассмотрение и при необходимости обработка всех элементов массива.
8. Изображение чего-нибудь, рассказ о ком-нибудь (о чём-нибудь) в письменной или устной форме.
13. Поименованная группа файлов, объединённых по какому-то признаку.
14. Процесс написания текста на компьютере.
16. Совокупность четко определённых действий для решения задачи за определённое число шагов.
21. Указатель места на экране монитора.
22. Так иногда называют столбец в таблице.

По вертикали

1. Непрошенное рекламное сообщение, сетевой “мусор”.
2. Язык программирования (как правило, для начинающих изучать последнее).
4. Жаргонное название результата обработки изображения на устройстве ввода информации в компьютер.
5. Число в системе условных обозначений символов.
7. Разновидность носителя информации.
9. Название клавиши.
10. Ввоз товаров из-за рубежа, а также вставка в документ приложений Windows объектов из других приложений.
11. Часть экрана, занимаемая приложением или документом Windows.
12. ... данных.
15. Синоним слова “дорожка” (участка магнитного диска).
17. Конечное число точек на плоскости, соединённых отрезками кривых линий.
18. Устройство для соединения двух участков локальной сети.
19. Знак препинания.
20. Элемент языка разметки гипертекста.

Ответы (можно не ко всем терминам) присылайте в редакцию.

Пропущенное число

Какое число надо поставить вместо символа “?” в последовательности:
10, 11, 12, 13, 14, 15, 16, 17, 20, 22, 24, 31, 100, ?, 10000

Кто где живёт и в чём одет?

Решите, пожалуйста, две задачи⁴.

1. Коля и Саша носят фамилии Чернов и Белов. Какую фамилию носит каждый из них, если Саша с Черновым живут в разных домах?

2. Три подруги вышли в белом, зелёном и синем платьях и в белых, зелёных и синих туфлях. Известно, что только у Ани цвета платьев и туфель совпадают. Ни туфли, ни платье Светы не были белыми. Наташа была в зелёных туфлях. Определите цвета туфель и платья каждой из девушек.

Ответы присылайте в редакцию. Можно решать не обе задачи.

Переставить шашки

На столе лежат 6 шашек в ряд попеременно — чёрная, белая, чёрная и т.д.:



Справа и слева имеется свободное место, достаточное для размещения четырёх шашек. Требуется переместить шашки так, чтобы на этих шести местах слева оказались все белые, а вслед за ними все чёрные шашки. При этом перемещать на свободное место нужно сразу две шашки, не меняя порядка, в котором они лежат.

Алгоритм решения задачи оформите в виде:

Исходное состояние	___ Ч Б Ч Б ___
Ход 1	___ Ч Б __ Ч Б __
...	

Два sudoku

Решите, пожалуйста, две японские головоломки “судоку”:

1) простую:

		5	9					
1	7						3	6
					6			
		1	5	4	8	9		
		7		3		5		
		2	1	6	7	3		
			4					
2	3						5	7
					2	1		

2) сложную:

		3		2	4	8	1	
	4		3	8		9		
	2			5	6			
4		8	1	6				
					9			
		2		7			4	6
			8		3			
			6	4				
6	7							

Ответы (можно не на все головоломки) присылайте в редакцию.

Срок представления ответов ко всем заданиям — 30 сентября.

⁴ Задачи предназначены для учащихся 1–7-х классов.

Екатерина Логвиновна Ющенко

Когда говорят о связи двух понятий — “программирование” и “женщины”, то чаще всего имеют в виду двух женщин — Аду Лавлейс [1] и Грейс Хоппер [2].

Но есть еще одна женщина, чей вклад в теорию и практику программирования не менее значителен. Это Екатерина Логвиновна Ющенко (девичья фамилия Рвачева).

Е.Л. Ющенко (8 декабря 1919 – 15 августа 2001) — кибернетик, основатель первой советской школы теоретического программирования, автор одного из первых в мире языков программирования высокого уровня.

Екатерина Логвиновна родилась на Украине, в бывшей столице запорожских казаков городе Чигирине, в семье учителя истории и географии Логвина Федоровича Рвачева, выходца из Подмосковья.

Училась на первом курсе физико-математического факультета Киевского университета имени Т.Г. Шевченко. Но в 1937 году, из-за того, что её отца посчитали украинским националистом (хотя по-украински он говорил с явным акцентом, сразу выдававшим его российское происхождение) и арестовали⁵, Катю из университета исключили.

Е.Л. Ющенко впоследствии рассказывала [3]: “Летом 1938 года я подала документы в Московский университет. Мне сообщили, что меня приняли, но без предоставления общежития. Это меня не устраивало. От кого-то узнала, что университет в Воронеже никому не отказывает в жилье. Приехала туда в сентябре, уже шли занятия. Ознакомившись с моими документами, в которых было сказано, что родители арестованы, мне сказали, что могут принять и поселить в общежитии, но стипендию не назначат.

Случайно, выходя из института, увидела на доске объявлений телеграмму в адрес Воронежского университета, присланную из Самарканда. В ней сообщалось, что Узбекский государственный университет объявляет набор студентов, причём обеспечивает их общежитием, стипендией и оплачивает дорогу.

После всех мытарств и унижений при попытках продолжить учёбу, — а другого я просто не мыслила, — это показалось возможным спасением.

К счастью, так и получилось. Я вся отдалась учёбе. Не пропускала ни одной лекции, читала и перечитывала учебники, перерешала все задачи, имевшиеся в учебниках и задачниках. И так было все три года. Возможность завершить, наконец, своё образование окрылила меня и помогала пережить нагрянувшую беду. Но я всегда помнила своих родителей и так далёкую от Самарканда Украину. Иногда мне было особенно тяжело и тогда я пыталась выразить свои чувства стихами”.

Вот эти проникновенно-пронзительные строчки, написанные в мае 1939 года в Самарканде, когда Катя заканчивала первый курс университета, проведя целый год вдали от родины (приводятся на языке оригинала):

Журба по Україні (Тоска по Украине)

Я знаю, що таке журба по рідній, милій Україні

Я знаю, як рида душа в далекім краї, на чужині.

Я знаю, тяжко зустрічат пору весняну за горами,

Зітхат та мріять, як дзюрчать струмки весняні під ногами.

Зима змішалась з літнім днем, цвіте урюк, дощ не змовкає,

І на сирий, слизький дивал, вітер пилюсточки збиває...

Не чути запахів квіток і соловейка в темнім гаю,

Не чути гуркїту річок, як то бувало в ріднім краю...

⁵ В 50-х годах Л.В. Рвачев и его жена были реабилитированы.

*О краю рідний, щирій мій, ти, краю вірний, серцю милий,
Нащо ж шукаю я собі другого місця для могили?
Ні, не залишила тебе, я тут — душа моя з тобою,
І раз вернувшись назавжди вже не розстанусь я з тобою...⁶*

“Весной 1941 года я досрочно сдала экзамены за третий курс, — продолжала рассказ Екатерина Логвиновна, — и поехала к матери, которая жила в Овруче. По дороге я рассчитывала заехать к брату, что жил под Киевом в селе Бородянка.

На киевский вокзал поезд пришел поздно ночью 21-го июня. Пригородным, ровно в полночь, я добралась до Бородянки. Полустанок был безлюден. К селу вела грунтовая дорога. Подождав немного и никого не увидев, я пошла по ней, едва таща тяжёлый чемодан с подарками — фруктами, орехами, книгами. Через километр дорога раздвоилась. По какой идти? Решила ждать до утра. Оттащила чемоданы на бугорок с травой, присела сама и... уснула. Разбудил сильный гул над головой. Прямо надо мной один за другим пролетели семь необычного вида самолётов с крестами на крыльях. «Санитарные, что ли?» — подумала я. Когда вдалеке раздались глухие взрывы, удивилась: «Вот работяги и в воскресенье работают!». Наступал рассвет. Появился прохожий. Дом брата оказался недалеко. Мне были очень рады, угостили молоком, пораспрашивали и все легли досыпать. Проснулись поздно. Брат вышел на улицу и быстро вернулся:

— Говорят Германия объявила нам войну и утром бомбили Киев!

Так я встретила первый день войны...

Я вернулась в Самарканд. Оказалось, что университет объединили со Среднеазиатским государственным университетом в Ташкенте и мне пришлось поехать туда. Появилась в университете как раз к началу занятий. Одновременно устроилась на работу на военном предприятии, выпускающем прицелы для танковых пушек. Ко мне приехали мать, сестра и брат.

Через год, когда я закончила университет, встала проблема — как жить дальше. Войне не видно конца, а назначения на работу я не получила.

Еще когда я училась в университете, студентам был зачитан призыв ЦК ВЛКСМ прийти на помощь шахтёрам Ангреновского угольного комбината, чтобы увеличить добычу угля, так необходимого стране, поскольку Донбасс был оккупирован фашистами.

Город Ангрэн был в 150 км от Ташкента. Я решила, что в такое трудное для страны время мне надо быть там, вместе со всеми, кто откликнулся на призыв. Мне определили профессию — запальщик-взрывник. Обычно на нее подбирали опытных шахтёров, проработавших полтора-два года. А тут — сразу...

В 1943 году на комбинат пришел приказ — всех, имеющих законченное педагогическое образование, вернуть в школы! Приказы тогда неукоснительно соблюдались. Нас собрали, поблагодарили за самоотверженную работу. Я вернулась в Ташкент и 1943–1944 учебный год работала преподавателем математики и физики в средней школе.

Сестра получила вызов из родного села, уже освобождённого от оккупантов. Всей семьёй мы вернулись в Украину, я устроилась учительницей в этом селе и вела уроки математики, физики, черчения и даже украинской литературы. В следующем году по приглашению старшего брата, учителя физики, переехала в г. Стрый и 1945–1946 учебный год работала в городской средней школе.

Узнав, что во Львове открылся филиал Института математики АН Украины, я поехала туда и — тут мне крупно повезло — первым, к кому я обратилась, оказался Борис Владимирович Гнеденко, будущий академик АН Украины. Увидев в моём дипломе одни пятёрки, он тут же предложил работу в его отделе теории вероятностей...

⁶ Последние две строки стихотворения (перевод Д.М. Златопольского):

Нет, не забыла я тебя, моя душа — с тобой.

И однажды вернувшись к тебе, — уже не расстанусь...

Я занималась специальными вопросами теории вероятностей. Мной были получены оригинальные результаты по многомерным устойчивым законам распределения, важным для развития квантовой механики. Я подготовила и в 1950 году успешно защитила кандидатскую диссертацию. Мой первый оппонент, академик Андрей Николаевич Колмогоров, дал, — мне приятно сказать это, — высокую оценку моей работы.

В 1950 году Гнеденко избрали действительным членом АН Украины, его отдел перевели в Киевский институт математики АН Украины. Я к этому времени уже имела вполне приличную квартиру. Когда Борис Владимирович предложил мне переехать в Киев, я сразу же согласилась, не спросив, где придется жить, хотя уже ждала ребёнка. Наверно, это было опрометчиво с моей стороны, и я почувствовала это сразу. Мы поселились в маленькой комнатухе. Для новорождённого сына не хватало ни света, ни тепла, зато было вдоволь сырости и это сказалось на его здоровье.

Институт математики был в эти годы на взлёте. Его директор академик Александр Юрьевич Ишлинский занимался теорией гироскопов — очень актуальной проблемой на то “предспутниковое” время. Ему надо было производить сложные расчёты. Для этой цели был закуплен комплект счётно-аналитических машин и установлен в подвале здания президиума АН Украины. Меня назначили руководителем вновь созданной вычислительной лаборатории, пообещав позаботиться о лучшем жильё.

Когда Сергея Алексеевича Лебедева перевели в Москву, а его лабораторию — в Институт математики, директором которого в 1952 году стал Гнеденко, он прикомандировал меня, моего мужа Алексея Андроновича Ющенко, Юрия Владимировича Благовещенского и Владимира Семеновича Королюка к этой лаборатории, расположенной под Киевом в одном из зданий бывшего монастыря в Феофании.

А.Ю. Ишлинский выполнил свое обещание. Сначала меня переместили в большую комнату дома на Крещатике, а затем предоставили финский домик в Феофании. Так я оказалась в историческом месте, где под руководством Лебедева в 1948–1951 годах была создана первая в континентальной части Европы ЭВМ МЭСМ⁷.



В должности старшего научного сотрудника Института математики АН УССР Е.Л. Ющенко работала до 1957 года, после чего стала заведующей отделом Института кибернетики АН УССР.

С 1954 года она работала в области вычислительной математики, разрабатывала алгоритмы для МЭСМ, в том числе алгоритмы решения задач внешней баллистики для ракетно-космических комплексов, разработки планов производства и т.п.

Екатерина Ющенко становится первым в СССР доктором физико-математических наук, которому эта степень была присвоена за работы по программированию. Она предложила один из первых в мировой практике языков программирования — “Адресный язык”, конструкции которого вошли в состав современных языков.

⁷ Малая электронная счётная машина. — Прим. ред.

Екатерина Логвиновна была в числе авторов (с Б.В. Гнеденко и В.С. Королюком) одного из первых учебников по программированию в СССР — книги “Элементы программирования”, изданной в 1961 году, переизданной в 1964 году, а также в переводе: в 1964 году в ГДР и Венгрии, а 1969 году во Франции.

Под её руководством и с непосредственным участием была создана серия инструментально-технологических комплексов. За работы в этой отрасли ей были присуждены две Государственные премии Украины и премия Совета Министров СССР, а за теоретические разработки в области компьютерной алгебры — академическая премия имени В.М. Глушкова. Также Е.Л. Ющенко награждена Орденом княгини Ольги. Была членом-корреспондентом Национальной академии наук Украины, почётным членом Академии наук Молдавии.

Е.Л. Ющенко получила 5 авторских свидетельств на изобретение, ею разработано 8 Государственных стандартов Украины. Она является автором более 200 научных работ, в том числе 23 монографий и учебных пособий, часть которых была переведена и издана в Германии, Чехии, Венгрии, Франции и Дании.

Ею подготовлено 45 кандидатов и 11 докторов наук.

Была членом редколлегии журналов “Кибернетика и системный анализ” и “Проблемы программирования”, членом Научного совета по проблеме “Кибернетика” Национальной академии наук Украины, членом квалификационного совета по присуждению учёных степеней.

В апреле 1943 г. тогда шахтёр-запальщик Катя Рвачева, находясь в далеком-далеком от Украины городе Ангрене, где она добывала уголь, так необходимый стране для победы над фашистскими захватчиками, словами стихотворения “Зори” выразила свою заветную мечту:

*...Вечер огни разослал по долине,
Чудится: звёзды в Ангрене в плену.
Хочется мне поскорее к вершине,
К звёздам поближе... Сама не пойму.
Её мечта исполнилась...*

Литература

1. Ада Лавлейс. <https://ru.wikipedia.org/>
2. Грейс Хоппер. <https://ru.wikipedia.org/>
3. Малиновский Б.М. История вычислительной техники в лицах. – К.: Фирма “КИТ”, ПТОО “А.С.К.”, 1995.
4. http://www.icfcst.kiev.ua/MUSEUM/Ushchenko-memoirs_r.html

Внимание! Конкурс!

Конкурс № 2 “Взвешивание крупы”

Разработайте алгоритм решения такой задачи: “В пакете содержится 9 кг муки. Необходимо при помощи чашечных весов с гирями распределить эту крупу по двум пакетам: в один — 2 кг, в другой — 7 кг. Имеется неограниченное количество гирь в 50 и 100 г”.

Ответы направьте в редакцию до 30 сентября по адресу mirinfo@infojournal.ru. Пожалуйста, укажите в ответе свои фамилию и имя, населённый пункт и номер школы, а также фамилию, имя и отчество учителя информатики.