



Ларина Элла Семеновна

*Учитель информатики высшей категории
МОУ лицея №2 г. Волгограда.*

Сторож, скобочка, отрезок... Какая между ними связь?

Изучать программирование по силам как «технарю», так и «гуманитарию», хотя «гуманитариям» труднее даётся алгоритмизация. Но стоит немного изменить подход в преподавании программирования, сориентировав курс на решение не математических, а «житейских» задач, как сразу же появится результат – ребята, считавшие себя не способными программировать, на волне интереса начинают с лёгкостью постигать вершины программистского мастерства!

И проблема недостатка времени, отведённого на изучение программирования в школьном курсе информатики, легко решится путём умелой группировки большого количества задач по принципу единого метода решения.

Сторож, охраняющий музей, приходит на работу к определённом часу. Его сменяет другой сторож, второго – меняет третий. Допустим, мы знаем время прихода и время ухода каждого сторожа. Можно ли сказать, когда музей был без охраны? А написать программу, в которую в качестве входных данных вводятся время прихода и ухода каждого из N сторожей, а в качестве выходных – выводится фраза «Музей всегда был под охраной» или «Ворам путь открыт»?

А вот ещё задача: « N отрезков на координатной прямой заданы координатами своих концов. Определите, образуют ли они связную область (состоящую из пересекающихся

отрезков) или таких связных областей несколько».



Какая же между этими задачами связь? Ещё больше тумана я напускаю, если приведу условие третьей задачи: «На вход программы вводится арифметическое выражение, в котором приоритет выполняемых операций устанавливаются скобки. Проверьте, правильно ли они расставлены?». Вы удивитесь, если я скажу, что и третья задача очень тесно связана с первыми двумя! В чём заключается эта связь? Не приезжает же сторож на работу верхом на скобке и не в длинах отрезков же измеряется время работы сторожей?

Конечно, нет! Хотя... (все остальные фантастические версии будут рассмотрены на страницах другого журнала). Моя же версия такова – все эти задачи объединены единым методом решения.

Но перед разбором этого интересного алгоритма решения трёх задач введём некоторые упрощения в их условия. Допустим, первый сторож приходит в музей и выключает сигнализацию, а последний сторож её включает, ночью же музей находится под охраной только сигнализации.

В задаче со скобками договоримся, что «избыточность» скобок не является неправильной расстановкой, допустим, и в выражении $((a + b))$ скобки расставлены верно.

И ещё условимся, что время прихода и время ухода сторожей различны, как и координаты начала и конца отрезков.

Приступим, наконец, к разработке общей стратегии в решении этих задач.

Для начала нам потребуется хранить входные данные в одномерном массиве – время прихода-ухода сторожей, пары координат отрезков (начала и конца), символов арифметического выражения.

А «изюминкой» в решении будет использование дополнительного массива «флажков» в ситуациях, когда необходимо отметить наступление и окончание какого-то события (открытие – закрытие скобок, приход – уход сторожа, начало – конец отрезка). При наступлении события «флажок» примет значение «1», при окончании – «-1». Наша работа будет базироваться на типовых алгоритмах обработки одномерных массивов (потребуется знания типовых алгоритмов суммирования элементов массива, сортировки).

Начнём с задачи о расстановке скобок. Суть решения такова:

Допустим, дано арифметическое выражение:

$(a + b((4a - 8)b + 3) - 15b)$,
которое вводим в переменную stroka.

• Со строкой stroka будем работать, как с одномерным массивом.

Массив Flag заполняем флажками:

«1» – если элемент массива stroka – открывающаяся скобка;

«-1» – если элемент массива stroka – закрывающаяся скобка (рис. 1).

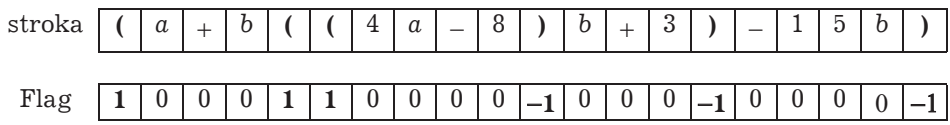


Рис. 1

• Суммируем элементы массива Flag. Если в процессе перебора элементов сумма станет отрица-

тельной, значит, скобки расставлены неверно. Итоговая сумма должна быть равна 0. В нашем случае

сумма не стала отрицательной, а в конце перебора она обнулилась

(рис. 2), значит, скобки расставлены верно:

Flag	1	0	0	0	1	1	0	0	0	0	-1	0	0	0	-1	0	0	0	0	-1
S	1	1	1	1	2	3	3	3	3	3	2	2	2	2	1	1	1	1	1	0

Рис. 2

Решение задачи на Паскале:

```
const m=10;
var   flag: array [1..m] of integer;
      stroka: string;
      i,s,n,x: integer;
begin
  s:=0;
  writeln ('введите ар. выражение');
  readln (stroka);
  n:=length (stroka);
  for i:=1 to n do
    begin
      if stroka [i]="(" then flag [i]:=1;
      if stroka [i]=")" then flag [i]:=-1;
    end;
  s:=0;
  for i:=1 to n do
    begin
      s:=s+flag [i];
      if s<0 then x:=1;
    end;
  if (s=0) and (x=0) then writeln ('скобки расставлены верно')
  else writeln ('скобки расставлены неверно');
end.
```

Тест:

Дано:	$(4*5+1)*((2/6-2*3)+4))$	$(6+56))-90*(5-2*(3-7/3)$
Результат:	скобки расставлены верно	скобки расставлены неверно

Приступаем к решению задачи об охране музея. Вспомним условие: «В музее работают сторожа. Для каждого сторожа известно время при-

хода на работу и время ухода. Определить, всегда ли музей охранялся».

Пример входных данных приведен в табл. 1:

Таблица 1

	Время прихода	Время ухода
1 сторож	8.00	12.00
2 сторож	11.00	16.00
3 сторож	15.00	19.30
4 сторож	20.00	23.50

Идея решения:

- Заполняем массивы (смотри рис. 3):

А: временем прихода и ухода сторожей (анализируй данные в табл. 1);

Flag:

«1» – если соответствующий элемент массива А – время прихода сторожа;

«-1» – если элемент массива А – время ухода сторожа;

A	8.00	12.00	11.00	16.00	15.00	19.30	20.00	23.50
Flag	1	-1	1	-1	1	-1	1	-1

Рис. 3

- Сортируем массив А, одновременно с перестановкой элементов

массива А переставляя элементы массива Flag (рис. 4):

A	8.00	11.00	12.00	15.00	16.00	19.30	20.00	23.50
Flag	1	1	-1	1	-1	-1	1	-1

Рис. 4

- Суммируем элементы второго массива. Если текущая сумма обнулилась (но конец массива не достиг-

нут), то, значит, галерея осталась без охраны (см. рис. 5 – в 19.30 сторож ушёл, а смена ещё не пришла).

A	8.00	11.00	12.00	15.00	16.00	19.30	20.00	23.50
Flag	1	1	-1	1	-1	-1	1	-1

Сумма=1 Сумма=2 Сумма=1 Сумма=2 Сумма=1 **Сумма=0** Сумма=1 Сумма=0

Рис. 5

Решение задачи на Паскале:

```

const m=20;
var   a, flag: array [1..m] of integer;
      i,s,k,n,x: integer;
begin
  writeln ('количество сторожей');
  readln (n);
  j:=1;
  for i:=1 to n do
    begin
      writeln ('время прихода, ухода');
      readln (a[j], a[j+1]);
      flag [j]:=1;
      flag [j+1]:=-1;
      j:=j+2;
    end;
  for j:=2*n downto 2 do
    for i:=1 to j-1 do

```

```

        if a[i]>a[i+1] then
            begin
x:=a[i]; a[i]:=a[i+1]; a[i+1]:=x;
x:=flag[i]; flag [i]:=flag [i+1]; flag [i+1]:=x;
end;
        k:=0;
        s:=0;
        for i:=1 to 2*n do
            begin
                s:=s+flag [i];
                if s=0 then k:=k+1;
            end;
        if k=1 then writeln ('галерея всегда охранялась')
            else writeln ('галерея оставалась без охраны', k-1,
'раз');
        end.

```

Тест:

<i>Дано:</i>	$n=3$	$n=3$
	8	8
	10	10
	9	9
	12	12
	11	13
	13	14
<i>Результат:</i>	охранялась	не охранялась 1 раз

Ну и, наконец, третья задача:
« N отрезков на координатной прямой заданы координатами своих концов. Определить количество **связных областей**».

Идея решения:

- Заполняем массивы:

A: координатами начала и конца отрезков;

Flag:

«1» – если соответствующий элемент массива A – координата

начала отрезка;

«-1» – если элемент массива A – координата конца отрезка.

- Сортируем массив A, одновременно переставляя элементы массива Flag.

- Суммируем элементы массива Flag. Если текущая сумма обнулилась (но конец массива не достигнут), то, значит, получена одна связная область. Количество обнулений и будет количеством связных областей.

Решение задачи на Паскале:

```

const m=20;
var   a, flag: array [1..m] of integer;
      i,j,s,n,k,x: integer;
begin
    writeln ('количество отрезков'); readln (n);
    j:=1;
    for i:=1 to n do
        begin
            writeln ('введите координаты начала и конца отрезка');

```

```

        readln (a[j], a[j+1]);
        flag [j]:=1;
        flag [j+1]:=-1;
        j:=j+2;
    end;
    for j:=2*n downto 2 do
        for i:=1 to j-1 do
            if a[i]>a[i+1] then
begin
x:=a[i]; a[i]:=a[i+1]; a[i+1]:=x;
x:=flag[i]; flag [i]:= flag [i+1]; flag [i+1]:=x;
end;
        s:=0;
        k:=0;
        for i:=1 to 2*n do
            begin
                s:=s+flag [i];
                if s=0 then k:=k+1;
            end;
        writeln ('количество связанных областей=', k);
    end.

```

Тест:

Дано:	$n=3$	$n=3$
	1	1
	5	3
	4	4
	9	9
	11	11
Результат:	15	15
	2	3

Решив все три задачи, можно приступить к обдумыванию частных случаев:

- Все сторожа меняют друг друга в течение суток (и ночью также музей охраняется сторожем).
- Попробуйте установить в программе также случай правиль-

ной, но «избыточной» расстановки скобок.

- Время ухода сторожа может совпадать с временем прихода его сменщика, как и координаты концов отрезков.

Успехов вам в программировании!

Литература

1. Светозарова Г.И., Мельников А.А., Козловский А.В. Практикум по программированию на языке Бейсик: учеб. пособие для вузов. – М.: Наука, Гл. ред. физ.-мат. лит., 1988.
2. Драгене В.А., Григас Г.К., Аугутис К.Ф. 100 задач по программированию: книга для учащихся. – М.: Просвещение, 1993.
3. Абрамов С.А., Зима Е.В. Начала программирования на языке Паскаль. – М.: Наука, 1987.
4. <http://intuit.ru/studies/courses/2293/593/info>