2011 год

Всероссийская олимпиада школьников по информатике
II (муниципальный этап)

Методические рекомендации по разбору предложенных олимпиадных задач
9-11 классы

Задача 1. Сумма прописью **15 баллов**

Основная идея решения задачи заключается в построении вспомогательного алгоритма (функции), позволяющего получив на входе число (достаточно использования типа byte), на выходе дать его запись в текстовой форме. Следует обратить внимание, что при сложении двухзначных чисел может получиться трёхзначное число не более 198. В остальном, решение сводится к трёхкратному применению функции (для двух слогаемых и результата) и конкатенации.

Пример функции преобразования числа в текст:

```
function NumToText(n: byte): string;
 const c1: array[0..10] of string = (’ноль’,’один’,’два’,’три’,’четыре’,
                      ’пять’,’шесть’, ’семь’,’восемь’,’девять’,’десять’);
       c11_19: array[11..19] of string = (’одиннадцать’,’двенадцать’,
         ’тринадцать’,’четырнадцать’,’пятнадцать’,’шестнадцать’,’семнадцать’,
         ’восемнадцать’,’девятнадцать’);
       cdes: array[0..9] of string = (’’,’’,’двадцать’,’тридцать’,
         ’сорок’,’пятьдесят’, ’шестьдесят’,’семьдесят’,’восемьдесят’,
         ’девяносто’);
 var n1,n2: byte;
begin
 n1 := n mod 10; n2 := n mod 100 div 10;
 case n of
  0..10: Result := c1[n];
  11..19: Result := c11_19[n];
  20..99: if n1 <> 0 then Result := cdes[n2] + ’ ’ + c1[n1]
              else Result := cdes[n2];
  else if n = 100 then Result := ’сто ’
              else Result := ’сто ’+ NumToText(n mod 100);
 end;
end;
```

Задача 2. Угадай число **25 баллов**

Самый выгодный способ решить эту задачу – смоделировать происходящий процесс (непосредственно саму игру). Фрагмент программы, осуществляющий подобное моделирование может иметь следующий вид:

```
while s <> ’=’ do begin
   if (s[1] = ’>’) then begin
       t := center;
       center := right-round((right-center)/2);
       left := t;
   end;
   if (s[1] = ’<’) then begin
       t := center;
```

```
      center := left+round((center-left)/2);
      right := t;
   end;
   delete(s,1,1);
end;
```

Задача 3. Ступеньки **25 баллов**

Одним из способов решения задачи является моделирование процесса движения шарика по проходным ячейкам массива, имеющим значение «0». При этом, следует осуществлять контроль значения h (высоты ступеньки). Фрагмент программы, моделирующий процесс движения может иметь следующий вид:

```
y := 0; x1 := 0;
while steps[x1,y] = 1 do inc(x1);
for x := x1 to m-1 do begin
   while steps[x,y] = 0 do inc(y);
end;
```

Задача 4. Супербутерброд **35 баллов**

Возможно следующее решение задачи.

Пусть один из бутербродов выглядит так (0 — пустая часть бутерброда, 1 — с сыром):

0	1	0	0	0	1
1	0	0	0	0	0
0	0	0	0	0	1
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	0	0	0

При вращении бутерброда по часовой стрелке (а можно и против) будем оставлять кусочки сыра в верхней левой четверти.

Тогда в приведённом примере верхняя левая четверть будет выглядеть так: жёлтые и голубые сырки совпадают с верхним рисунком

1	1	1
1	1	0
0	0	0

Таким образом, остаётся в стопке таких массивов подсчитать количество единиц, стоящих на соответствующих местах — максимум из этих сумм и будет ответом.

Фрагмент программы для поворота бутерброда, где t – трехмерный массив [номер бутерброда, строка, столбец], может принимать следующий вид:

```
//поворачиваем бутерброд
 for i:=1 to mm do
             for j:=1 to mm do begin
                 if t[k,m+1-i,m+1-j]=1 then t[k,i,j]:=t[k,m+1-i,m+1-j];
                 if t[k,j,m-i+1]=1 then t[k,i,j]:=t[k,j,m-i+1];
                 if t[k,m-i+1,j]=1 then t[k,j,i]:=t[k,m-i+1,j];
             end;
```

```
//если m – нечетное, то перекладываем угловые кусочки при их наличии
         if m mod 2 = 1 then
            for i:=1 to mm do begin
                if t[k,mm+1,m-i+1]=1 then  t[k,mm+1,i]:=t[k,mm+1,m-i+1];
                if t[k,m-i+1,mm+1]=1 then  t[k,mm+1,i]:=t[k,m-i+1,mm+1];
                if t[k,i,mm+1]=1     then  t[k,mm+1,i]:=t[k,i,mm+1];

                if t[k,mm+1,m-i+1]=1 then  t[k,i,mm+1]:=t[k,mm+1,m-i+1];
                if t[k,m-i+1,mm+1]=1 then  t[k,i,mm+1]:=t[k,m-i+1,mm+1];
                if t[k,mm+1,i]=1 then  t[k,i,mm+1]:=t[k,mm+1,i];
            end;
```

2011 год

Всероссийская олимпиада школьников по информатике
II (муниципальный этап)

Методика проверки решений задач и описание системы оценивания 9-11 классы

Участники Олимпиады должны сдавать на проверку решения в виде исходного текста программы на одном из разрешенных языков программирования. Проверка решений каждого участника должна осуществляться в следующей последовательности:

- компиляция исходного текста программы;
- последовательное исполнение программы с входными данными, соответствующими тестам из набора тестов для данной задачи, подготовленного предметно-методической комиссией по информатике соответствующего этапа;
- сравнение результатов исполнения программы на каждом тесте с правильным ответом.

При компиляции исходного текста программы, которую участник сдал на проверку, необходимо учитывать следующее.

1) Жюри должно использовать вполне определенные команды компиляции, соответствующие выбранному участником языку программирования. Таблица команд компиляции доводится до сведения всех участников перед началом каждого тура и должна содержаться в Памятке участнику.
2) Желательно учитывать, что размер файла с исходным текстом программы не должен превышать 256 Кбайт. Время компиляции программы не должно превышать одной минуты.

В случае нарушения принятых жюри ограничений решение участника считается неправильным и никакие баллы за эту задачу участнику не начисляются. Информация об этих ограничениях также должна быть размещена в Памятке участнику.

При исполнении программы на каждом тесте, в первую очередь, жюри должно определить, нарушаются ли присутствующие в условии этой задачи ограничения на время работы программы на отдельном тесте и размер доступной программе памяти в процессе ее исполнения. В случае нарушения имеющих место ограничений баллы за этот тест участнику не начисляются.

При проведении муниципального этапа Олимпиады рекомендуется использовать системы проверки в автоматическом режиме с использованием специализированной программной системы проведения соревнований.

Если приведенные в условии задачи ограничения не нарушаются в процессе исполнения программы с входными данными, соответствующими конкретному тесту, то после завершения исполнения программы осуществляется проверка правильности полученного ответа. Эта проверка может осуществляться как путем сравнения полученных выходных данных с правильными ответами, так и с использованием проверяющих программ.

Все представленные на проверку решения участников сначала должны проходить предварительное тестирование на тестах из примера или примеров, приведенных в условии задачи.

При проверке решений участников с использованием специализированной программной среды соревнований процесс предварительной проверки осуществляется в

течение тура по мере посылки решений на сервер соревнований. В зависимости от возможностей проверяющей системы на окончательную проверку может приниматься либо последнее прошедшее предварительное тестирование решение одной и той же задачи, либо то, которое он должен указать. В любом случае, участник Олимпиады должен быть проинформирован до начала тура, каким образом будет определяться решение, принятое проверяющей системой для окончательной проверки. Эту информацию также следует разместить в Памятке участнику.

Максимальное количество баллов, которые может набрать участник – 100.

Примеры входных файлов представлены в файлах inputXX.txt, outputXX.txt, где XX – номер теста. Значение «XX» равное «00» соответствует тесту из условия задачи и оценивается из расчёта 1 балл. При проверке, имя входного файла должно быть заменено на имя, указанное в условии задачи. Входные файлы и эталонные файлы результата размещены в папках, номер которых соответствует номеру задачи.

Распределение баллов по тестам

Тест № (XX)	**Количество баллов**			
	Сумма прописью (15 баллов)	**Угадай число (25 баллов)**	**Ступеньки (25 баллов)**	**Супербутерброд (35 баллов)**
00	1	1	1	1
01	2	3	2	4
02	2	3	2	5
03	2	3	2	5
04	2	3	2	5
05	2	3	2	5
06	2	3	2	5
07	2	3	2	5
08	–	3	3	–
09	–	–	3	–
10	–	–	4	–