

Отчет о практике по программированию  
ученицы ... класса школы № 57

... ..

Тема: Математические ребусы

Руководитель практики: Р. В. Гусарев

Оценка:

Июнь 2013

## **Постановка задачи:**

Цель программы – решение математических ребусов на сложение. Математические ребусы на сложение – примеры на сложение, в которых все цифры заменены буквами так, что одной букве соответствует ровно одна цифра. Решить ребус - означает восстановить первоначальный вид математического равенства.

Некоторые математические ребусы имеют несколько вариантов решения. При разгадывании математических ребусов обычно условием ставится проверка всех возможных вариантов. Поэтому в основе программы лежит алгоритм перебора всех возможных соответствий «буква – цифра» для всех различных букв примера и выявление тех из них, при подстановке которых данное равенство является верным.

## **Метод решения:**

### **Алгоритм перебора:**

Программа создает соответствие «буква-цифра» для каждой буквы из последовательности различных букв так, чтобы различным буквам соответствовали различные цифры. При этом, для сокращения перебора учитывается то, что некоторым буквам могут соответствовать не все цифры (подробнее об этом см. ниже). Далее программа перебирает все последовательности таких соответствий и подставляет их в исходное равенство. Если равенство выполняется, то оно выводится на экран.

### **Пример:**

Ребус: 'АХ+АХ=ЭХ'.

Последовательность букв: 'ХАЭ'.

Буквам могут соответствовать следующие цифры:

Х: [0], А: [1,2,3,4], Э: [2,3,4,5,6,7,8,9] (подробнее об этом см. ниже).

Перебор последовательностей соответствий для 'ХАЭ':

<u>012</u>	023	034	042
013	<u>024</u>	035	043
014	025	<u>036</u>	045
015	026	037	046
016	027	038	047
017	028	039	<u>048</u>
018	029		049
019			

Подчеркнутые комбинации являются решениями:  $10+10=20$

$$20+20=40$$

$$30+30=60$$

$$40+40=80$$

### Сокращение перебора:

При переборе количество последовательностей соответствий для ребуса, содержащего 2 различные буквы, т. е. 'АБ', где А: [0,1,2,3,4,5,6,7,8,9], Б: [0,1,2,3,4,5,6,7,8,9], равно  $10*9$ . Для 10 различных букв -  $10!$ . Для того чтобы сократить перебор и уменьшить время выполнения программы, в программе предусмотрены следующие методы сокращения перебора:

- **По количеству различных букв:**

Количество различных букв должно быть не больше 10, иначе решений не будет.

- **По длине слагаемых и суммы:**

Длина каждого слагаемого не должна превышать длины суммы.

- **Первая буква слагаемого/суммы:**

Первой букве каждого слагаемого и суммы не может соответствовать цифра 0.

- **2 слагаемых:**

Длина суммы либо на 1 больше длины наибольшего из слагаемых, либо равна ей.

Если  $АБ+В=ГДЕ$ , то Г: [1]. (Если длина суммы на 1 больше длины наибольшего из слагаемых, то первой букве суммы может соответствовать только цифра 1.)

Если  $AB+AB=ГД$ , то А: [1, 2, 3, 4], Г: [2, 3, 4, 5, 6, 7, 8, 9]. (Если длина суммы и длины слагаемых равны и если первые буквы слагаемых равны, то первым буквам слагаемых могут соответствовать цифры 1, 2, 3, 4, а первой букве суммы – 2, 3, 4, 5, 6, 7, 8, 9.)

Если  $AB+ВГ=ДЕ$ , то Д:[3, 4, 5, 6, 7, 8, 9]. (Если длина суммы и длины слагаемых равны и если первые буквы слагаемых не равны, то первой букве суммы могут соответствовать цифры 3, 4, 5, 6, 7, 8, 9.)

Если  $...A+...A=...A$ , то А: [0]. (Если последние буквы слагаемых равны между собой и равны последней букве суммы, то этой букве может соответствовать только цифра 0.)

Если  $...A+...A=...Б$ , то Б: [0, 2, 4, 6, 8]. (Если последние буквы слагаемых равны между собой и не равны последней букве суммы, то последней букве суммы могут соответствовать только четные цифры: 0, 2, 4, 6, 8.)

Если  $...A+...Б=...А$ , то Б: [0]. (Если последняя буква одного слагаемого не равна последней букве другого слагаемого, но равна последней букве суммы, то последней букве другого слагаемого может соответствовать только цифра 0.)

Если  $...A...+...A...=...A...$ , то А: [0, 9].

Если  $...A...+...Б...=...A...$ , то Б: [0, 9].

- **От 3 до 10 разных слагаемых:**

Длина суммы либо на 1 больше длины наибольшего из слагаемых, либо равна ей.

Если  $...A+...+...A=...A$ , то А: [0], если количество слагаемых четно, и А: [0, 5], если количество слагаемых нечетно.

- **От 2 до 10 одинаковых слагаемых:**

Все, что написано выше, а также:

N – количество слагаемых.

Если N четное, то последней букве суммы могут соответствовать четные цифры 0, 2, 4, 6, 8. Если N=5, то последней букве суммы могут соответствовать цифры 0, 5. Если N=10, то последней букве суммы может соответствовать только цифра 0.

Если длина слагаемого равна длине суммы, то :

- при N=2 первой букве слагаемого могут соответствовать цифры 1, 2, 3, 4, а первой букве суммы – 2, 3, 4, 5, 6, 7, 8, 9;

- при N=3 первой букве слагаемого могут соответствовать цифры 1, 2, 3, а первой букве суммы – 3, 4, 5, 6, 7, 8, 9;

- при N=4 первой букве слагаемого могут соответствовать цифры 1, 2, а первой букве суммы – 4, 5, 6, 7, 8, 9;

- при N=5 первой букве слагаемого может соответствовать только цифра 1, а первой букве суммы – 5, 6, 7, 8, 9;

- при N=6 первой букве слагаемого может соответствовать только цифра 1, а первой букве суммы – 6, 7, 8, 9;

- при N=7 первой букве слагаемого может соответствовать только цифра 1, а первой букве суммы – 7, 8, 9;

- при N=8 первой букве слагаемого может соответствовать только цифра 1, а первой букве суммы – 8, 9;

- при N=9 первой букве слагаемого может соответствовать только цифра 1, а первой букве суммы – 9.

Если длина слагаемого на 1 меньше длины суммы, то :

- при N=2 первой букве суммы может соответствовать только цифра 1;

- при N=3 первой букве суммы могут соответствовать цифры 1, 2;

- при N=4 первой букве суммы могут соответствовать цифры 1, 2, 3;

- при N=5 первой букве суммы могут соответствовать цифры 1, 2, 3, 4;

- при N=6 первой букве суммы могут соответствовать цифры 1, 2, 3, 4, 5;

- при N=7 первой букве суммы могут соответствовать цифры 1, 2, 3, 4, 5, 6;

- при N=8 первой букве суммы могут соответствовать цифры 1, 2, 3, 4, 5, 6, 7;

- при N=9 первой букве суммы могут соответствовать цифры 1, 2, 3, 4, 5, 6, 7, 8.

## **Внешняя спецификация:**

### **Руководство пользователя:**

Запуск программы производится с консоли.

Вначале появляется краткое описание программы.

Дальше появляется строка 'Нажмите "п", чтобы ввести пример, или "в", чтобы выйти из программы: '.

Если ввести "в", то программы закончит работу.

Если ввести "п", то появляется строка 'Нажмите "р", чтобы ввести пример вручную, или "ф", чтобы загрузить пример из файла: '.

Если ввести "р", то программа попросит ввести пример. Примером является строка без пробелов, состоящая из русских/латинских букв и знаков + и =.

Если ввести "ф", то программа попросит ввести имя текстового файла (вводить с расширением, например: input.txt). Он должен находиться в текущем каталоге. Файл может содержать один или несколько примеров: на каждой строчке один

пример. Если примеров несколько, программа попросит ввести номер строки с примером, который вы хотите решить.

Дальше выводятся решения ребуса в виде верного равенства.

Если количество решений не больше 20, то они выведутся все. Если же количество решений больше 20, то выведутся 20 решений и программа спросит, выводить ли их дальше. Если да, то программа будет выводить следующие 20 решений до тех пор, пока они существуют и пока они запрашиваются.

После окончания решения ребуса выведется количество решений, которые выведены выше.

Дальше снова появляется строка 'Нажмите "п", чтобы ввести пример, или "в", чтобы выйти из программы: '.

## **Внутренняя спецификация:**

Когда программа получила пример, с помощью функции *create\_dic\_and\_ss (primer)* она создает список из различных букв *ss* и словарь *dic*, в котором для каждой буквы перечислены возможные подстановки.

С помощью функции *first (ss, dic)* программа генерирует первую последовательность соответствий *podst*.

Функция *search (podst, mn, ss, dic)* ищет какой элемент и как можно изменить в *podst* (*mn* – это множество, состоящее из всех различных цифр из *podst*). Эта функция выдает номер элемента *i* и значение *znach*, которое нужно присвоить этому элементу.

Функция *change (podst, i, znach, mn, ss, dic)* генерирует следующую последовательность соответствий *podst*. Эта функция рекурсивна.

Готовая последовательность соответствий проверяется функцией *check (podst, primer, ss)*. Если при подстановке равенство является верным, то оно выводится.

Функция *create\_dic\_and\_ss (primer)* использует несколько вспомогательных функций: *clon (parts, ss, tt)* если все слагаемые равны, *slag\_2 (parts, ss, tt)* если в ребусе 2 слагаемых, *slag\_3\_to\_10 (parts, ss, tt)* если в ребусе от 3 до 10 слагаемых. (*parts* – это список частей ребуса, разделенных знаками + и =; *tt* – это список такой же длины, что и *ss*, в котором *k*-ый элемент является списком возможных значений *k*-ой буквы из *ss*. Т.е. *ss* – ключи словаря, *tt* – значения ключей).

В этих функциях проверяются условия для сокращения перебора, описанные в Методе решения.

Функция *readfile(filename)* построчно читает из файла примеры.

## Описание тестов:

### Тест 1:

Ввод: вручную

Пример: удар+удар=драка

Количество различных букв: 5

Количество решений: 1

Решение:  $8126+8126=16252$

### Тест 2:

Ввод: вручную

Пример: четыре+четыре=восемь

Количество различных букв: 10

Количество решений: 3

Решения:  $239153+239153=478306$

$235193+235193=470386$

$467806+467806=935612$

Время выполнения программы: 70 с

### Тест 3:

Ввод: из файла

Пример: двести+двести+двести+двести+двести=тысяча

Количество различных букв: 10

Количество решений: 1

Решения:  $135964+135964+135964+135964+135964=679820$

Время выполнения программы: 10 с

### Тест 4:

Ввод: из файла

Пример: три+два=пять

Количество различных букв: 9

Количество решений:  $\geq 160$

Примеры решений:  $463+579=1042$

$689+574=1263$

$679+584=1263$

$684+579=1263$

$674+589=1263$

$$794+582=1376$$

$$784+592=1376$$

$$839+246=1085$$

$$849+237=1086$$

$$839+247=1086$$

$$846+239=1085$$

$$847+239=1086$$

$$836+249=1085$$

### **Тест 5:**

Ввод: из файла

Пример: сорок+сорок+сорок+сорок+сорок=двести

Количество различных букв: 10

Количество решений: 1

Решение:  $87172+87172+87172+87172+87172=435860$

Время выполнения программы: 50 с

### **Тест 6:**

Ввод: из файла

Пример: один+шесть=семь

Количество различных букв: 10

Количество решений: 0

### **Тест 7:**

Ввод: вручную

Пример: один+один+один+два=пять

Количество различных букв: 10

Количество решений: 29

Решения:  $1396+1396+1396+382=4570$

$$1287+1287+1287+235=4096$$

$$1694+1694+1694+638=5720$$

$$1738+1738+1738+726=5940$$

$$1728+1728+1728+746=5930$$

$$1698+1698+1698+630=5724$$

$$1623+1623+1623+609=5478$$

$$1724+1724+1724+736=5908$$

$$1827+1827+1827+859=6340$$



1972+1972+1972+934=6850  
1952+1952+1952+974=6830  
1924+1924+1924+958=6730  
1749+1749+1749+785=6032  
1835+1835+1835+897=6402  
1974+1974+1974+930=6852  
1954+1954+1954+970=6832  
1928+1928+1928+950=6734  
1973+1973+1973+905=6824  
1745+1745+1745+793=6028  
2147+2147+2147+139=6580  
2395+2395+2395+301=7486  
2409+2409+2409+431=7658  
2431+2431+2431+405=7698  
2591+2591+2591+567=8340  
2537+2537+2537+549=8160  
2607+2607+2607+693=8514  
2713+2713+2713+765=8904  
2745+2745+2745+701=8936  
2783+2783+2783+715=9064

### **Тест 8:**

Ввод: вручную

Пример: два+два=четыре

Количество различных букв: 8

Количество решений: 0

### **Тест 9:**

Ввод: вручную

Пример: плюс+плюс=минус

Количество различных букв: 8

Количество решений: 15

Решения: 6390+6390=12780

6790+6790=13580

6920+6920=13840

7280+7280=14560

7290+7290=14580

$$7640+7640=15280$$

$$7690+7690=15380$$

$$7820+7820=15640$$

$$7920+7920=15840$$

$$7930+7930=15860$$

$$8270+8270=16540$$

$$9230+9230=18460$$

$$9270+9270=18540$$

$$9320+9320=18640$$

$$9360+9360=18720$$

## **Оглавление:**

Постановка задачи

Метод решения

Внешняя спецификация

Руководство пользователя

Внутренняя спецификация

Описание тестов