SUAVe Open Problem: for presentation at May 4 2010 SUAVe meeting

Your mission, should you accept it, is to write SAS code to read the following data lines, which each contain an item #, description, and amount. The desired result is a SAS dataset containing three variables:

- Item (numeric)
- Description
- Amount (numeric)

Here are some data lines you can use to test your code:

```
34 Unwashed shirts 2.50
5764 Things that make you go hmmm 576.77
66004 My favourite Martian 35.69
23 Supercalifragilisticexpialidocious 55678.55
678 My mother and your mother were hanging out clothes 576.23
55 I'm writing this too late at night 44.12
678 Where is my brain? 78234.45
```

Unfortunately, the data is not organised for straightforward processing. A space separates the item (which appears first on each line) from the description, and another space following the description separates that from the amount. The complication is that spaces may also lie within the description, and you do not know how many (if any) spaces are in the description.

Your solution should be able to read data lines like the above and arrive at a dataset that looks like the following:

Item	Description	Amount
34	Unwashed shirts	2.5
5764	Things that make you go hmmm	576.77
66004	My favourite Martian	35.69
23	Supercalifragilisticexpialidocious	55678.55
678	My mother and your mother were hanging out clothes	576.23
55	I'm writing this too late at night	44.12
678	Where is my brain?	78234.45

Your solution should be able to handle any input lines that meet these rules. You can assume lines do not exceed 80 characters and that the maximum description length is 60 characters.

Note: Solutions on the following pages were presented at the SUAVe meeting on May 4th 2010. Thanks to all the presenters!

Solution #1. Gord Nigh (BC Ministry of Forests and Range)

```
data test;
input line $ 1- 80;  * input whole line of data at once;
Item = 0.0; * create numeric variables Item and Amount - so they print out  right
justified;
Amount = 0.0;
x = index(line, ' '); * find the first space in the line;
y = index(left(reverse(line)), ' '); * find the last space in the line;
Item = substr(line, 1, x - 1); * pick off the Item number;
Description = substr(line, x + 1, length(line) - y - x); * pick off the Description;
Amount = substr(line, length(line) - y + 1, y); * pick off the Amount;
cards;
34 Unwashed shirts 2.50
5764 Things that make you go hmmm 576.77
66004 My favourite Martian 35.69
23 Supercalifragilisticexpialidocious 55678.55
678 My mother and your mother were hanging out clothes 576.23
55 I'm writing this too late at night 44.12
678 Where is my brain? 78234.45
;
run;

proc print;
var Item Description Amount;
run;

quit;
```

Solution #2. Stephen Gibbs (BC Ministry of Health Services)

```
*************************************************************************************;
* SUAVe Open Problem: for presentation at May 6 2010 SUAVe meeting                  *;
* Your mission, should you accept it, is to write SAS code to read the following   *;
* data lines, which each contain an item #, description, and amount. The desired   *;
* result is a SAS dataset containing three variables:                               *;
* • Item (numeric)                                                                  *;
* • Description                                                                     *;
* • Amount (numeric)                                                                *;
* Steve Gibbs                                                                       *;
*************************************************************************************;

* Create data as one char variable *;
data sg1;
input var1 $ 1-80;
cards;
34 Unwashed shirts 2.50
5764 Things that make you go hmmm 576.77
66004 My favourite Martian 35.69
23 Supercalifragilisticexpialidocious 55678.55
678 My mother and your mother were hanging out clothes 576.23
55 I'm writing this too late at night 44.12
678 Where is my brain? 78234.45
;
run;

* Create desired 3 variables using SCAN for numeric vars and COMPRESS for char*;
* Use BEST8 format to reduce 1st row (and similar) to 1dp *;
* The use of ' ' in the AMOUNT scan statment prevents the scan stopping at '.' *;
data final (drop=var1);
 attrib item length=8 label='Item'
  description length=$60. label='Description'
  amount length=8 format=best8. label='Amount';
 set sg1;
 item=input(scan(var1,1),8.);
 description=compress(var1,'1234567890.');
 amount=input(scan(var1,-1,' '),8.);
 proc print label noobs;
run;

*********************************************************************;
*********************************************************************;
```

Solution #3. Ariel Lade (BC Ministry of Health Services)

```
*************************************************************************************;
* ARIEL LADE;
*************************************************************************************;

DATA PRICES;
   INPUT @1 INPUT $80. ;
   Item = SCAN(INPUT,1,' ');
   Amount = SCAN(INPUT,-1,' ');
   Description = SUBSTR(INPUT,length(item)+2,length(INPUT)-length(item)-
length(trim(amount))-2);

DATALINES;
34 Unwashed shirts 2.50
5764 Things that make you go hmmm 576.77
66004 My favourite Martian 35.69
23 Supercalifragilisticexpialidocious 55678.55
678 My mother and your mother were hanging out clothes 576.23
55 I'm writing this too late at night 44.12
678 Where is my brain? 78234.45
;
PROC PRINT DATA=PRICES NOOBS;
   TITLE "Output";
RUN;


*************************************************************************************;
*************************************************************************************;
```

Solution #4. Ray Ghouse (BC Ministry of Health Services)

```
Data raw;
INPUT VAR1 $80.;
CARDS;
34 Unwashed shirts 2.50
5764 Things that make you go hmmm 576.77
66004 My favourite Martian 35.69
23 Supercalifragilisticexpialidocious 55678.55
678 My mother and your mother were hanging out clothes 576.23
55 I'm writing this too late at night 44.12
678 Where is my brain? 78234.45
;
run;

data temp;
      length Description $60;
      set raw;
      if _n_=1 then do;
      prxid_item_desc_amt=prxparse("/(\d+) ([\w\W]+) (\d+\.?\d{0,2})/");
      end;
      retain prxid_item_desc_amt;
      itemdesc_match=prxmatch(prxid_item_desc_amt, VAR1);
      call prxposn(prxid_item_desc_amt, 1, pos_num, len_num);
      call prxposn(prxid_item_desc_amt, 2, pos_description, len_description);
      call prxposn(prxid_item_desc_amt, 3, pos_amount, len_amount);
      Item=substrn(VAR1, pos_num, len_num);
      Description=substrn(VAR1, pos_description, len_description);
      Amount=substrn(VAR1, pos_amount, len_amount);
run;

proc print data= temp;
var item description amount;
run;

/* Some useful notes:
Defining of regular expression using PRXPARSE and capture buffers.
Capture buffers, i.e. the data of interest is enclosed within brackets, are used to
identify a distinct data-pattern with the help of the surrounding data.

The expression consists of 3 parts: number, character string, and a number with up to two
decimal points.
Here is a breakdown of what each argument in the PRXPARSE statement means:
 \d+: data consists of multiple digits. May be replaced with something like \d{1,10};
 [\w\W]+: data has more than one word or non-word characters;
 \d+\.?\d{0,2}: more than one digit with up to two decimal points;
 the '?' matches the previous expression zero or one times. May be replaced with
something like \d{1,8}\.?\d{0,2};

The PRXPARSE function is used to create a regular expression. Since this expression is
compiled, it is usually placed in the DATA step following a statement such as IF _N_ = 1
then ….
Since this statement is executed only once, you also need to retain the value returned by
the PRXPARSE function.
Combining _N_ and RETAIN when used with the PRXPARSE function prevents executing the
function for each iteration of the DATA step.

PRXMATCH() is executed to locate the matching strings, prior to using CALL PRXPOSN() to
locate the capture buffers.

The last three lines of the data step create the variables of interest used in the print
statement. */
```

Solution #5. Dave Hosick (BC Ministry of Health Services)

```
********************************************************************************;
* DAVID HOSICK;
********************************************************************************;

Data sample_data;
INPUT rawText $80.;
CARDS;
34 Unwashed shirts 2.50
5764 Things that make you go hmmm 576.77
66004 My favourite Martian 35.69
23 Supercalifragilisticexpialidocious 55678.55
678 My mother and your mother were hanging out clothes 576.23
55 I'm writing this too late at night 44.12
678 Where is my brain? 78234.45
;
run;


Data Results (keep = Item Description Amount);
      Set sample_data;
      Format Item 8.0;
      Format Description $60.;
      Format Amount 8.2;
      patternID = prxparse('/ [\w\s\x27\x3f]+ /');
      call prxsubstr(patternID,rawText,myStart,myLength);
      Item = Substr(rawText,1,myStart);
      Description = Substr(rawText,myStart+1,myLength-2);
      Amount = Substr(rawText,myStart+myLength);
Run;


********************************************************************************;
********************************************************************************;
```