Simple Statistics Calculator

Part 1: Console Input

Pseudocode

Total = 0

Inputstore = 0

Get user input

While (input != Carriage return key)

{

Shift value of total one to left eg.(1 becomes 10,0 stays 0)

Inputstore = input(in decimal form)

total = total + inputstore

]

Testing

INPUT	RESULT(in R4)	RESULT CONDITION	REASON FOR INPUT
1234	4D2	correct	Reasonable medium sized input
0	0	correct	unique number
-12	fffffee0	incorrect	negative number
Values after return	same as value before	correct	to check if return quits program
5000000000	2A05F200	incorrect	Very large number
1	1	correct	unique value

Part 2: Stat Evaluation

Pseudocode

```
Initialise all registers to be used
Set R5 as 10 to shift values by one decimal place later
label:consoleinputstage
reset r4 to get next value
get input
while(input != cr){
        while(input!= space){
                if(input> ‘0’ && input< ‘9’){
                do steps in console input stage
                }
        }
Reset r11 to calculate new mean
Sum = total + sum
Count++
If(count == 1){
Max =total
Min = total
}
Else if(total > max) max = total
If(total < min) min = total
Remainder = sum%count
Mean = sum/count
}
```

Testing

_ = space

INPUT	COUNT	SUM	MAX	MIN	MEAN	T/F
1_1_	2	2	1	1	1	t
2_	1	2	2	2	2	T
0_	1	0	0	0	0	t
2_1_	2	3	2	1	1	f

Reasons for testing each input

1_1_: Reasonable small input, easy to check and contains more than one term

2_: To compare against 1_1_

0_: Unique value

2_1_:Mean is non integer

Part 3: Console output

Final Program usage instructions

Enter numbers separated by space and then enter when u wish to calculate answer

Handles Sums up to 7 significant decimal places

Does not allow negative values

Displays mean up to one decimal place and all other values except range to whole number rounded down

Cant input decimal values

Calculates variance ,mean ,min ,max ,sum ,count and range

Pseudocode/logic

Initialise registers

Label: userInput

Reset register total

Use register as Boolean nextNumber =false

Label: read

{

If(userinput == cr or space skip this section and branch)

If(userinput < asci 0 or > asci 9 then branch to read to ignore result and get next char)

Set nextNumber = true

Do computation as described in previous parts

b read and repeat till enter or space char is inputted

}

Label: carriage return

{

output linefeed and ME_ for formatting purposes

Set another register as Boolean finalvalue = true

Check if nextNumber = true

If(nextnumber) branch to statCalc

Else branch to redirection

}

Label: space

{

If(nextnumber) branch to read

Else display a space char

Set finalvalue = false

}

statCalc

Same as previous pseudocode but extra bits for variance calculation

Label: redirection

{

Only reachable through cr key and just calculates mean and then skips variance calculations.

Usefull to avoid overwriting certain registers

}

If (Boolean finalvalue = false) branch to userInput for next numbers entered

;display part

Multiplies remainder of sum%count by 10

Then divides by count and result is added to 10*mean and stored in register mean

Use register power to get highest power of 10 that gives a non zero answer when mean%power

Label: display

{

R0 = mean % power

Sendchar to display char

Power = power /10

If(power ==10) display . char

If(power == 1) break loop

B display

}

Repeat similar steps to display all other values

Range = max value – min value

Testing

_ = space

NOTE: I didn't bother checking range every time as it is automatically correct if max and min are correct. Also due to space constraints I have changed layout of table. Negative numbers could be tested as the – sign could not inputted, same with decimals/fractions.

INPUTS	9999999	999999999	1 to 9	no value	0
MEAN	9999999.0	Failed	5.0	failed	0
VARIANCE	0	-	6	-	0
COUNT	1	-	9	-	1
SUM	9999999	-	45	-	0
MAX	9999999	-	9	-	0
MIN	9999999	-	1	-	0
T/F	t	f	t	f	t
REASON	large value	larger value	varied numbers	unique	unique

User and system limitations

As per results above it can handle up to 7 significant digits for sum before it breaks

Can't handle no value(ie. Just enter)

Doesn't allow negative values or decimal values/fractions

Displays mean up to one decimal place only and always rounds down. Similar with other stats