

Credit Cards

Student Activity

7 8 9 10 11 12



TI-Nspire



Coding



Student



60 min

Validating Credit Cards

Imagine you are building a website that requires financial transactions to take place. Users need to enter their credit card details; however before they proceed you need to validate their credit card number, this can be done before submitting the details to the financial institution. Being part of a website, you would need to do this automatically using some basic computer code. Most credit cards contain a 16 digit number. The numbers however are not random. There is a relatively simple process to check if a credit card number is valid.¹ Your task is to write a program that will automatically check to see if a credit card number is valid or not.



Step 1

- The credit card number shown above is: **1 4 3 2 4 6 7 8 9 8 7 6 5 4 3 2**
- We start by extracting every second digit: **1 4 3 2 4 6 7 8 9 8 7 6 5 4 3 2**
- Calculate the sum of these digits: **1 + 3 + 4 + 7 + 9 + 7 + 5 + 3 = 39**
- Double this value: **2 x 39 = 78** **[Answer 1]**

Step 2

- Using every second digit, count the digits ≥ 5 **1 4 3 2 4 6 7 8 9 8 7 6 5 4 3 2**
There are 4 digits greater or equal to 5. [Answer 2]

Step 3

- Extract the alternate digits: **1 4 3 2 4 6 7 8 9 8 7 6 5 4 3 2**
- Find the sum of these digits: **1 4 + 2 + 6 + 8 + 8 + 6 + 4 + 2 = 30** **[Answer 3]**

Step 4

- Find the sum of these three answers: **39 + 4 + 30 = 73**
- The final result must be a multiple of 10 if the credit card number is valid. **Invalid Credit Card!**

¹ Valid: *Just because a credit card number is 'valid', doesn't mean it is real or attached to an existing account.*

Question: 1.

Use the algorithm above to see if the following credit card number is valid: 9513 1081 5812 5353

Coding the Algorithm

Mathematical procedures that are based on simple logic are relatively easy to code or program. In this case the user will put in their 16 digit number and the program will return the result “Valid” or “Invalid”.

Instructions:

Start a new document; insert a calculator application followed by a program.

Call the program: ValidCredit

Note that ‘ValidCredit’ is one word as program names cannot contain spaces.

Several variables will be used in this program, two things need to be done in order to use these variables consistently and without interfering with potential ‘user’ variables.

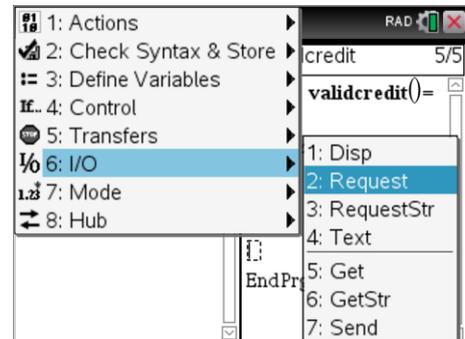
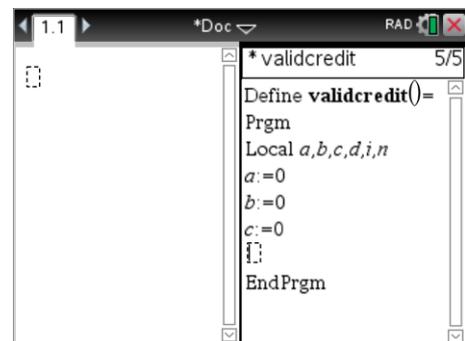
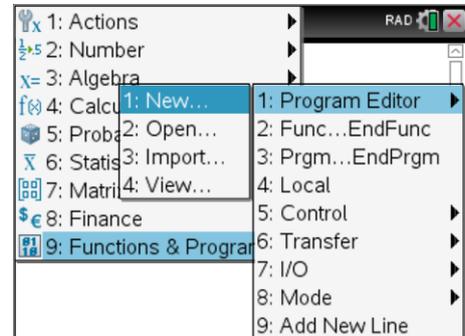
- Declare the variables as local.
Prevents them from appearing to ‘users’.
- Seed the variables.
Set some of the variables to zero so they do not contain previous data.

The credit card number needs to be entered by the user.

The **I/O** (Input / Output) menu contains the **Request** command.

Request “Enter credit card number:” , n

The credit card number will be stored into ‘n’.



The digits in the credit card need to be successively removed. In this example modular arithmetic will be used. Modular arithmetic can be thought of as ‘remainder’. $\text{Mod}(25,10) = 5$ since $25 \div 10 = 2 \text{ rem } 5$.

The integer command will also be used to remove unwanted decimal places. For example to extract the 100’s digit from 1234; $\text{Int}(\text{Mod}(1234,1000)/100) = 2$. Looking at this step by step:

- $\text{Mod}(1234,100) = 34$...since $1234 \div 1000 = 1 \text{ rem } 234$
- $234 \div 100 = 2.34$
- $\text{Int}(2.34) = 2$...since the part returned is the integer value only.

A loop is required to separate all 16 digits. The digits will be removed one by one. The above integer and modular arithmetic commands will be used. The logic statements such as **For** and **If – Then** can be obtained from the **Control** menu.

For $i, 1, 16$ *This starts a loop where $i = 1$, the first time through. The second time through the loop $i = 2 \dots$ until the last loop where $i = 16$.*

$$d := \text{int} \left(\frac{\text{mod}(n, 10^i)}{10^{i-1}} \right) \text{ Successive digits stored in } d.$$

Now that a digit has been collected from the credit card number, it needs to be determined whether the digit location is even or odd so that the appropriate step (or steps) can be executed.

Enter the instructions shown opposite. Note that there is an “IF” statement embedded in another “IF” statement. The embedded “IF” statement is only acted upon if the first condition is satisfied. Read over steps 1 and 2 in the original instructions to help understand what the program is doing here.

Variables a , b and c now hold parts of the answers in the original algorithm. **If** $2a + b + c$ is a multiple of 10, **then** the credit card number is valid, otherwise it is invalid.

The program is now ready to be checked and stored, press **Ctrl + B**.

To transfer control to the calculator application, press **Ctrl + Tab** to shift focus to the calculator application.

Press the **[VAR]** key to locate your program and press **Enter** to select and then **Enter** to run the program.

Test the program using the credit card number in Question 1.

```
*validcredit
a:=0
b:=0
c:=0
Request "Enter credit card num
For i,1,16
  d:=int(mod(n,10^i)/10^{i-1})
Endfor
EndPrm
```

```
a:=a+d
If d>=5 Then
  c:=c+1
Endif
Else
  b:=b+d
Endif
Endfor
EndPrm
```

```
Else
  b:=b+d
Endif
Endfor
If mod(2a+b+c,10)=0 Then
  Disp "Valid"
Else
  Disp "Invalid"
Endif
EndPrm
```

validcredit() "validcredit" stored

Enter credit card number: 158125353

Disp "valid"

Else

Disp "Invalid"

EndIf

Question: 2.

Use the program to test the following credit card numbers to see if they are valid.

- | | |
|------------------------|------------------------|
| a) 4049 0876 3177 5451 | b) 5248 9255 1236 4153 |
| c) 2158 2594 3168 2392 | d) 4148 2955 2187 5253 |

Question: 3.

A digit on a credit card receipt appears to be missing 4049 521A 5437 1333. Determine the value of the missing digit.

Coding Questions

These questions focus on an understanding of the coding used in this activity in addition to looking for modifications.

Question: 4.

The first IF statement includes the logic statement: $\text{mod}(i,2)=0$. In relation to the validation process what is this logic statement checking?

Question: 5.

With reference to the validation process, explain the statements: $a:= a + d$ and $c:= c + 1$

Question: 6.

Suppose all credit cards are changed to 20 digits. Identify the required changes to the program that would accommodate the new credit card changes.

Question: 7.

A new validation system requires the final sum to be a multiple of 17. Identify the necessary changes to the code.

Challenge Questions

Question: 8.

A digit is missing from a credit card receipt: X124 5132 4681 2552. Determine the value of missing digit.

Question: 9.

The first two digits are missing from a credit card receipt: XY24 5132 4681 2552. Determine the possible combinations for the two missing digits.

Note: Check some of your answers using your program.