

# Personal Polynomials

## Student Activity

7 8 9 10 11 12

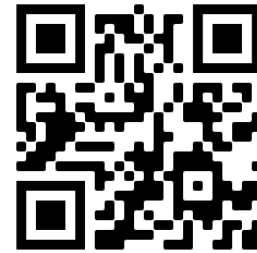


## Introduction

Not everyone has a street named after them, but everyone can have a polynomial. What does your polynomial look like? In this activity you will change your name into a set of points and determine the polynomial that passes through these points, your personal polynomial. How does your polynomial compare and interact with others? How much can you tell about someone's polynomial by just looking at their name?

You can turn your name or any text into numbers quickly and easily using TI-Nspire. Scan the QR code or follow the instructions below to see how to write a program to automate the process.

POLYNOMIAL WAY



## Instructions – Converting your name into numbers – A programming approach

Start a new TI-Nspire document and insert a Calculator application.

Type the command:

```
ord("T")
```

Next, type the command:

```
ord("t")
```

Notice the difference in values. Try other letters such as A and a.



### Technology Tip!

The 'ord()' command on the calculator returns the ASCII code for the corresponding text. ASCII code maps 128 characters, the more sophisticated UNICODE maps 2<sup>21</sup> characters.

### Question: 1.

What value needs to be subtracted from capital letters to compute the value of each letter such that A = 1, B = 2 and so on?

### Question: 2.

What value needs to be subtracted from lower case letters to compute the value of each letter such that a = 1, b = 2 and so on?

There are other useful text commands on the calculator that will prove useful.

Type the command:

```
mid("Maths",1,1)
```

Then try:

```
mid("Maths",2,1) and mid("Maths",3,1)
```

```
1.1 | *Doc | RAD | 84
ord("T")
mid("Maths",1,1)
```

### Question: 3.

Explain what the 'mid' command does.

In mathematics one function can be embedded inside another, these are referred to as composite functions. The output of one function feeds directly into the input of another. This may be expressed as:

$$f(g(x))$$

In this case, two programming commands "ord" and "mid" will be combined to produce a single result.

The output from the 'mid' command is fed into the input of the 'ord' command.

Type in the following combination:

```
ord(mid("Maths",2,1))
```

```
1.1 | *Doc | RAD | 84
ord("T")
mid("Maths",1,1)
ord(mid("Maths",2,1))
```



### Technology Tip!

A short cut for entering the `ord(mid("Maths",2,1))` combination is to start with `ord(` then arrow up to select `mid("Maths",2,1)` and press **Enter** to paste it directly into the `ord()` command.

The last piece in this puzzle is the "dim" command.

Try the following:

```
dim("Maths")
```

```
1.1 | *Doc | RAD | 84
ord("T")
mid("Maths",1,1)
ord(mid("Maths",2,1))
dim("Maths")
```

### Question: 4.

What does the `dim()` command do?

You're now ready to write a program!

From the **menu** select:

**Functions & Programs > Program Editor > New**

Enter the name for your program:

N2N (name to numbers)

The program application will appear in a split screen. Press Ctrl + 6 to ungroup the two pages. The program editor will be moved to Page 1.2. Navigate to Page 1.2 to start working on the program.

The first step in the program is to request the text to be converted into numbers. Text variables in programs are referred to as strings.

From the **menu** select

**I/O > RequestStr**

I/O = Input / Output

RequestStr = Request a string.

The Request-String command can include some prompting text relating to what is required.

Enter the following in the request command:

"Enter your name ",name

This creates a text prompt "Enter your name" and creates a 'string' variable 'name' that will hold the text to be converted into numbers and ultimately become a polynomial.

The value assigned to each letter will be determined character by character, a loop needs to be created to extract each letter using the commands explored earlier. The number of times the loop needs to be executed is determined by the number of letters in the 'name'.

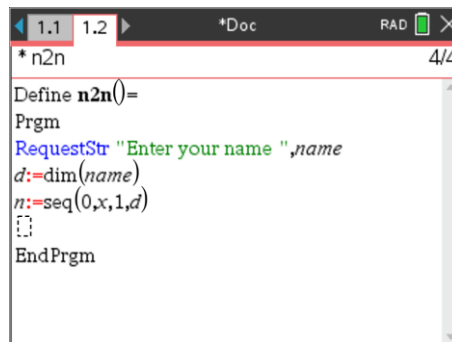
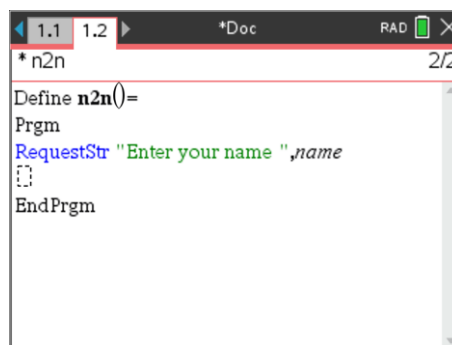
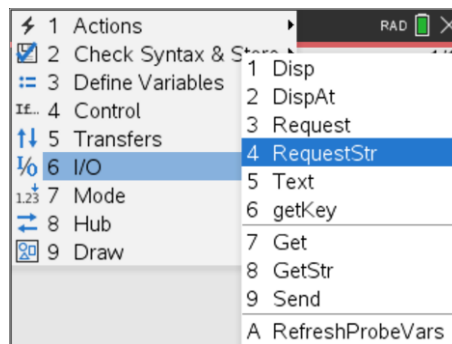
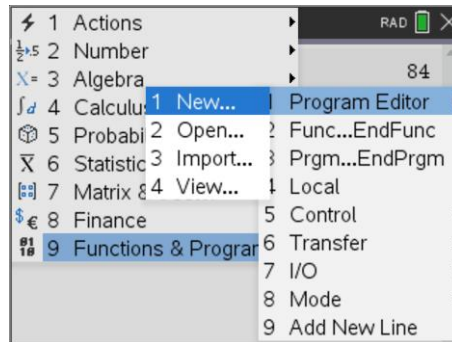
Type the following:

d:=dim(name)

A list also needs to be created to hold all the values.

n:=seq(0,x,1,d)

This creates a list filled with zeros, place holders for our final values.



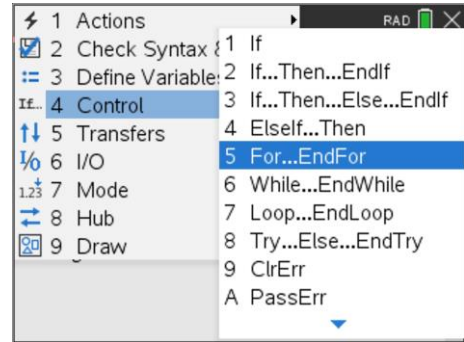
A “For” loop can be used to successively capture the value of each letter.

Use the **menu** and select:

**Control > For...EndFor**

Set the loop with a counter (i), starting at 1 and finishing at ‘d’.

See below right for more detail on the syntax of the For command.

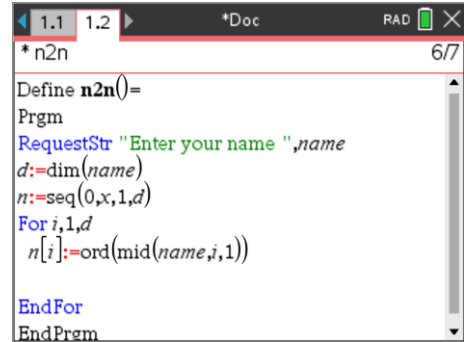


The first value in the list is equal to the numerical quantity aligned with the first letter of the ‘name’. To identify the first value in the list for ‘n’, use [ ] brackets.

Type:

`n[i]:=ord(mid(name,i,1))`

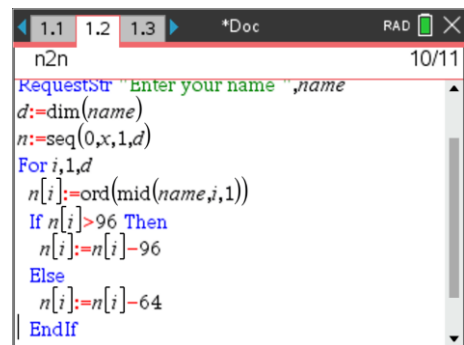
Recall from the previous questions how the ‘ord’ and ‘mid’ commands work.



The final step in the program is to make the letter values range between 1 & 26. This can be done using an **IF ... THEN ... ELSE** statement.

If the letter value is greater than 96, then it is lower case. The ASCII code for ‘a’ is 97. The ASCII code for ‘b’ is 98 and so on, subtracting 96 will align these values to 1 to 26 for a to z respectively.

The ASCII codes for upper case letters such as “A” start at 65, so we need to subtract 64.



Notice how lower case letters were checked first! Why?

The program is now ready to run. Press **Ctrl + R**, then **Enter** to execute.

**Question: 5.**

Write down your name as numbers from 1 to 26. [Note: On the calculator you can type: n]

**Question: 6.**

Write down the coordinates for the points that your polynomial must pass through: (1, #), (2, #) etc...

**Question: 7.**

What is smallest degree polynomial that will pass through all of your points? What assumptions do you need to make?

**Question: 8.**

Write down your polynomial as:  $f(x) = ax^n + bx^{n-1} + cx^{n-2} + \dots$  according to the degree of your polynomial. Using your points from Question 6, write down the simultaneous equations that need to be solved in order to determine the equation for your polynomial.



**Question: 9.**

Determine the solution to the simultaneous equations (Question 8) and hence write down the equation to your polynomial.

**Question: 10.**

Generate a table of values to show that your polynomial passes through the appropriate points.

**Question: 11.**

Determine the x and y axis intercepts of your polynomial.  
[Consult with your teacher if your personal polynomial does not have any x axis intercepts.]

**Exploring more Personal Polynomials****Question: 12.**

Hannah noticed that her polynomial had a smaller degree than she first thought. Explain why this is so.

**Question: 13.**

Arora and Mayam both noticed that Elle and Hannah's polynomials had a smaller degree than expected, but their polynomials were 'as expected'. Explain why this is the case.

**Question: 14.**

Amy joins the conversation and states that her name is unique. It too has a lower than expected degree, but for a different reason. Explain. Can you find another name that also has this property?

**Question: 15.**

Which of the following names would provide polynomials  $p(x)$  such that  $p(x) = 0$  has at least one solution:

- i) Daisy                  ii) Peter                  iii) Brian                  iv) Emma                  v) John

**Question: 16.**

Bindi notices that her polynomial  $p(x)$  has no solutions such that  $p(x) = 0$ . Find some other names that have no solutions to  $p(x) = 0$ . Explain how you searched and what criteria you used.

**Question: 17.**

According to their polynomials, where do Lilly and Lillian meet?

**Question: 18.**

Victor and his friend Victoria want to explore all the places where their polynomials meet. Without determining their equations:

- i) Identify six points where their polynomials will intersect.
- ii) Explain why their polynomials have seven points of intersection.

**Question: 19.**

Anna and Hannah claim to have very similar polynomials. Apply an appropriate transformation and compare the two polynomials.

**Question: 20.**

Are there any two names that share the same polynomial? Explain your reasoning.