Welcome to APS!

[aps]

Meeting 5 - 15/10/2019

Housekeeping

Thank you everyone who has paid their membership!

We now have a load of money and no idea what to do with it?

How should we spend it?

We want ideas

Or else we're gonna buy everyone a Rubber Duck

Bring stuff for testing!

Games students should be in the middle of building prototypes etc.

If you’ve got stuff that needs testing, bring it along, we’re happy to help!

Setup

Please find the 2 Websites below.

https://www.cemetech.net/projects/jstified/

https://tiroms.weebly.com

Ti84plus.rom

Task 1

Find documentation on programming a TI-84+ Calculator

Write a simple program that takes a user input.

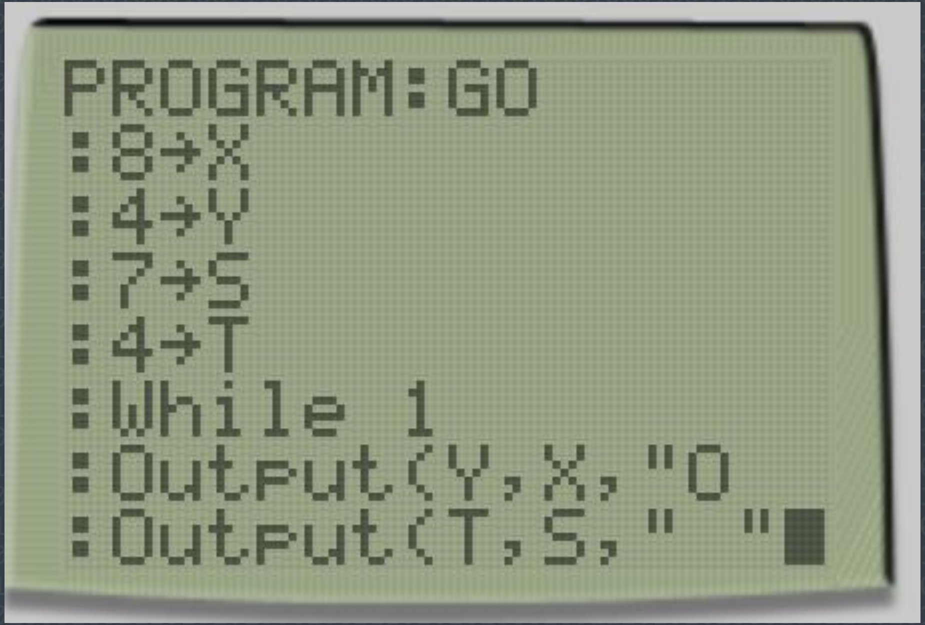
- Try a guessing Game

You have 10 Minutes!

Task 2

Let’s make a movable object.

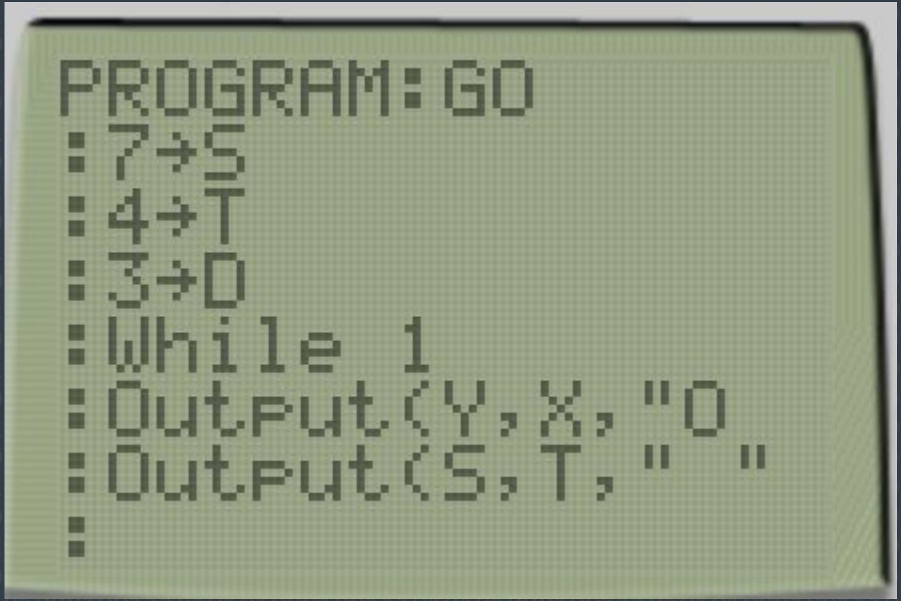
Task 2



```
PROGRAM:GO
:8→X
:4→Y
:7→S
:4→T
:While 1
:Output(Y,X,"O
:Output(T,S," "
```

Task 2

Insert a variable for direction.



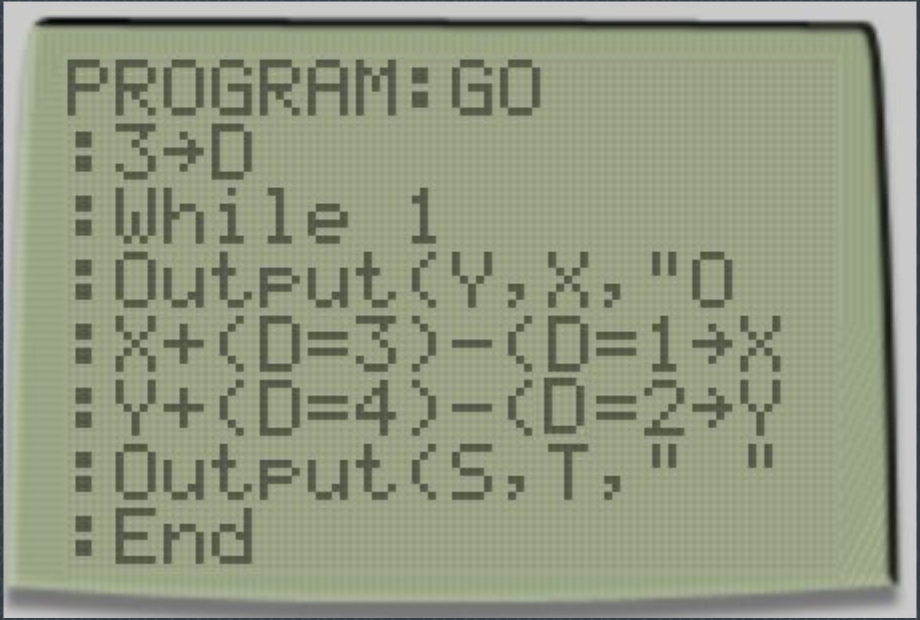
```
PROGRAM:GO
:7→S
:4→T
:3→D
:While 1
:Output(Y,X,"O
:Output(S,T," "
:
```

Task 2

Put in implicit conditionals for a direction variable.

http://tibasicdev.wikidot.com/sk:conditionals

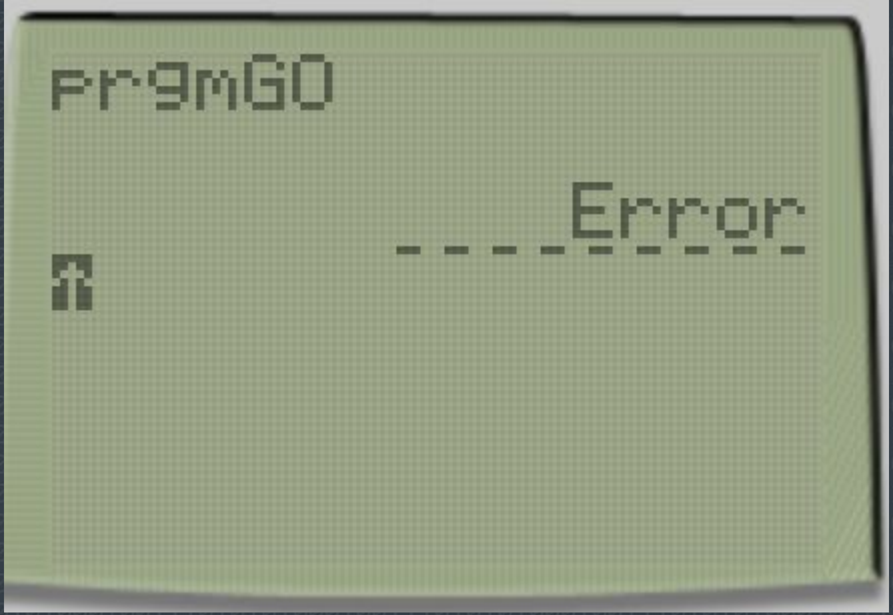
TEST for '='



```
PROGRAM:GO
:3→D
:While 1
:Output(Y,X,"O
:X+(D=3)-(D=1→X
:Y+(D=4)-(D=2→Y
:Output(S,T," "
:End
```

Task 2

It sucks!

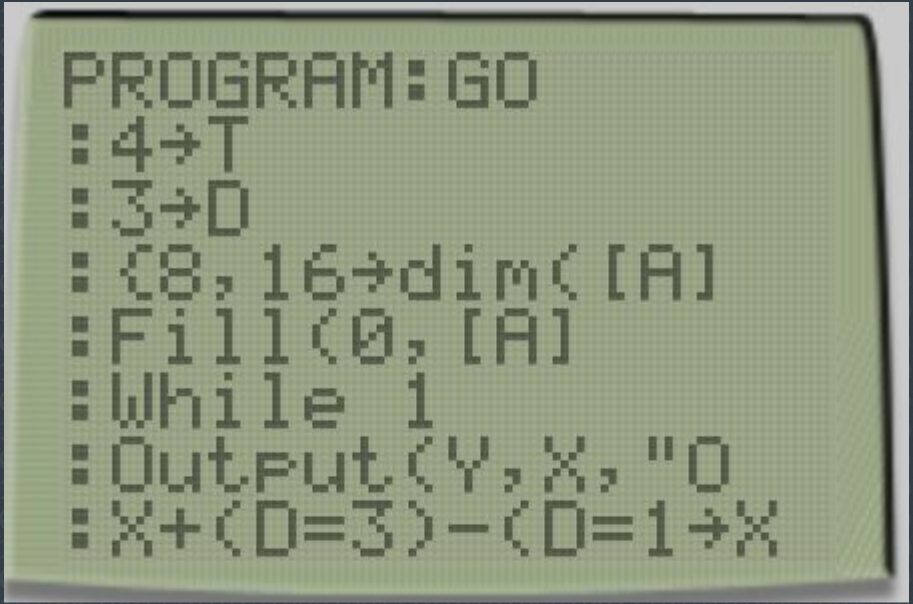


```
prgmGO
                Error
```

Task 2

Let’s make it a bit prettier.

Add a matrix that fills the matrix with 0s

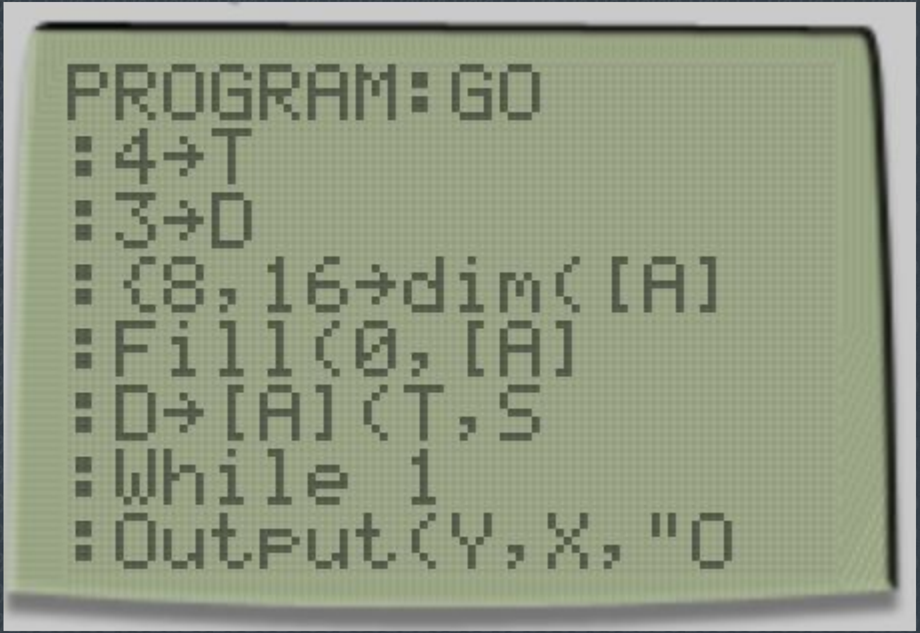


```
PROGRAM:GO
:4→T
:3→D
:{8,16→dim([A]
:Fill(0,[A]
:While 1
:Output(Y,X,"O
:X+(D=3)-(D=1→X
```

Task 2

And initialise the tail element and give it a direction to go in.

Remember S and T variables

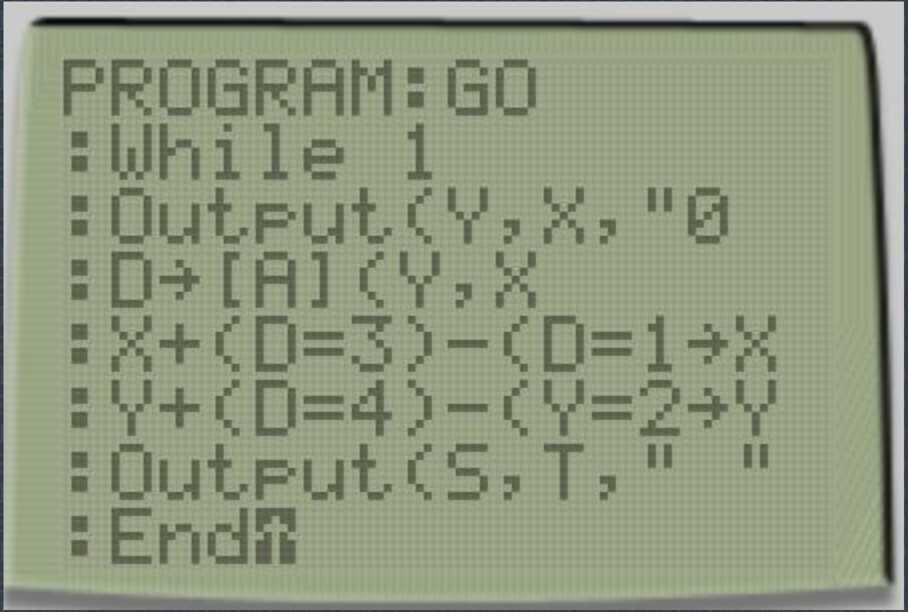


```
PROGRAM:GO
:4→T
:3→D
:{8,16→dim([A]
:Fill(0,[A]
:D→[A](T,S
:While 1
:Output(Y,X,"O
```

Task 2

Matrix will essentially store where our head is going at each point.

We can also store where it’s been.

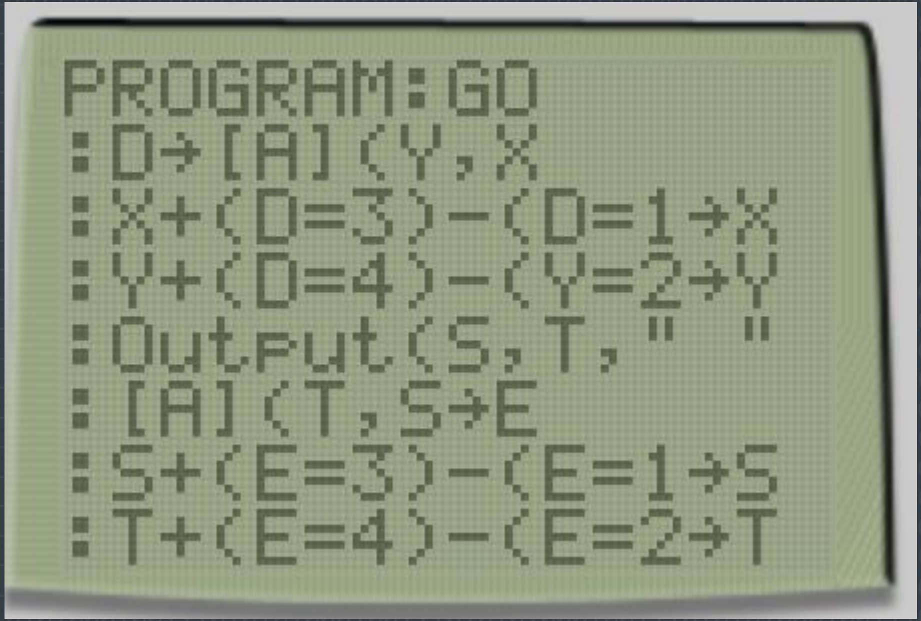


```
PROGRAM:GO
:While 1
:Output(Y,X,"0
:D→[A](Y,X
:X+(D=3)-(D=1→X
:Y+(D=4)-(Y=2→Y
:Output(S,T," "
:End
```

Task 2

Finally store the last position our Tail was in into a variable we’ll call E.

Use E to update S and T.

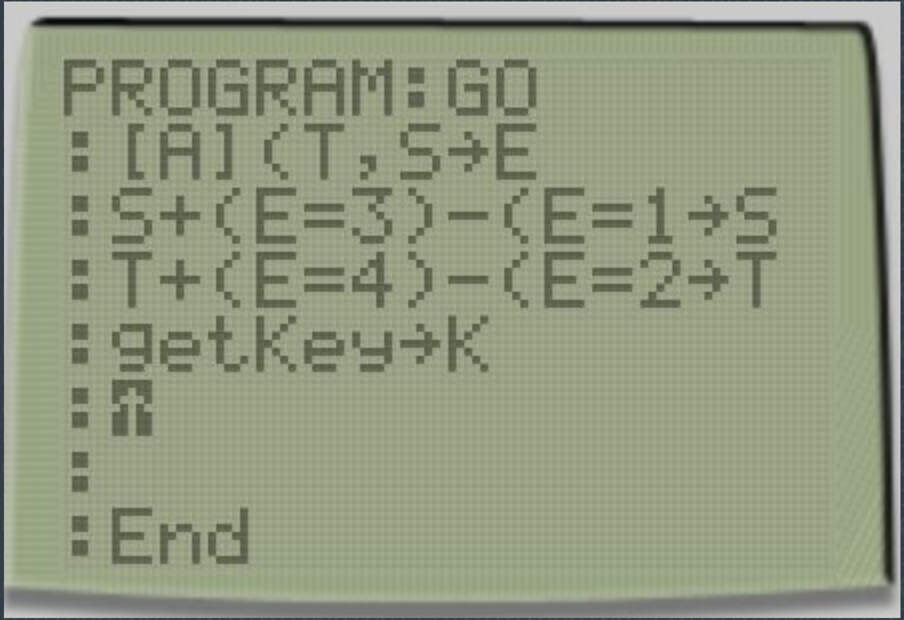


```
PROGRAM:GO
:D→[A](Y,X
:X+(D=3)-(D=1→X
:Y+(D=4)-(Y=2→Y
:Output(S,T," "
:[A](T,S→E
:S+(E=3)-(E=1→S
:T+(E=4)-(E=2→T
```

Task 2

Let’s finish up by moving the object.

Use getKey function.

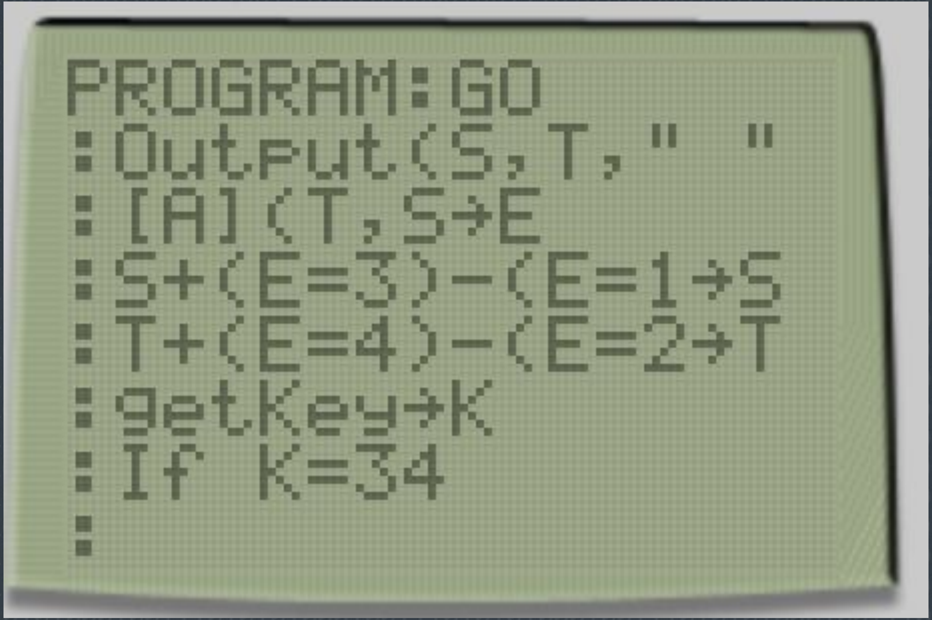


```
PROGRAM:GO
:[A](T,S→E
:S+(E=3)-(E=1→S
:T+(E=4)-(E=2→T
:getKey→K
:
:
:End
```

Task 2

We need to check if a directional key is pressed (If Statement)

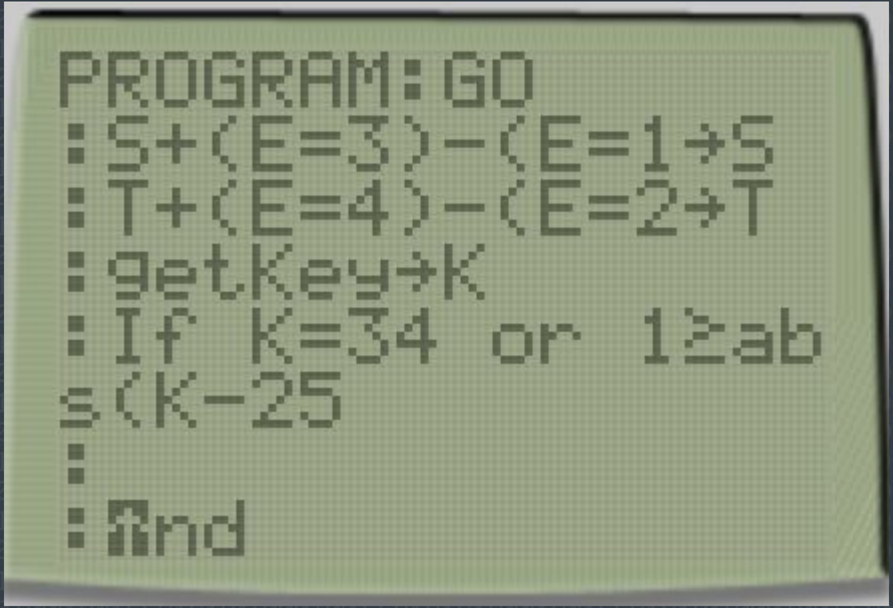
34 is the down button.



```
PROGRAM:GO
:Output(S,T," "
:[A](T,S→E
:S+(E=3)-(E=1→S
:T+(E=4)-(E=2→T
:getKey→K
:If K=34
:
```

Task 2

Now we’re going to cheat a little. The other directional keys correspond to 24,25 and 26 so we can check with a bit of maths.

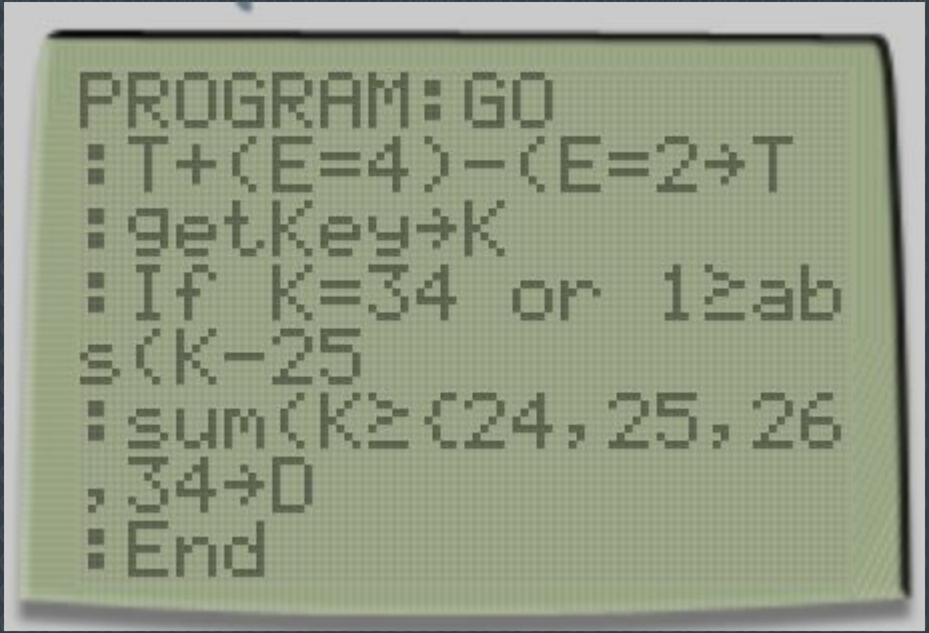


```
PROGRAM:GO
:S+(E=3)-(E=1→S
:T+(E=4)-(E=2→T
:getKey→K
:If K=34 or 1≥ab
s(K-25
:
:End
```

Task 2

Now to map the direction keys. We do this by finding how many keys are greater than K, which we store into D

List menu under STAT



```
PROGRAM:GO
:T+(E=4)-(E=2→T
:getKey→K
:If K=34 or 1≥ab
s(K-25
:sum(K≥{24,25,26
,34→D
:End
```

Task 3

Change the movement functions keys to correspond to numbers 1,2,3 and 5 for easier control on the keyboard.

IE - Re-write the IF statement.

Task 4

Change the movement functions – eg:

```
X+(D=3)-(D=1→X
```

To allow the screen to wrap.

Tips

Your screen is 16x8