addition(Z)

Add top numbers from stack, subtract where Z = -1

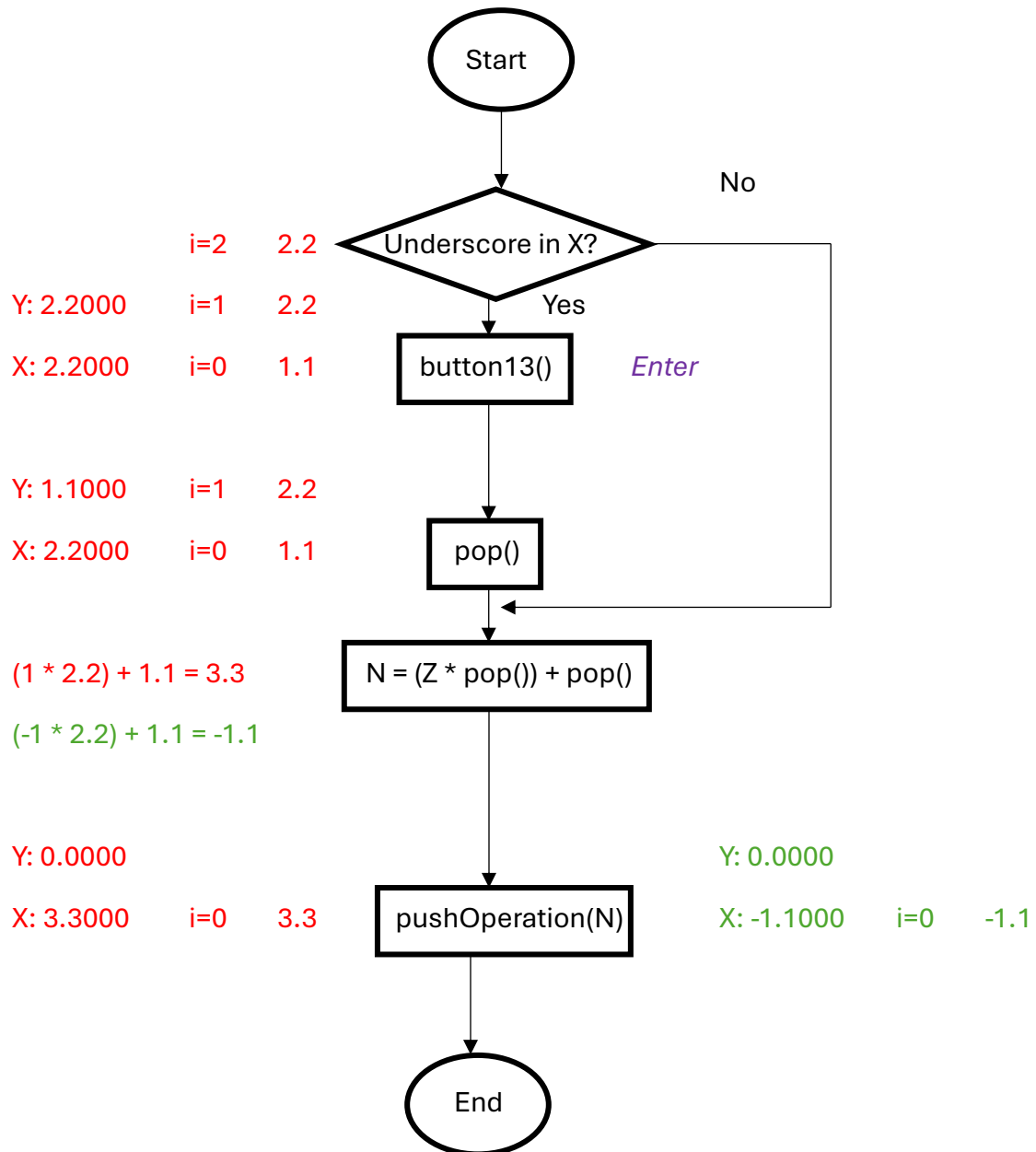Ex.   Z = 1   (Addition)   Ex.   Z = -1   (Subtraction)

Y: 1.1000   Y = 1.1000

X: 2.2_   X = 2.2000

Start

i=2   2.2   **Underscore in X?**   No

Y: 2.2000   i=1   2.2   Yes

X: 2.2000   i=0   1.1   **button13()**   *Enter*

Y: 1.1000   i=1   2.2

X: 2.2000   i=0   1.1   **pop()**

(1 * 2.2) + 1.1 = 3.3   **N = (Z * pop()) + pop()**

(-1 * 2.2) + 1.1 = -1.1

Y: 0.0000   Y: 0.0000

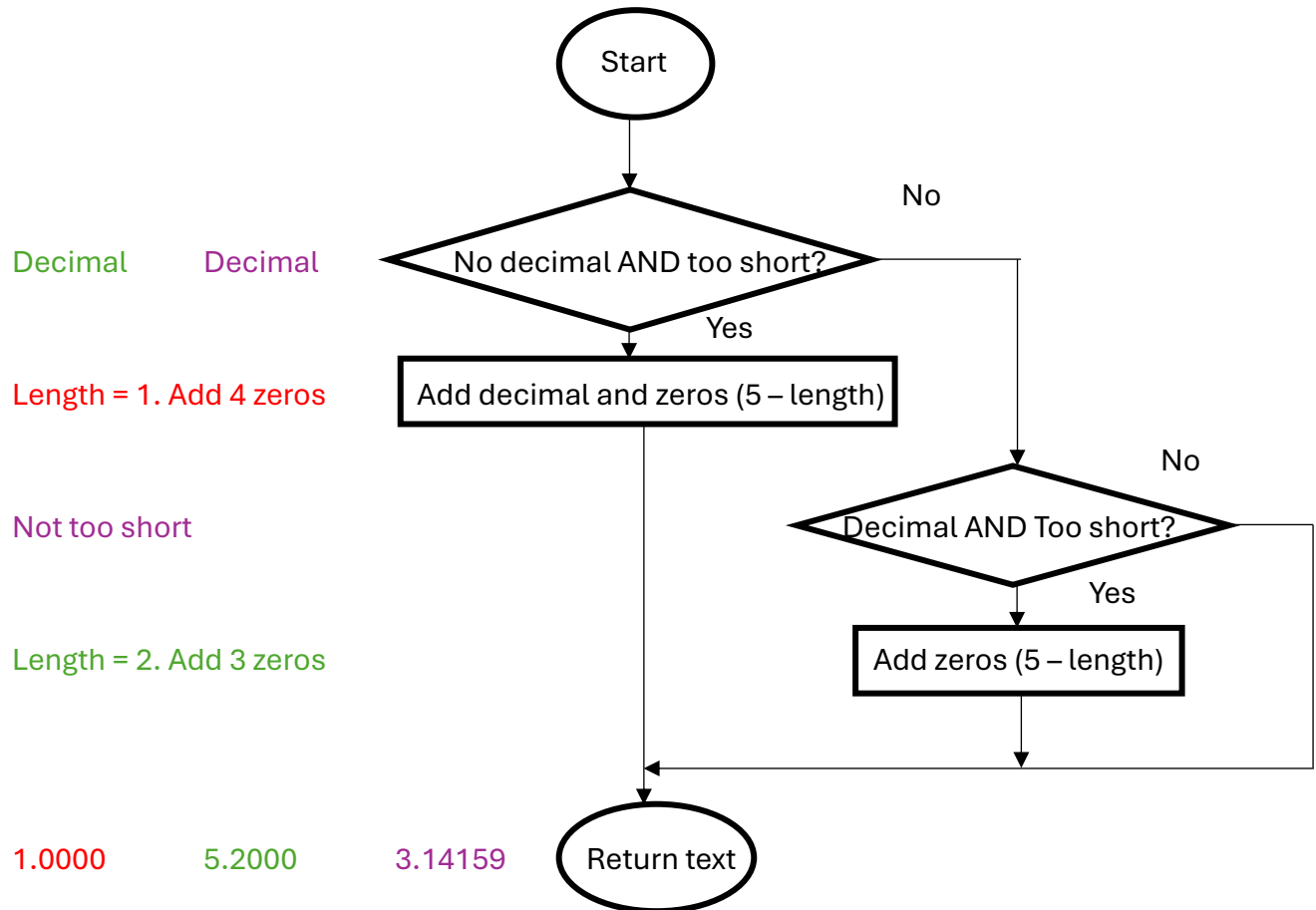X: 3.3000   i=0   3.3   **pushOperation(N)**   X: -1.1000   i=0   -1.1

End

appendZeros(text)

    Format text to add zeros

        Ex. "1" -> "1.0000"

        Ex. "5.2" -> "5.2000"

        Ex. "3.14159" -> "3.14159"

Start

No

Decimal   Decimal   No decimal AND too short?

Yes

Length = 1. Add 4 zeros   Add decimal and zeros (5 – length)

No

Not too short   Decimal AND Too short?

Yes

Length = 2. Add 3 zeros   Add zeros (5 – length)

1.0000   5.2000   3.14159   Return text
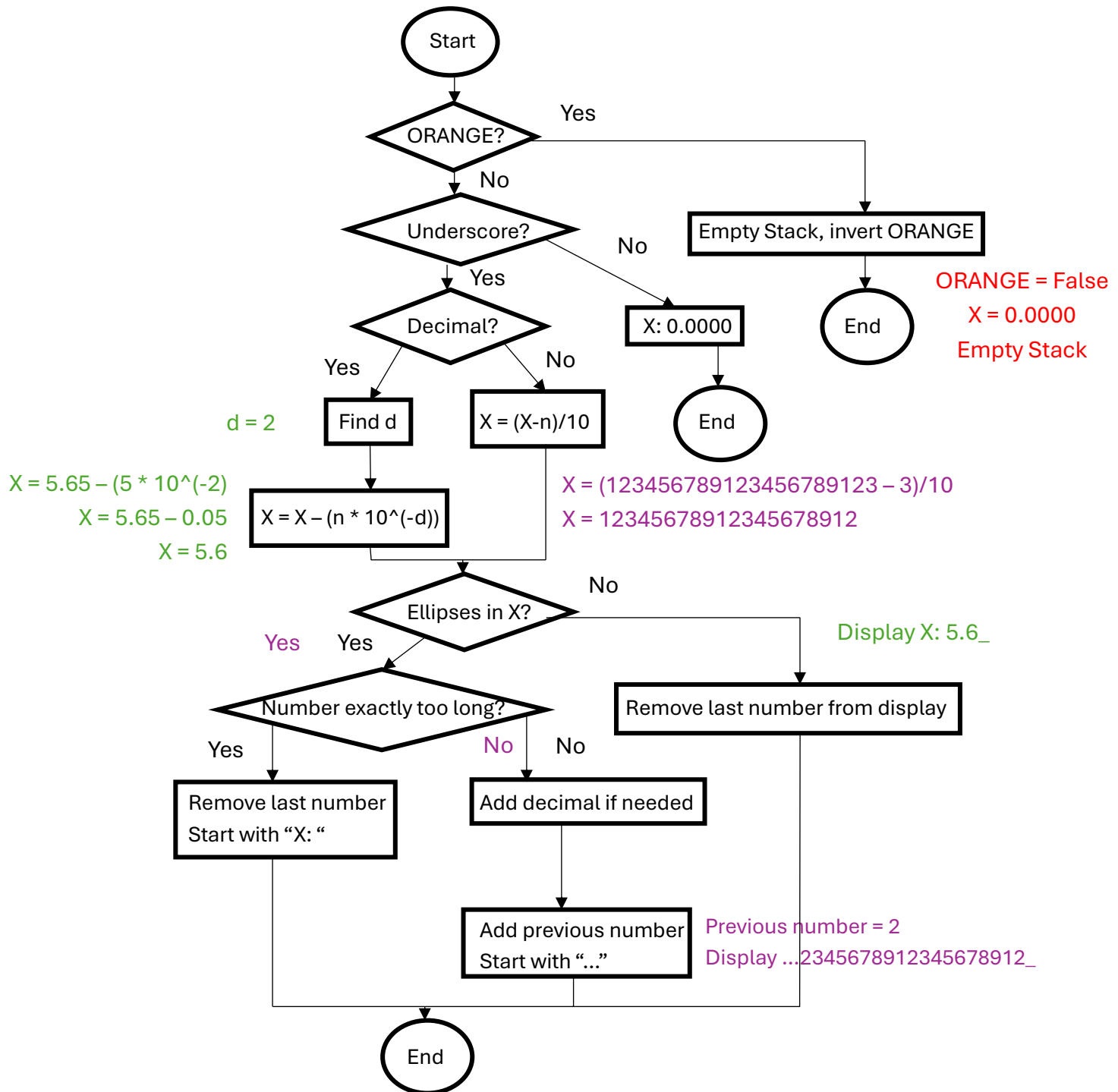
button17()

Backspace/Clear button

Let n be the number being removed and d the distance between the underscore and period

Ex. ORANGE = True, X: 5.0000, X = 5   Ex. ORANGE = False, Displaying …3456789123456789123_

Ex. ORANGE = False, Display X: 5.65_, X = 5.65   X = 123456789123456789123

Start

ORANGE? — Yes →

No ↓

Underscore? — No →

Yes ↓

Decimal?

Yes ↓   No →

d = 2   **Find d**   **X = (X-n)/10**   **X: 0.0000**

Empty Stack, invert ORANGE

ORANGE = False
X = 0.0000
Empty Stack

End

$X = 5.65 - (5 * 10^{-2})$
$X = 5.65 - 0.05$
$X = 5.6$

X = X – (n * 10^(-d))

$X = (123456789123456789123 – 3)/10$
$X = 12345678912345678912$

End

Ellipses in X? — No →

Yes
Yes ↓

Number exactly too long?

Yes ↓   No
No →

Remove last number from display

Display X: 5.6_

Remove last number Start with "X: "

Add decimal if needed

Add previous number Start with "…"

Previous number = 2
Display …2345678912345678912_

End

button22()

"Divide" button

Ex.   i=1   0   Y: 2.0000   Ex.   Y: 9.0000

        i=0   2   X: 0.0000   i=0   9   X: 3_
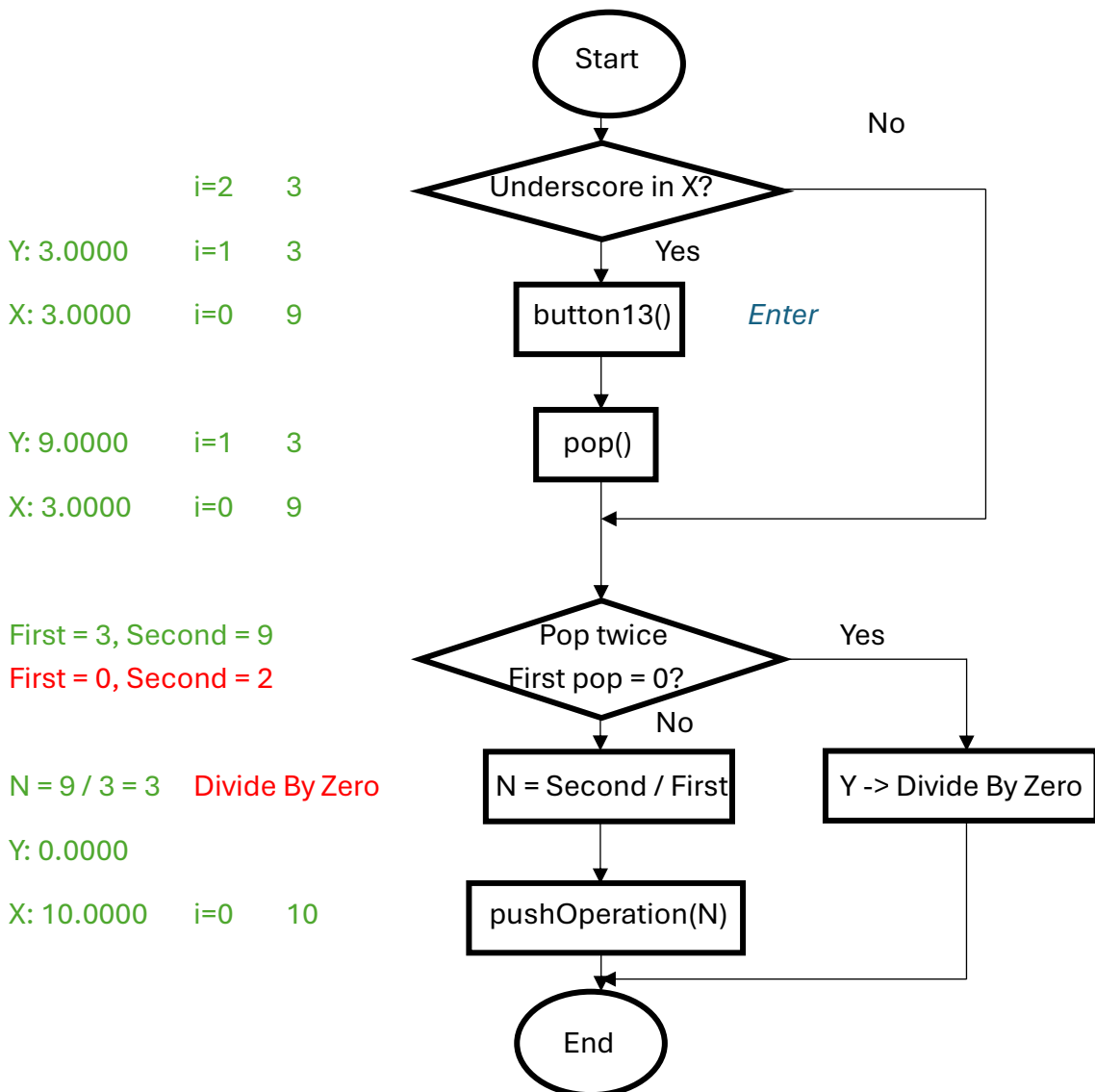
Outcome:

                      Divide By Zero   Y: 0.0000

                      X: 0.0000   i=0   3   X: 3.0000

```
                        ┌─────────┐
                        │  Start  │
                        └─────────┘
                             │
        i=2     3                                    No
Y: 3.0000  i=1  3     < Underscore in X? >──────────────┐
X: 3.0000  i=0  9            │ Yes                       │
                        ┌──────────┐                     │
                        │ button13()│   Enter            │
                        └──────────┘                     │
                             │                           │
Y: 9.0000  i=1  3      ┌──────────┐                      │
X: 3.0000  i=0  9      │  pop()   │                      │
                       └──────────┘                      │
                             │◄───────────────────────────
                             │
First = 3, Second = 9      Pop twice          Yes
First = 0, Second = 2   < First pop = 0? >──────────┐
                             │ No                    │
N = 9 / 3 = 3   Divide By Zero                       │
                    ┌────────────────┐   ┌───────────────────┐
Y: 0.0000           │ N = Second/First│   │ Y -> Divide By Zero│
                    └────────────────┘   └───────────────────┘
X: 10.0000  i=0  10        │                       │
                    ┌──────────────────┐           │
                    │ pushOperation(N) │           │
                    └──────────────────┘           │
                             │◄───────────────────────
                        ┌─────────┐
                        │   End   │
                        └─────────┘
```

button27()

"Multiply" button. Calls button13() and pop() to add the current number (if any) once to the stack. Pops twice from the stack and multiplies. Calls pushOperation(number) to update the stack and display.

Ex.   i=1   5   Y: 2.0000   Ex.   Y: 3.0000

       i=0   2   X: 5.0000   i=0   3   X: 4_

Outcome:

                   Y: 0.0000   Y: 0.0000

       i=0   10   X: 10.0000   i=0   12   X: 12.0000

Start

Underscore in X?   No

i=2   4

Y: 4.0000   i=1   4   Yes

X: 4.0000   i=0   3   button13()   Enter

Y: 3.0000   i=1   4   pop()

X: 4.0000   i=0   3

N = 5 * 2 = 10   N = 4 * 3 = 12   N = pop() * pop()

Y: 0.0000   Y: 0.0000

X: 10.0000   i=0   10   pushOperation(N)   X: 12.0000   i=0   12

End
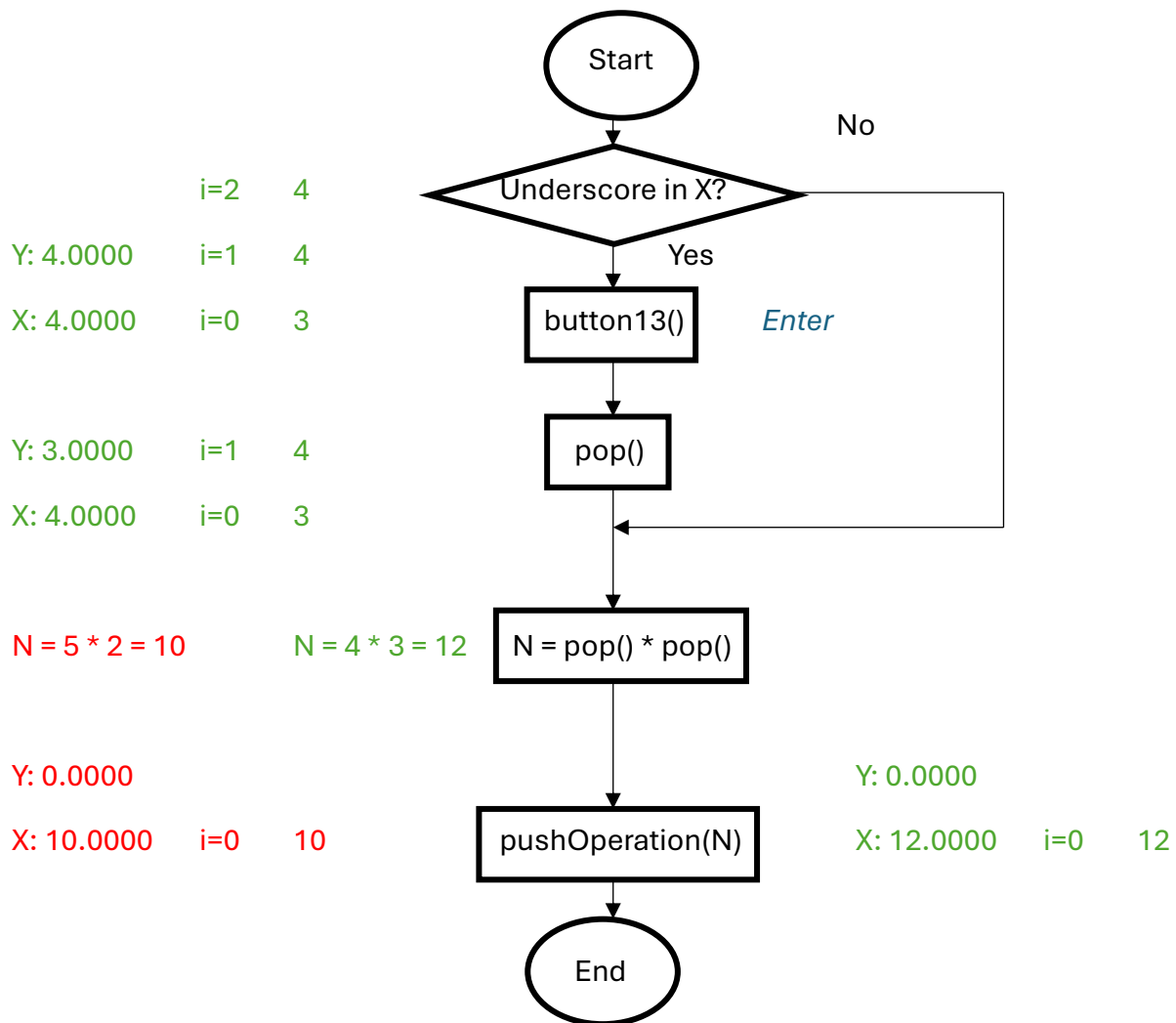
button35()

Decimal button. Does nothing if a number is being input and already has a decimal place. Starts a new number at "0._" if no number is being input. Otherwise, adds "._" to the display and scrolls if necessary.

Ex.   X: 0.02_   Ex.   X: 5.0000   Ex.   ...1234567891234567891_

            numX = 0.02   numX = 5   numX = 1234567891234567891

Outcome:

            X: 0.02_   X: 0._   ...234567891234567891._

            numX = 0.02   numX = 0   numX = 1234567891234567891

Start

Yes

No underscore   No decimal   Underscore AND Decimal

End

No

Underscore?   No

...1234567891234567891   Sub = Remove underscore   X: 0_   X: 0_

Len = 23   Len(X: number) < 16   numX = 0   numX = 0

Yes   No

Sub + "._"   "..." + Sub[4:] + "._"
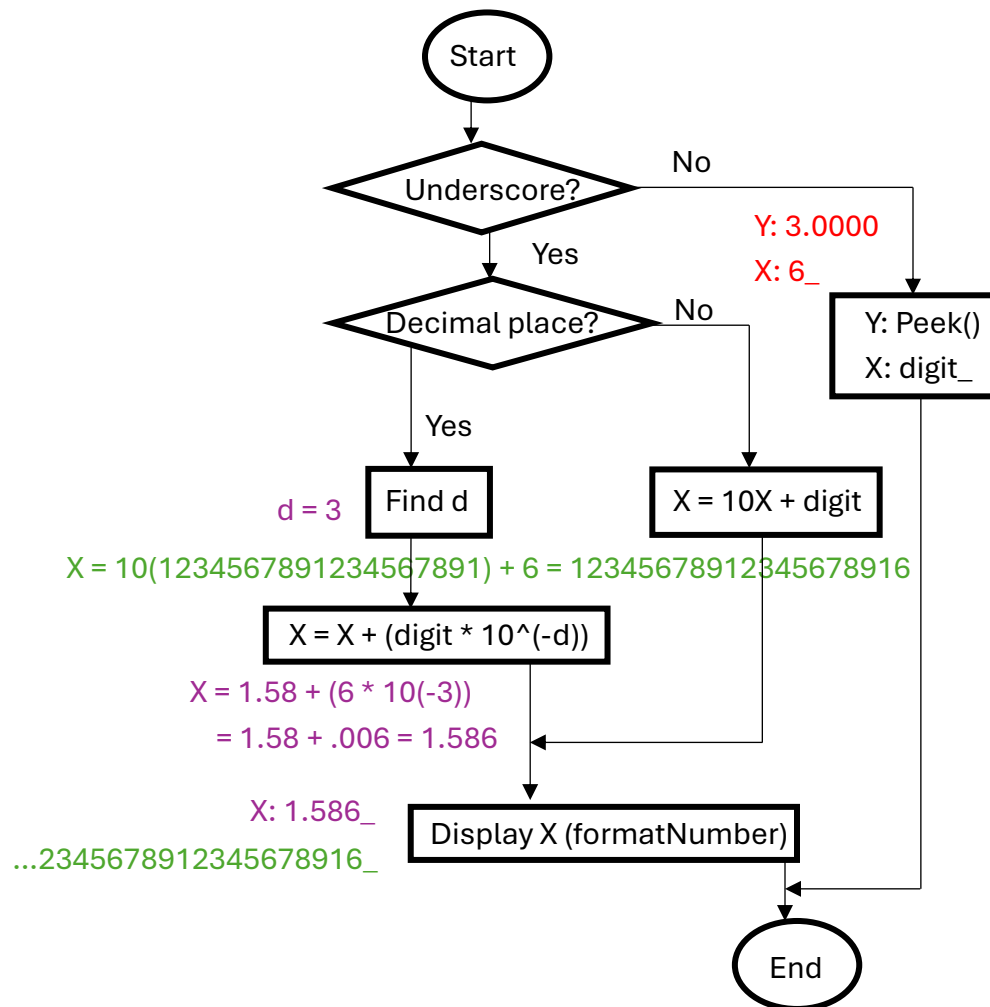
...234567891234567891._

End
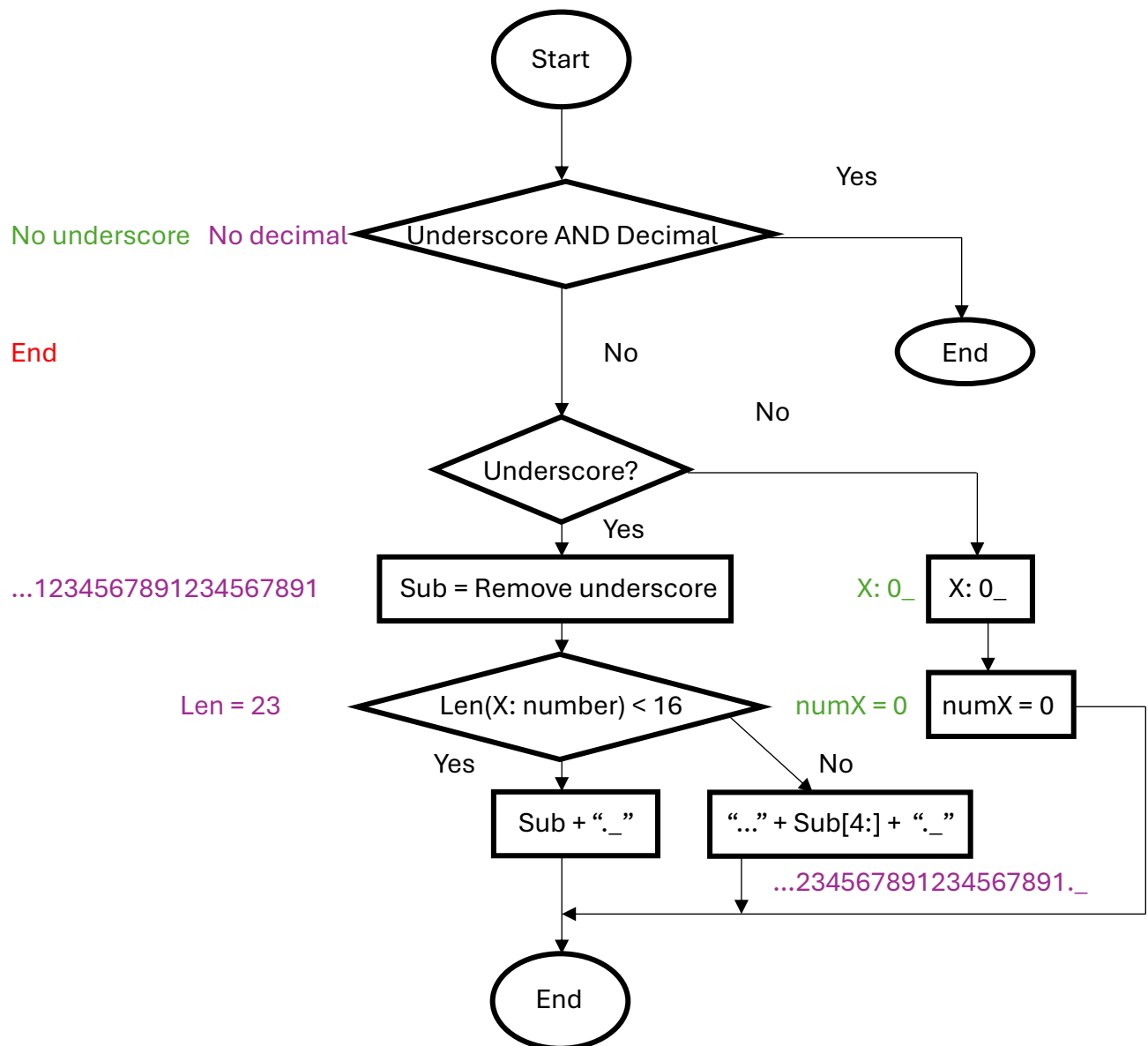
numberPressed(digit)...

Given a digit 0-9, update X and the display.

Let d be the distance between "." And "_" . Assume digit = 6.

Ex.   i = 0   3   X: 3.0000   Ex.   X = 1234567891234567891

Ex.   X: 1.58_   X = 1.58   ...1234567891234567891_

```
                          ┌─────────┐
                          │  Start  │
                          └─────────┘
                               │
                               ▼
                          ◇ Underscore? ◇ ──No──┐
                               │                │
                              Yes               │
                               ▼          Y: 3.0000
                        ◇ Decimal place? ◇ ──No──┐    X: 6_
                               │                 │
                              Yes                │    ┌──────────────┐
                               ▼                 │    │  Y: Peek()   │
   d = 3                 ┌──────────┐            │    │  X: digit_   │
                         │  Find d  │            │    └──────────────┘
                         └──────────┘       ┌──────────────────┐     │
                               │            │ X = 10X + digit  │     │
                               │            └──────────────────┘     │
                    ┌─────────────────────────────┐                  │
                    │ X = X + (digit * 10^(-d))   │                  │
                    └─────────────────────────────┘                  │
                               │                                     │
                               ▼                                     │
                    ┌─────────────────────────────┐                 │
                    │  Display X (formatNumber)   │                 │
                    └─────────────────────────────┘                 │
                               │                                     │
                               ▼                                     │
                          ┌─────────┐                                │
                          │   End   │◄───────────────────────────────┘
                          └─────────┘
```

d = 3

X = 10(1234567891234567891) + 6 = 12345678912345678916

X = X + (digit * 10^(-d))

X = 1.58 + (6 * 10(-3))
= 1.58 + .006 = 1.586

X: 1.586_

...2345678912345678916_

pushOperation(number)

Format display so

Y: Peek()   (Ex. 12345678912345678912345)

X: Number (Ex. 1)

Start

12345678912345678912345

Temp = Peek()

Temp too long?   No

Yes

12345678912345678912

Shorten Temp

appendZeros(Temp)

Y: 12345678912345678912

Display Y: Temp

1

Temp = number

Temp too long?   No

Yes

Shorten Temp

1.0000

appendZeros(Temp)

X: 1.0000

Display X: Temp

Stack

i=1   1

Push number to Stack

i=0   12345678912345678912345

End