C++ Vectors and Structs

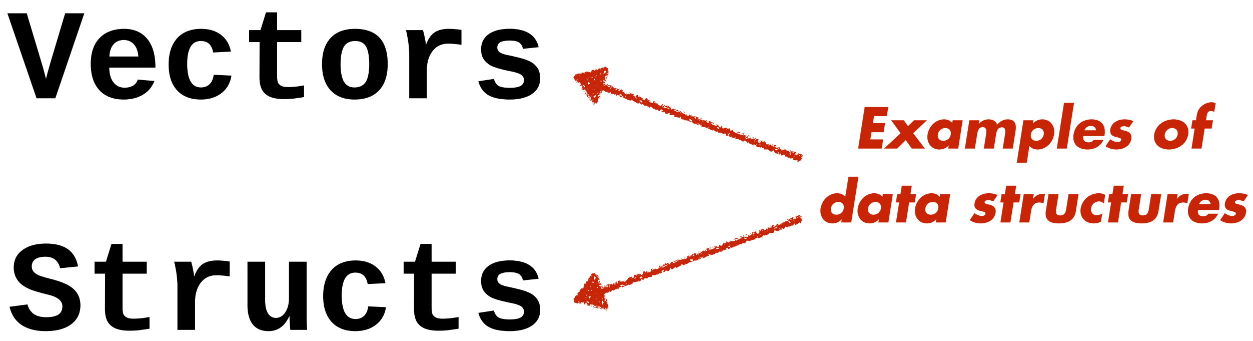
Robotics 102

Introduction to AI and Programming
University of Michigan and Berea College
Fall 2021

Wall following will require Vectors and Structs

Data Structures

A data structure organizes how data is stored and retrieved by a program



`Vectors`

`Structs`

Examples of data structures

Data Structures

We need:	Because variables alone are:
`Vectors`	*Not big enough*
`Structs`	*Not organized enough*

Consider a variable to be like a parking spot

Consider a variable to be like a parking spot

variable

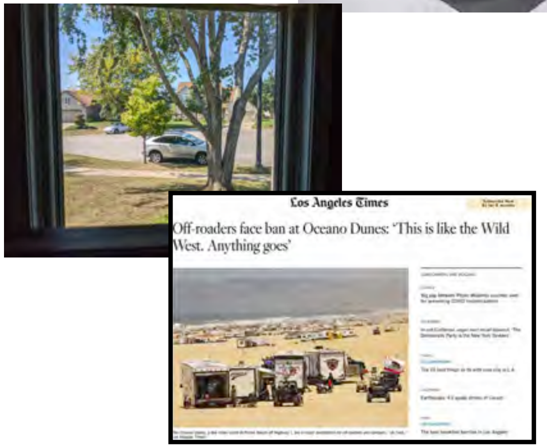
variable

An individual variable stores a single element of information
(as a basic data type)

Los Angeles Times

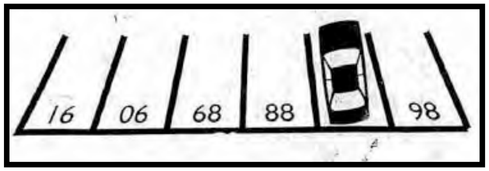
Off-roaders face ban at Oceano Dunes: 'This is like the Wild West. Anything goes'

variable



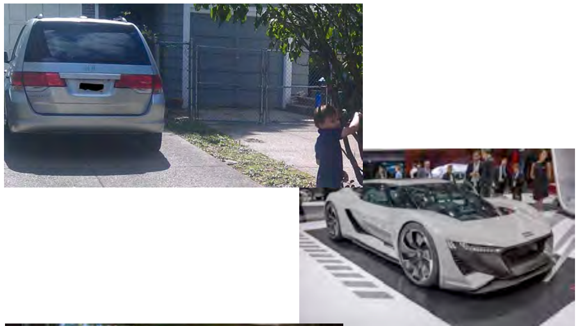
Los Angeles Times

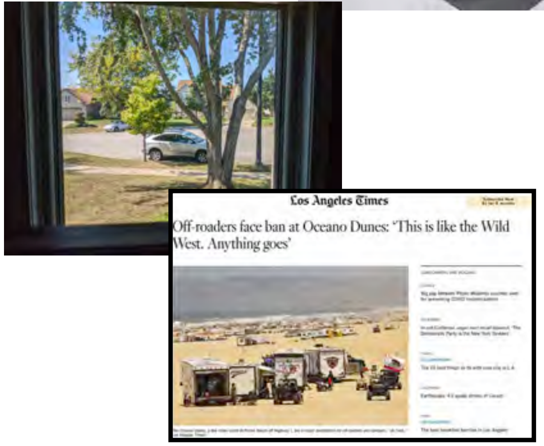
Off-roaders face ban at Oceano Dunes: 'This is like the Wild West. Anything goes'



variable

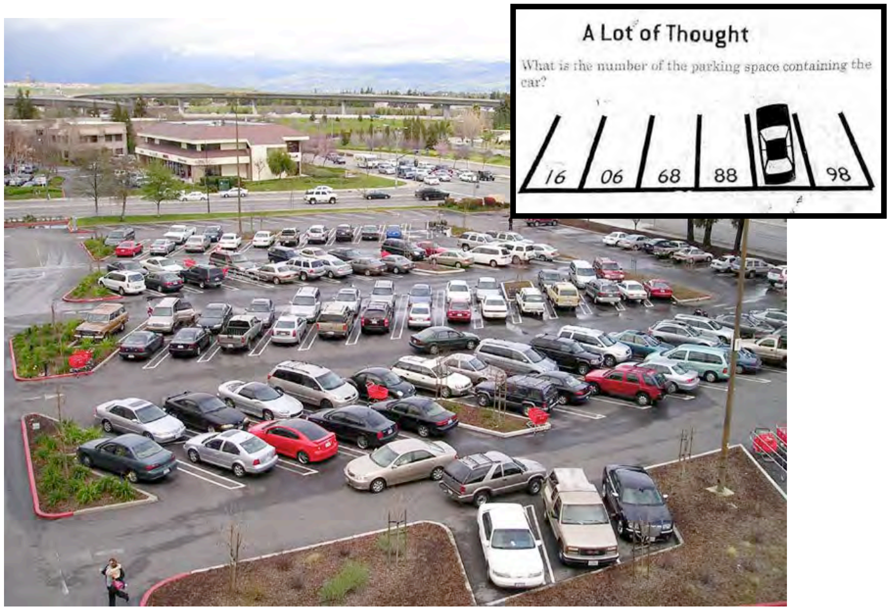
vector

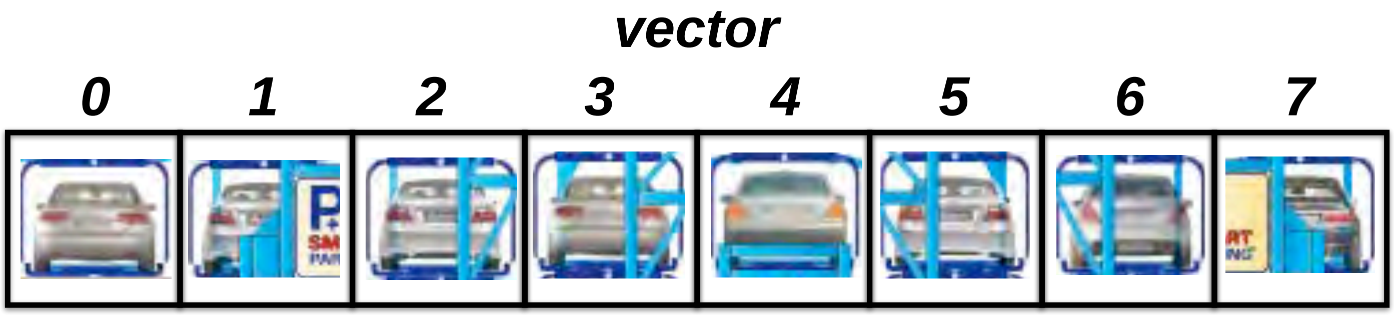




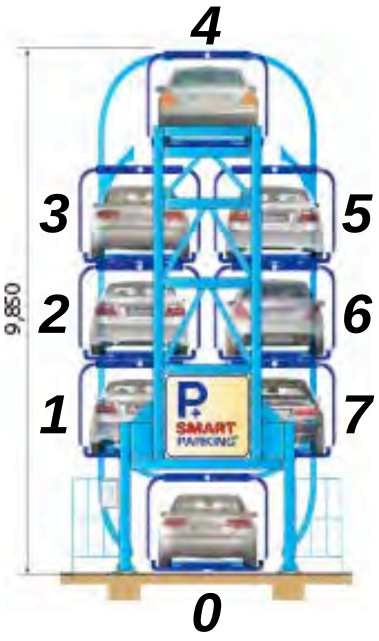
Off-roaders face ban at Oceano Dunes: 'This is like the Wild West. Anything goes'

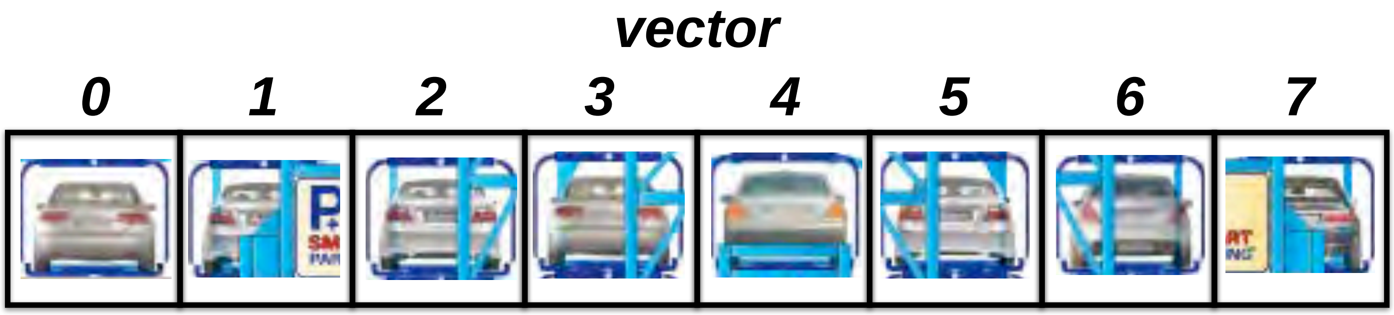
A vector stores a sequence of elements.
A vector is an abstract data type.

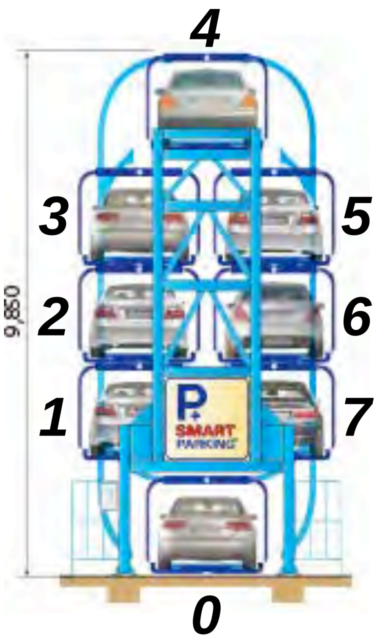




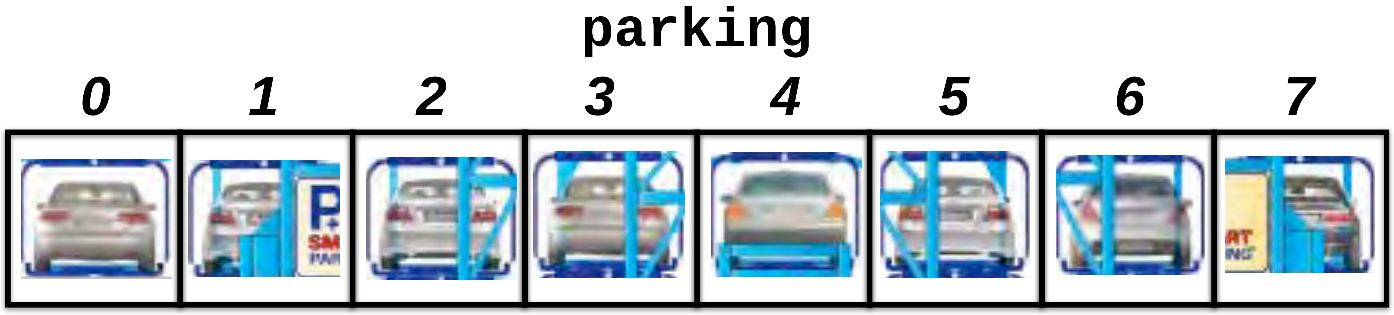
A vector stores a sequence of elements.



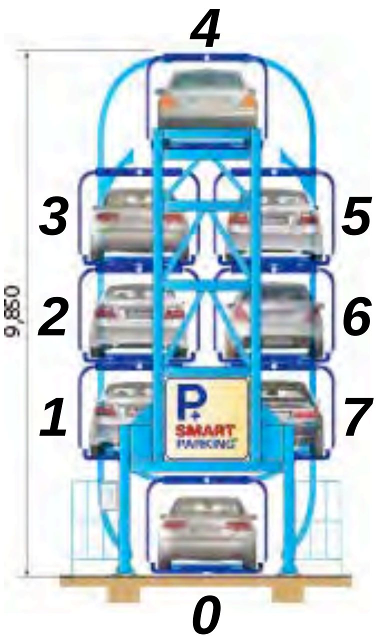




How can we refer to this car?



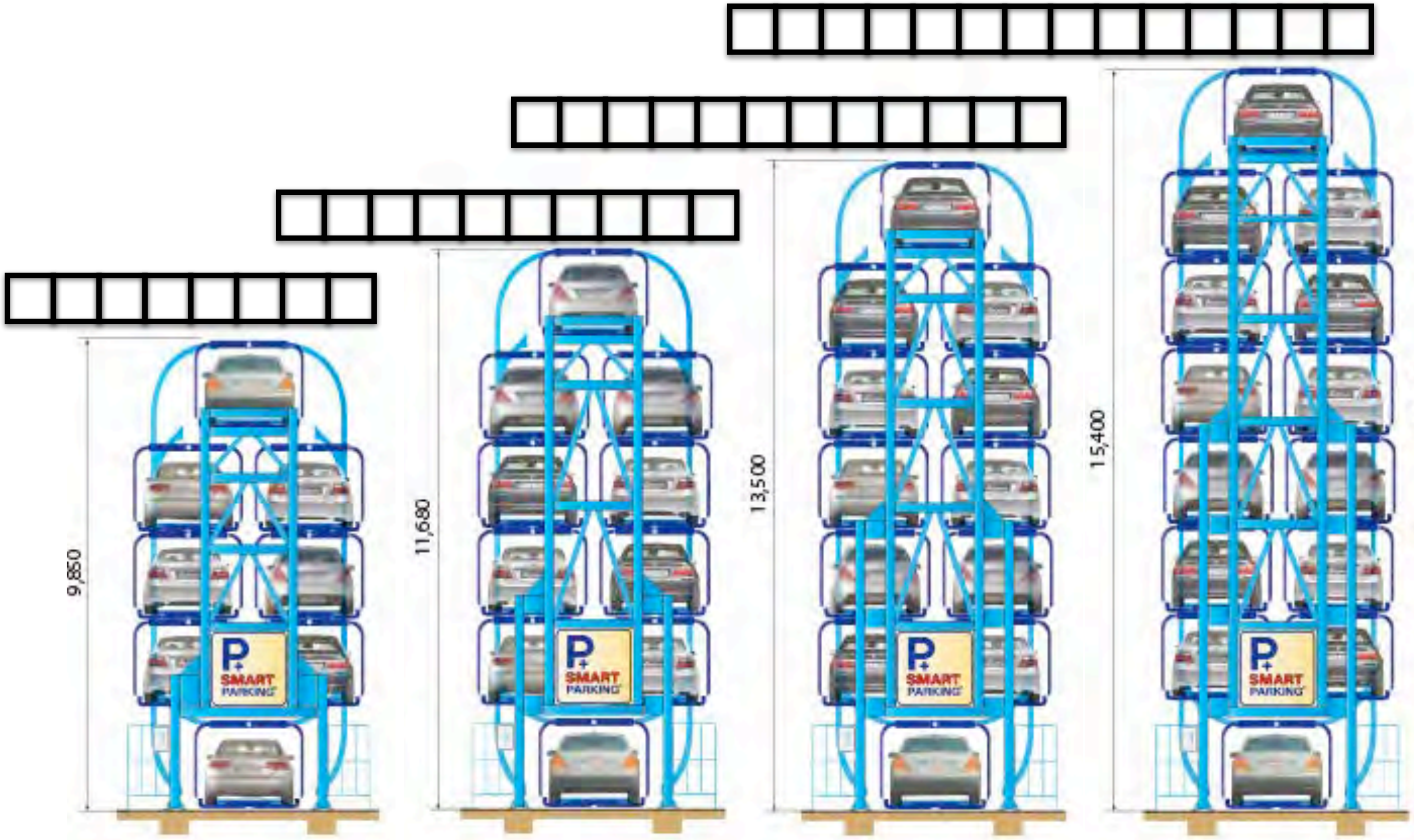
Parking System

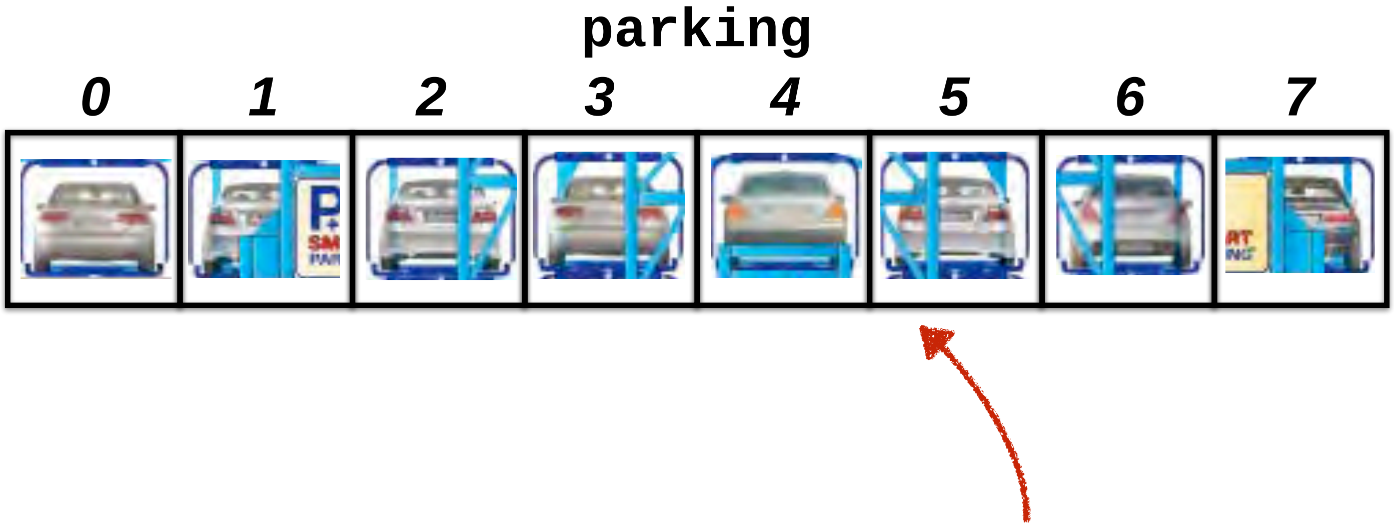


A vector can be *indexed into* for referring to a specific element.

`parking[5]` refers to the sixth element in this vector

A vector can grow or shrink as the number elements increases or decreases.





`parking`

0 *1* *2* *3* *4* *5* *6* *7*

How do we describe this car computationally?

2013 Ford Fusion

A structure (or *struct*) defines
larger and compositional concepts.

A struct is composed of elements
that describe properties.

A struct is a user defined data that can
compose elements of any data type.

struct car

model	*make*	*year*	*licence*

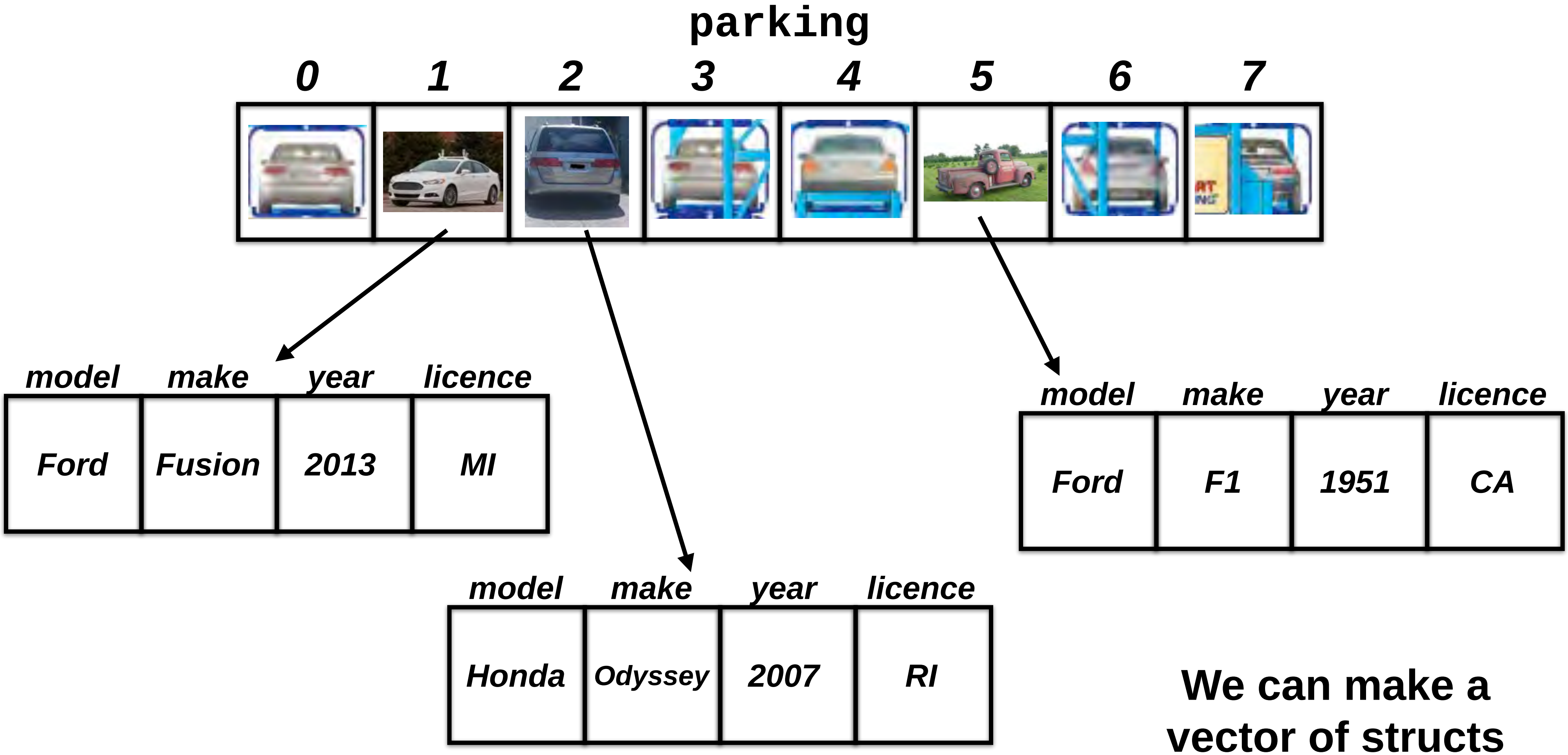
2013 Ford Fusion

Possible instances of *struct car*

model	*make*	*year*	*licence*
Ford	***Fusion***	***2013***	***MI***

model	*make*	*year*	*licence*
Honda	**Odyssey**	***2007***	***RI***

model	*make*	*year*	*licence*
Ford	***F1***	***1951***	***CA***



`parking`

0	1	2	3	4	5	6	7

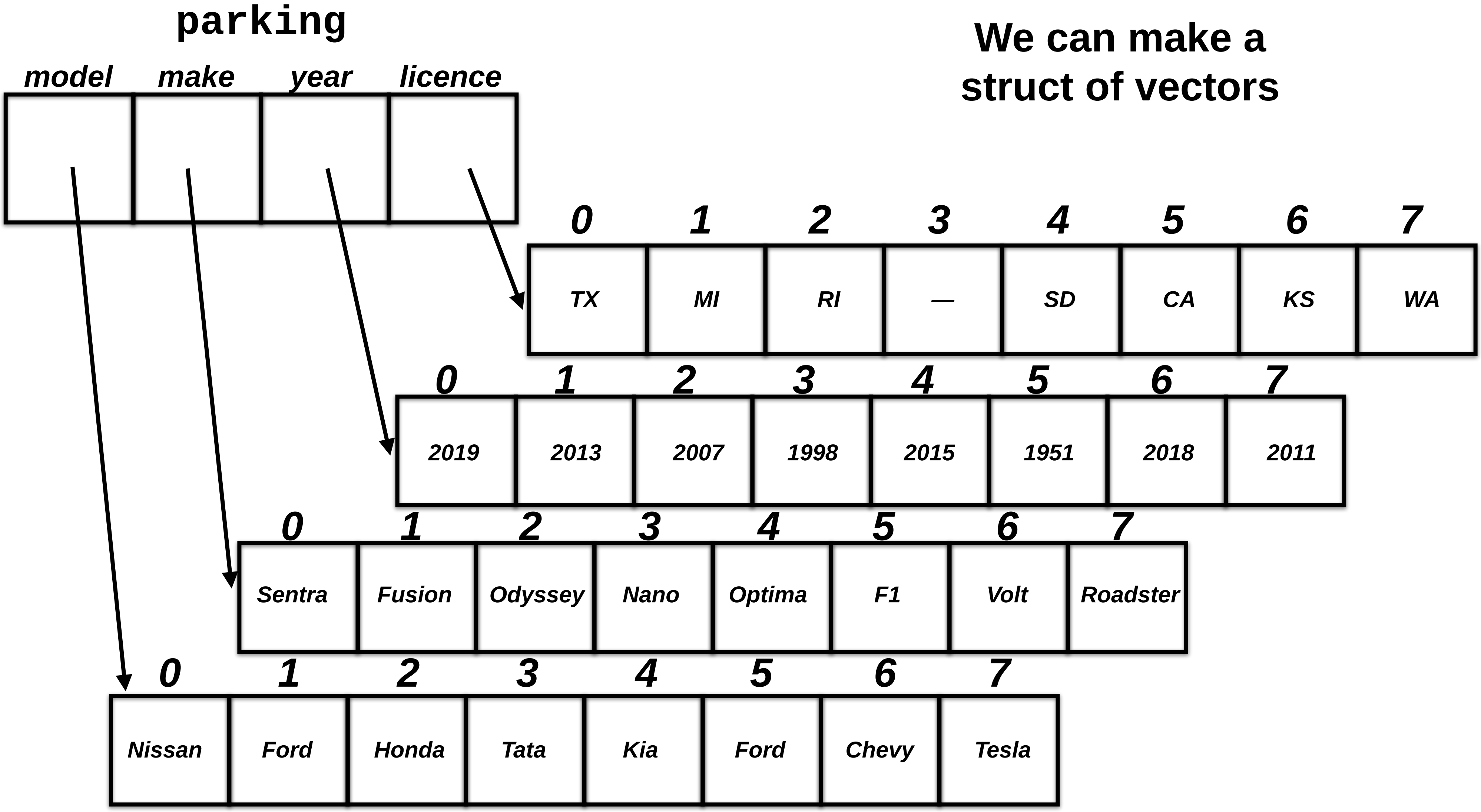
model	make	year	licence
Ford	Fusion	2013	MI

model	make	year	licence
Honda	Odyssey	2007	RI

model	make	year	licence
Ford	F1	1951	CA

We can make a vector of structs or…

We can make a struct of vectors



`parking`

0	1	2	3	4	5	6	7
TX	MI	RI	—	SD	CA	KS	WA

0	1	2	3	4	5	6	7
2019	2013	2007	1998	2015	1951	2018	2011

0	1	2	3	4	5	6	7
Sentra	Fusion	Odyssey	Nano	Optima	F1	Volt	Roadster

0	1	2	3	4	5	6	7
Nissan	Ford	Honda	Tata	Kia	Ford	Chevy	Tesla

Vectors

Structs

Done

hello

```
Hello World!
Chad is in Robotics 102
```

calculator (Version 24)

```
Please type a number and press enter: 22
Please type another number and press enter: 7
What is 22 plus 7? 29
What is 22 minus 7 ? 15
What is 22 times 7 ? 154
What is 22 divided by 7 ? 3.14286
```

calculator (Version 41)

calculator (Version 54)

```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
3*4+8 = 20
Please type an operation (one of: + - * / q): -
Please type a number and press enter: 10
3*4+8-10 = 10
Please type an operation (one of: + - * / q): /
Please type a number and press enter: 5
3*4+8-10/5 = 2
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 51
3*4+8-10/5*51 = 102
Please type an operation (one of: + - * / q): q
```

☑ Program Structure

☑ Compile/Execute

☑ Operators

☑ Data Types

☑ Variables

☑ User Input/Output

☑ Functions

☑ Branching

☑ Iterators

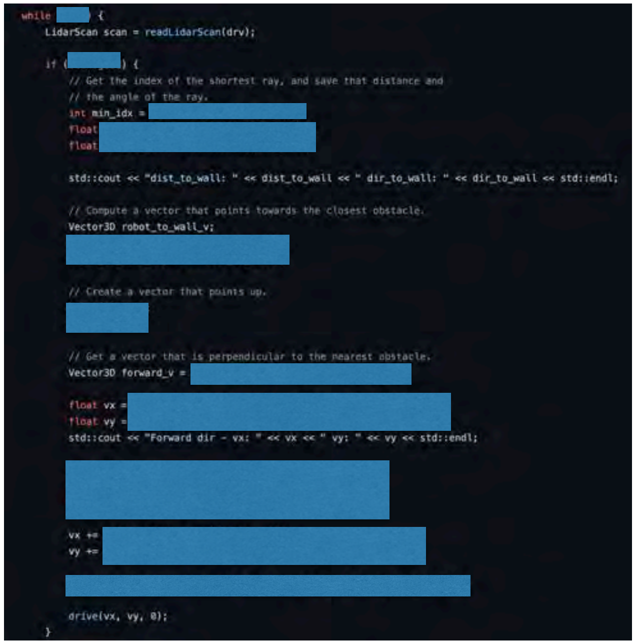
☐ Vectors

☐ Structs

☐ File Input/Output

Now

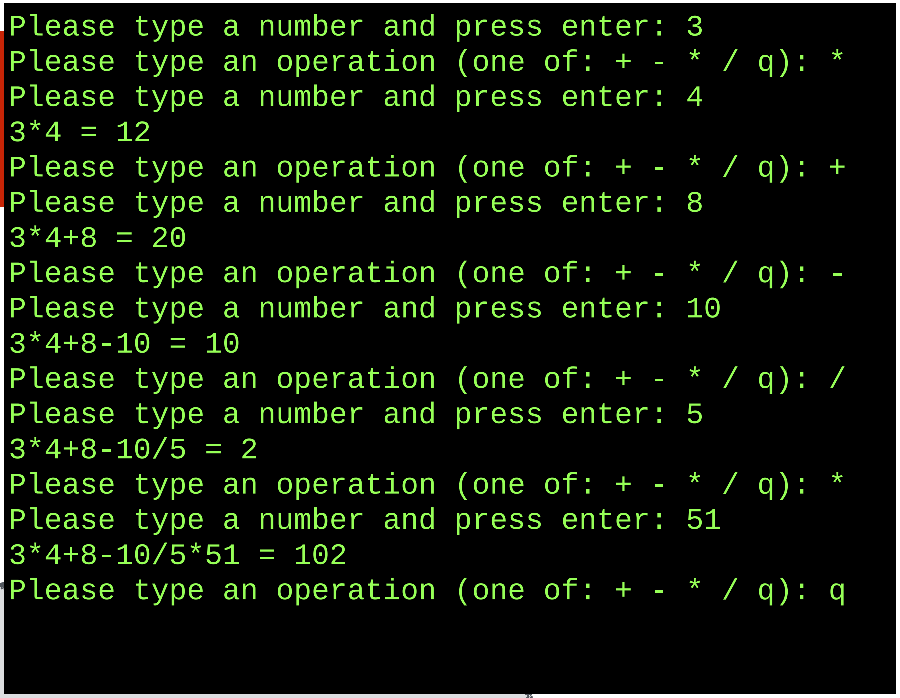
wall_follower.cpp - Project 1



```
while [redacted] {
    LidarScan scan = readLidarScan(drv);

    if ([redacted]) {
        // Get the index of the shortest ray, and save that distance and
        // the angle of the ray.
        int min_idx = [redacted]
        float [redacted]
        float

        std::cout << "dist_to_wall: " << dist_to_wall << " dir_to_wall: " << dir_to_wall << std::endl;

        // Compute a vector that points towards the closest obstacle.
        Vector3D robot_to_wall_v;
        [redacted]

        // Create a vector that points up.
        [redacted]

        // Get a vector that is perpendicular to the nearest obstacle.
        Vector3D forward_v = [redacted]

        float vx = [redacted]
        float vy =
        std::cout << "Forward dir - vx: " << vx << " vy: " << vy << std::endl;

        [redacted]

        vx += [redacted]
        vy +=

        [redacted]

        drive(vx, vy, 0);
    }
```

Our calculator is not done yet

Our calculator is



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
3*4+8 = 20
Please type an operation (one of: + - * / q): -
Please type a number and press enter: 10
3*4+8-10 = 10
Please type an operation (one of: + - * / q): /
Please type a number and press enter: 5
3*4+8-10/5 = 2
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 51
3*4+8-10/5*51 = 102
Please type an operation (one of: + - * / q): q
```

calculator.cpp (Version 54) - Condensed

`main()`

```
getNumber(myNumber);
getOperator(myOperator);
while (myOperator != 'q') {
   getNumber(myOtherNumber);
   performOperation(myNumber,myOperator,myOtherNumber,resultNumber);
   outputResult(myNumber,myOperator,myOtherNumber,resultNumber);
   myNumber = resultNumber;
   getOperator(myOperator);
}
```

Our calculator is not done yet

Can we keep a history of operations?

Can we undo the last operation?

This `3*4+8-10/5*51 = 102` does not look right

This `3*4+8-10/5*51 = 102` does not look right

Improper mathematical notation

This `3*4+8-10/5*51 = 102` does not look right

Improper mathematical notation

3*4+8-10/5*51 =

12+8-10/5*51 =

12+8-2*51 =

12+8-102 =

20-102 =

-82

This `3*4+8-10/5*51 = 102` does not look right

Improper mathematical notation

3*4+8-10/5*51 =

12+8-10/5*51 =

12+8-2*51 =

12+8-102 =

20-102 =

-82

pemdas.cpp

```
#include <iostream>
int main() {

   std::cout << 3*4+8-10/5*51 << "\n";

}
```

Trust but verify

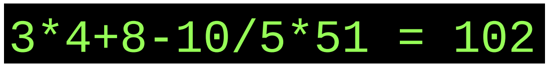
```
-82
```

This `3*4+8-10/5*51 = 102` does not look right

Improper mathematical notation

We did not cover how to produce this output

3*4+8-10/5*51 = -82



This `3*4+8-10/5*51 = 102` does not look right

Improper mathematical notation

3*4+8-10/5*51 = -82

We did not cover how to produce this output

The left side of the equation could be updated as a string data type

"3*4+8-10/5*51"

What is a C++ string data type ?

A `string` variable is a sequence of characters

Each element of a string has data type `char`

"3*4+8-10/5*51" represented like

'3'	'*'	'4'	'+'	'8'	'-'	'1'	'0'	'/'	'5'	'*'	'5'	'1'

Remember our first program

What is a C++ string data type ?

A `string` variable is a sequence of characters

Each element of a string has data type `char`

hello00.cpp

```
#include <iostream>
int main()
{
   std::cout << "Hello World!";
}
```

What is a C++ string data type ?

A string variable is a sequence of characters

Each element of a string has data type char

hello10.cpp

```
#include <iostream>
#include <string>

int main()
{
   std::string hello = "Hello World!";
   std::cout << hello;
}
```

What is a C++ string data type ?

A `string` variable is a sequence of characters

Each element of a string has data type `char`

hello10.cpp

```
#include <iostream>
#include <string>

int main()
{
   std::string hello = "Hello World!";
   std::cout << hello;
}
```

```
Hello World!
```

What is a C++ string data type ?

The + operator is “overloaded” to concatenate strings

hello11.cpp

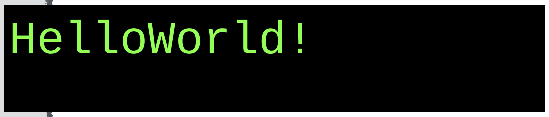
```
#include <iostream>
#include <string>

int main()
{
   std::string hello = "Hello";
   std::string world = "World!";
   std::cout << hello + world;
}
```

What is a C++ string data type ?

The + operator is “overloaded” to concatenate strings

hello11.cpp

```
#include <iostream>
#include <string>

int main()
{
   std::string hello = "Hello";
   std::string world = "World!";
   std::cout << hello + world;
}
```



```
HelloWorld!
```

What is a C++ string data type ?

The + operator cannot concatenate strings with numbers

hello12.cpp

```
#include <iostream>
#include <string>

int main()
{
   std::string hello = "Hello";
   float onezerotwo = 10.2;
   std::cout << hello + onezerotwo;
}
```

What is a C++ string data type ?

```
hello12.cpp:8:23: error: invalid operands to binary expression ('std::string' (aka
      'basic_string<char, char_traits<char>, allocator<char> >') and 'float')
   std::cout << hello + onezerotwo;
                ~~~~~ ^ ~~~~~~~~~~
```

hello12.cpp

```
#include <iostream>
#include <string>

int main()
{
   std::string hello = "Hello";
   float onezerotwo = 10.2;
   std::cout << hello + onezerotwo;
}
```

What is a C++ string data type ?

The function `std::to_string` can convert numbers to strings, but…

hello13.cpp

```
#include <iostream>
#include <string>

int main()
{
   std::string hello = "Hello";
   float onezerotwo = 10.2;
   std::cout << hello + std::to_string(onezerotwo);
}
```

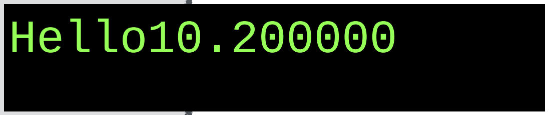
What is a C++ string data type ?

The function `std::to_string` can convert numbers to strings, but…

hello13.cpp

```
#include <iostream>
#include <string>

int main()
{
   std::string hello = "Hello";
   float onezerotwo = 10.2;
   std::cout << hello + std::to_string(onezerotwo);
}
```

it will not look right



```
Hello10.200000
```

This `3*4+8-10/5*51 = 102` does not look right

Improper mathematical notation

3*4+8-10/5*51 = -82

We did not cover how to produce this output

Another idea:

Store operands and operators in vectors

calculator.cpp (Version 61)

```
/*
   Let's write an infix calculator program for real numbers with variables
      that takes numbers from user input, uses functions for modularity,
      performs calculations with infinitely many consecutive operations,
      and stores the entire mathematical expression in vectors
*/

#include <iostream>
#include <vector>  // this enables the program to use C++ Vector data types

// Global variables to store all operands and operators
// Note: we should really try to avoid using global variables
std::vector <float> allOperands;  // vector of all operands entered by user
std::vector <char> allOperators;  // vector of all operators entered by user
```

C++ vector library supports vector data types

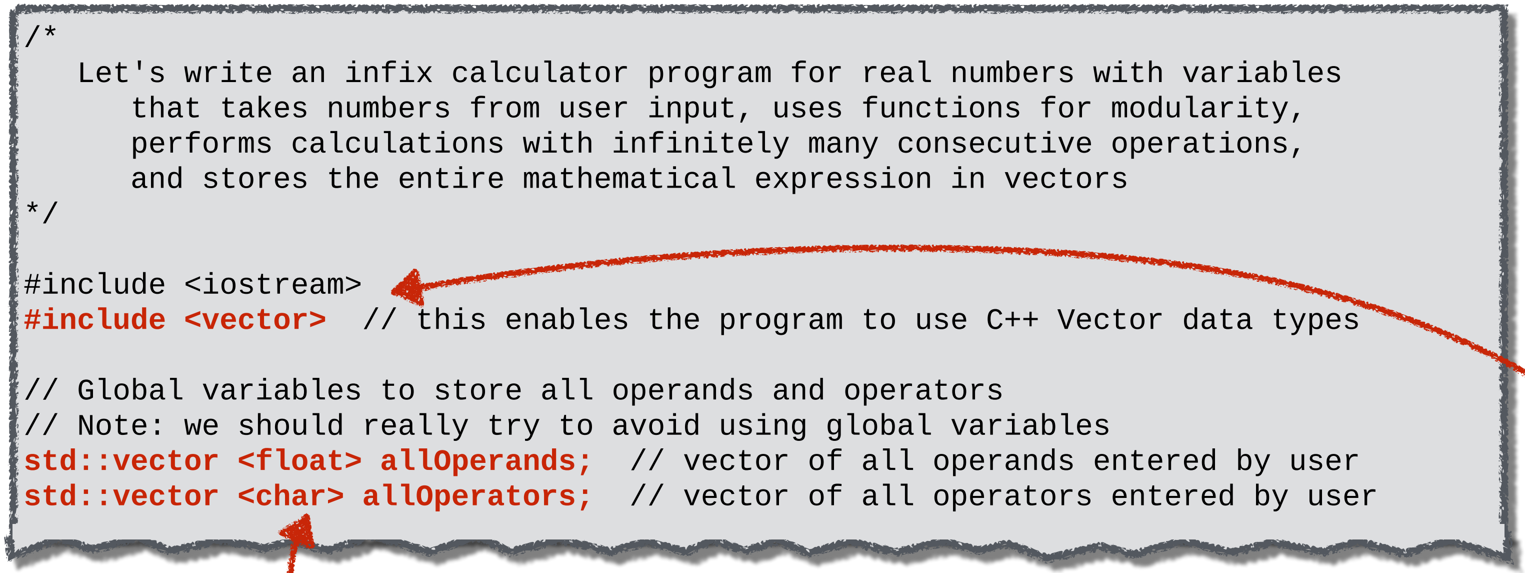
calculator.cpp (Version 61)

```
/*
   Let's write an infix calculator program for real numbers with variables
      that takes numbers from user input, uses functions for modularity,
      performs calculations with infinitely many consecutive operations,
      and stores the entire mathematical expression in vectors
*/

#include <iostream>
#include <vector>  // this enables the program to use C++ Vector data types

// Global variables to store all operands and operators
// Note: we should really try to avoid using global variables
std::vector <float> allOperands;  // vector of all operands entered by user
std::vector <char> allOperators;  // vector of all operators entered by user
```

C++ vector library supports vector data types

Declarations for two vectors, one of `float` data type and the other of `char` data type

calculator.cpp (Version 61)



```
/*
   Let's write an infix calculator program for real numbers with variables
      that takes numbers from user input, uses functions for modularity,
      performs calculations with infinitely many consecutive operations,
      and stores the entire mathematical expression in vectors
*/

#include <iostream>
#include <vector>  // this enables the program to use C++ Vector data types

// Global variables to store all operands and operators
// Note: we should really try to avoid using global variables
std::vector <float> allOperands;  // vector of all operands entered by user
std::vector <char> allOperators;  // vector of all operators entered by user
```

C++ vector library supports vector data types

Declarations for two vectors, one of `float` data type and the other of `char` data type

A vector can be created for any defined data type

calculator.cpp (Version 61)

```cpp
/*
   Let's write an infix calculator program for real numbers with variables
      that takes numbers from user input, uses functions for modularity,
      performs calculations with infinitely many consecutive operations,
      and stores the entire mathematical expression in vectors
*/

#include <iostream>
#include <vector>  // this enables the program to use C++ Vector data types

// Global variables to store all operands and operators
// Note: we should really try to avoid using global variables
std::vector <float> allOperands;  // vector of all operands entered by user
std::vector <char> allOperators;  // vector of all operators entered by user
```

Functions

```
   addTwoNumbers()
subtractTwoNumbers()
multiplyTwoNumbers()
  divideTwoNumbers()
       getNumber()
    getOperation()
  performOperation()
    outputResult()
            main()
```

Note: these vectors are not declared within a function.
Thus, they are global variables.

Global variables are defined for all functions across the program.

calculator.cpp (Version 61)

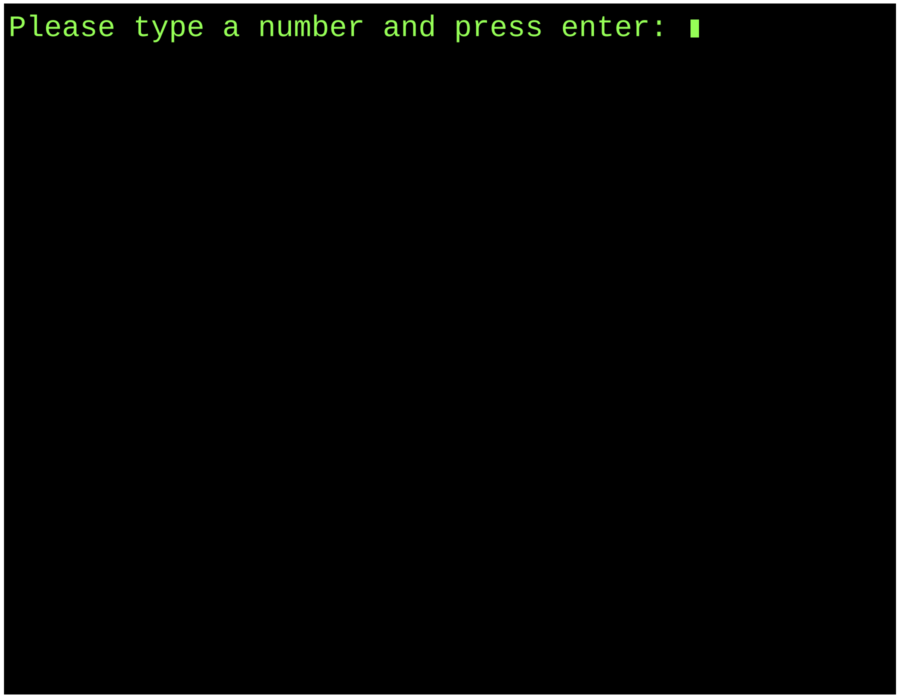
```
/*
   Let's write an infix calculator program for real numbers with variables
      that takes numbers from user input, uses functions for modularity,
      performs calculations with infinitely many consecutive operations,
      and stores the entire mathematical expression in vectors
*/

#include <iostream>
#include <vector>  // this enables the program to use C++ Vector data types

// Global variables to store all operands and operators
// Note: we should really try to avoid using global variables
std::vector <float> allOperands;  // vector of all operands entered by user
std::vector <char> allOperators;  // vector of all operators entered by user
```

Functions

```
    addTwoNumbers()
subtractTwoNumbers()
multiplyTwoNumbers()
  divideTwoNumbers()
         getNumber()
      getOperation()
  performOperation()
      outputResult()
              main()
```

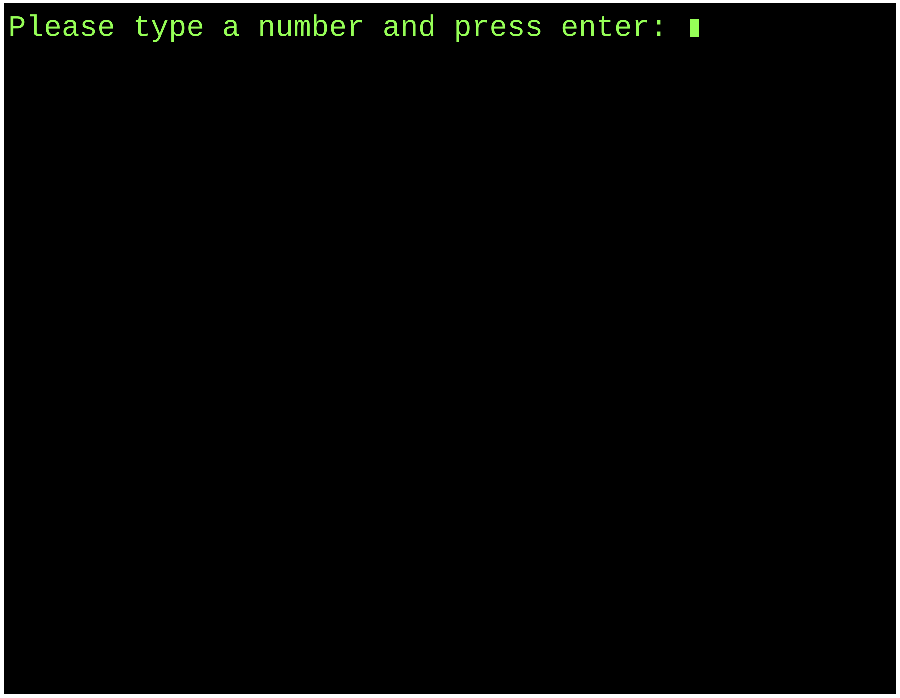
Let's see how these vectors should behave

calculator61

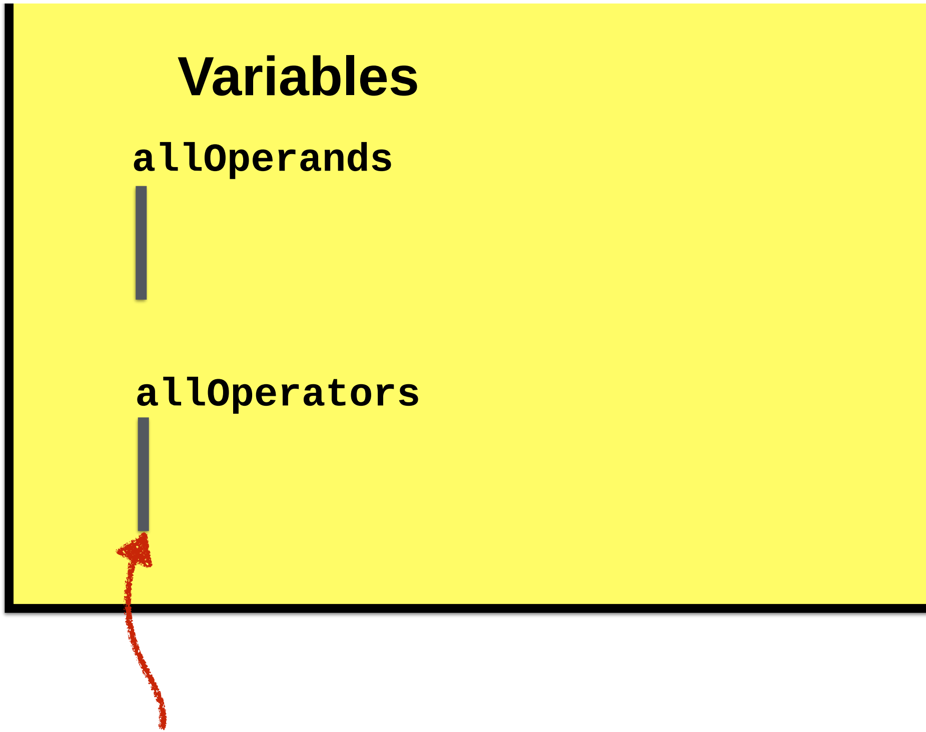


```
Please type a number and press enter: ▮
```

Let's see how these vectors should behave calculator61



```
Please type a number and press enter: ▮
```

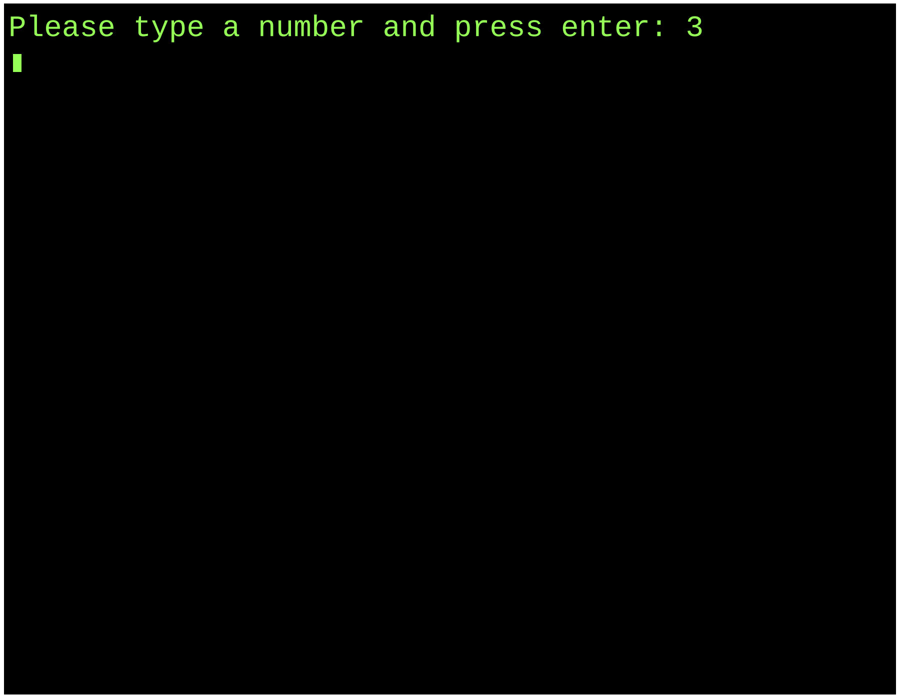


Variables

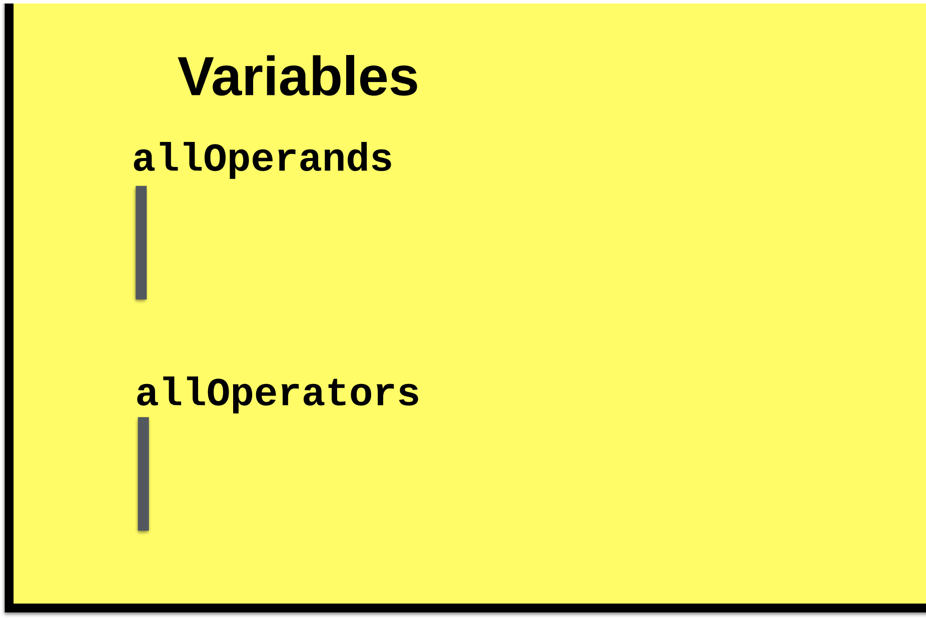
Both vectors start empty

Let's see how these vectors should behave

calculator61



```
Please type a number and press enter: 3
▮
```



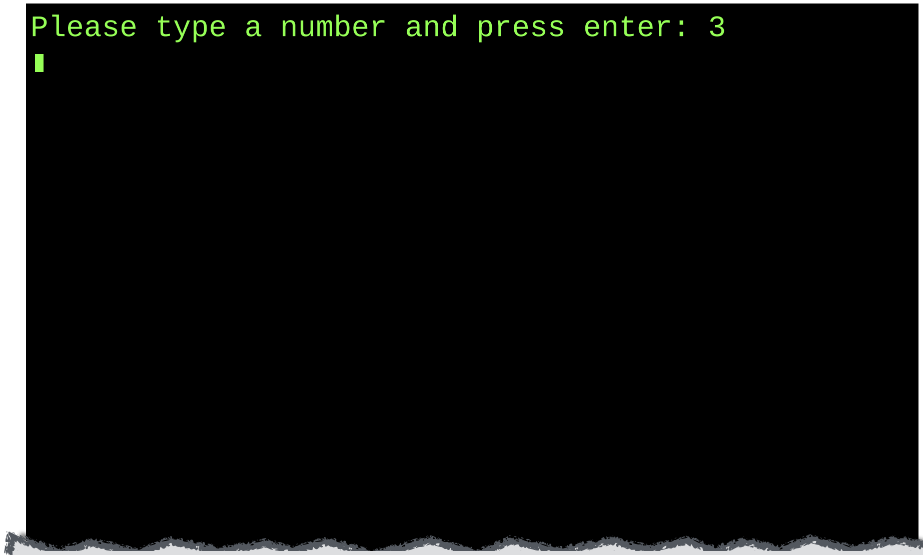
Variables

`allOperands`

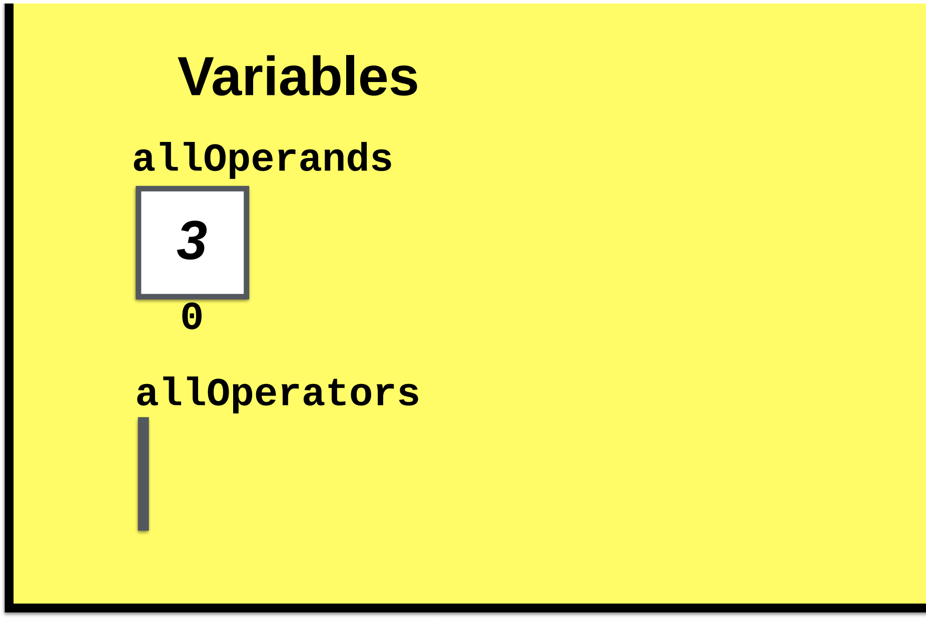
`allOperators`

Let’s see how these vectors should behave

calculator61



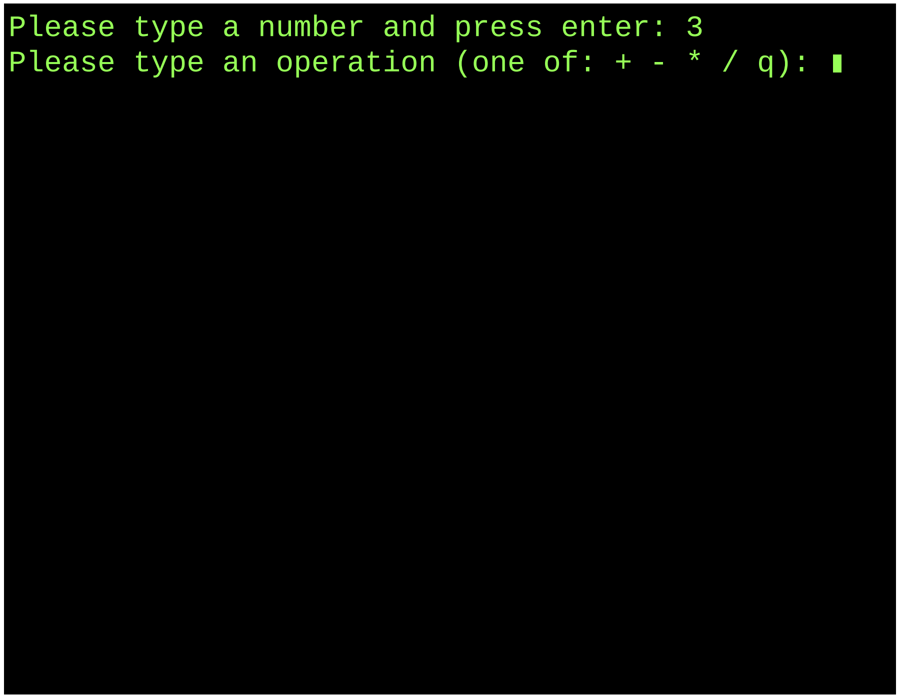
```
Please type a number and press enter: 3
▮
```



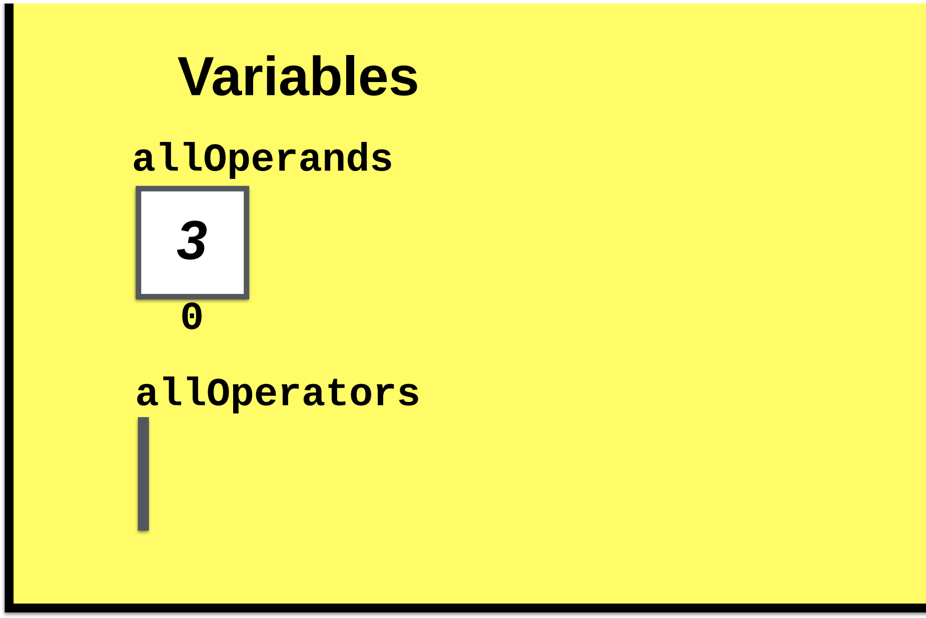
```
// Function to prompt the user to input a number that is returned in a variable
bool getNumber(float &number) {

   // Ask the user to give us a number for our next operand
   std::cout << "Please type a number and press enter: ";
   // Wait for the user to enter a number and assign it return variable
   std::cin >> number;
   // Store number at the end of vector of operands
   allOperands.push_back(number);  // push_back is a member function of vector

   return false;
}
```

Push element for operand onto vector

calculator61

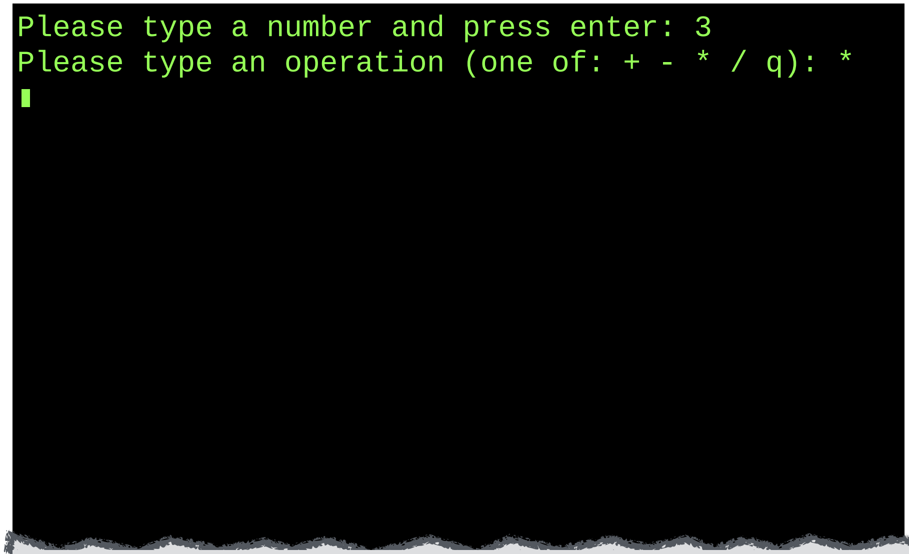


```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): ▮
```



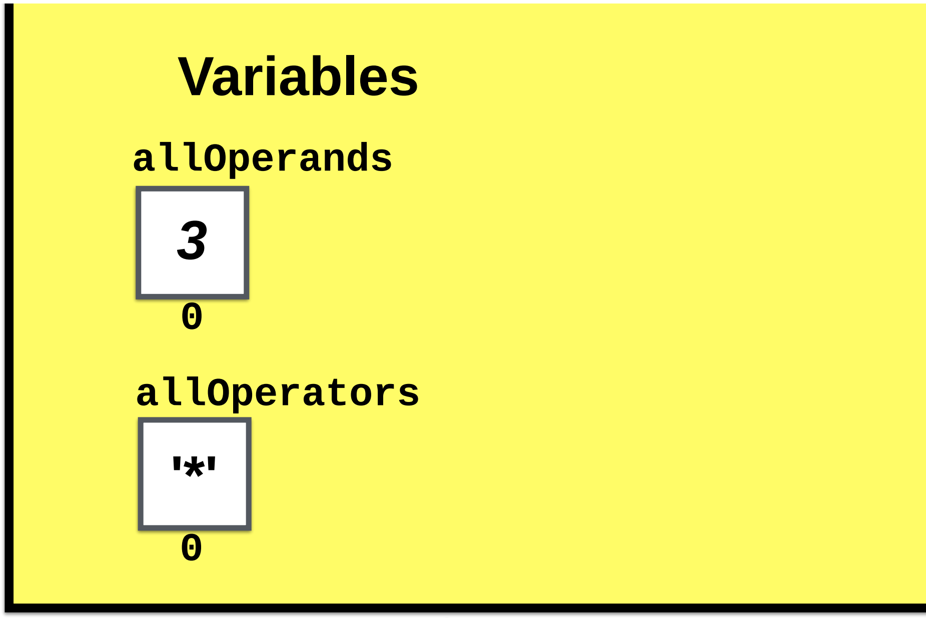
Variables

calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
▮
```

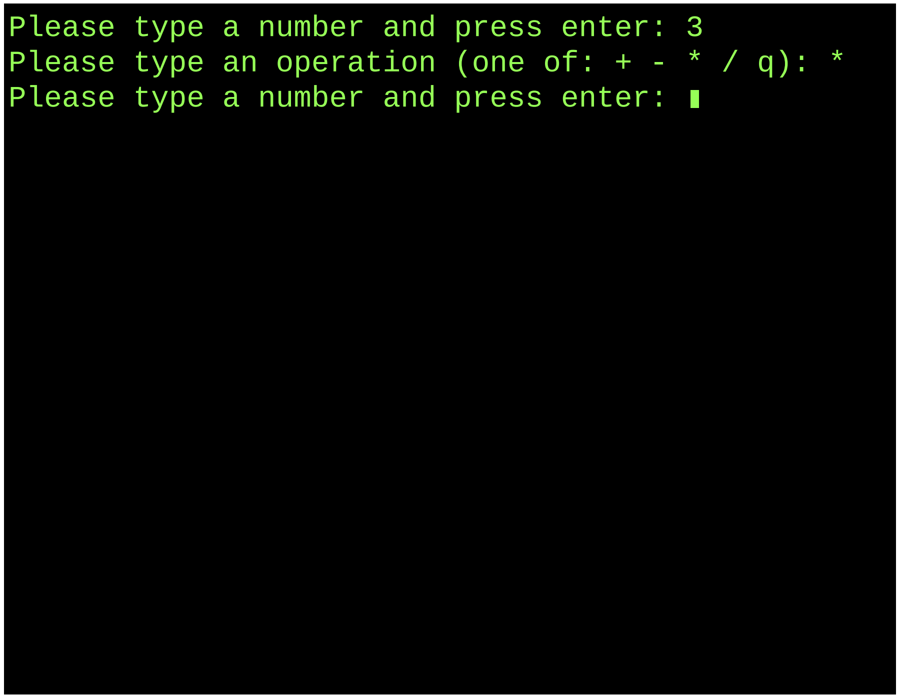
Variables



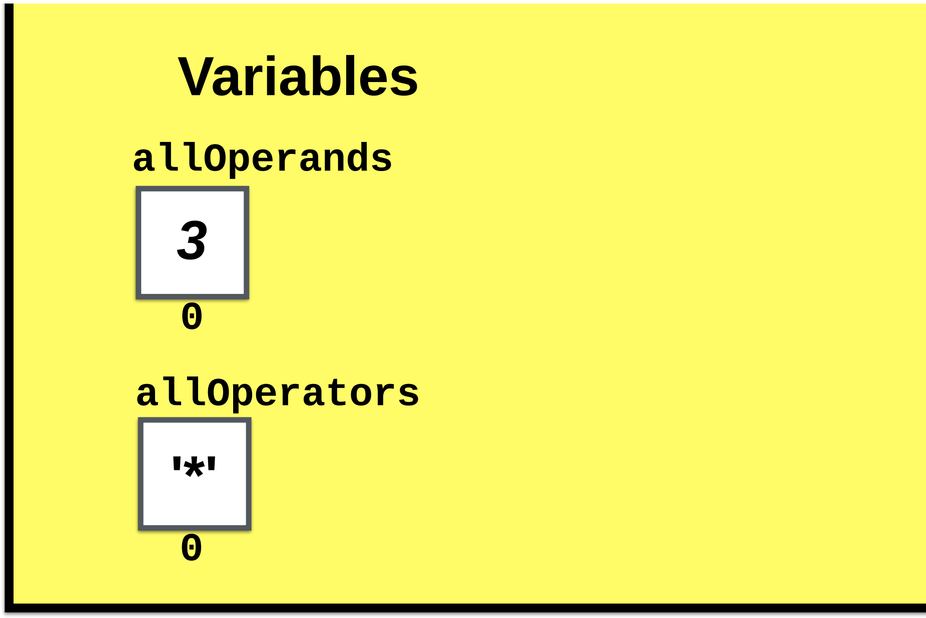
```
// Function to prompt the user to input a number that is returned in a variable
bool getOperator(char &operation) {
   // Note: "operator" is a reserved word in C++; it cannot be a variable name

   // Ask the user to input a character for our next operator
   std::cout << "Please type a math operator (one of: + - * /): ";
   // Wait for the user to enter operator and assign it variable operator
   std::cin >> operation;
   // Store character at the end of vector of operators
   allOperators.push_back(operation);  // push_back member function of vector

   return false;
}
```

Push element for operator onto vector

calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 
```



Variables

allOperands

3

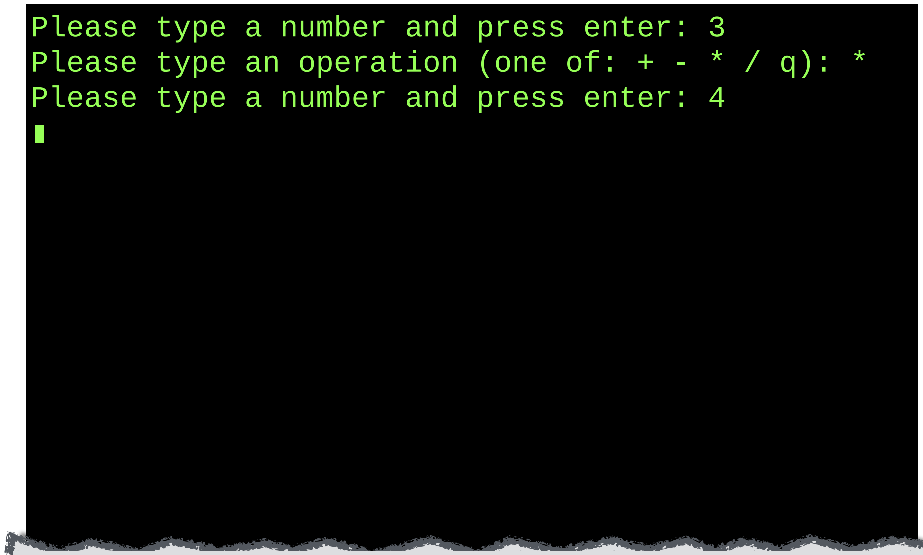
0

allOperators

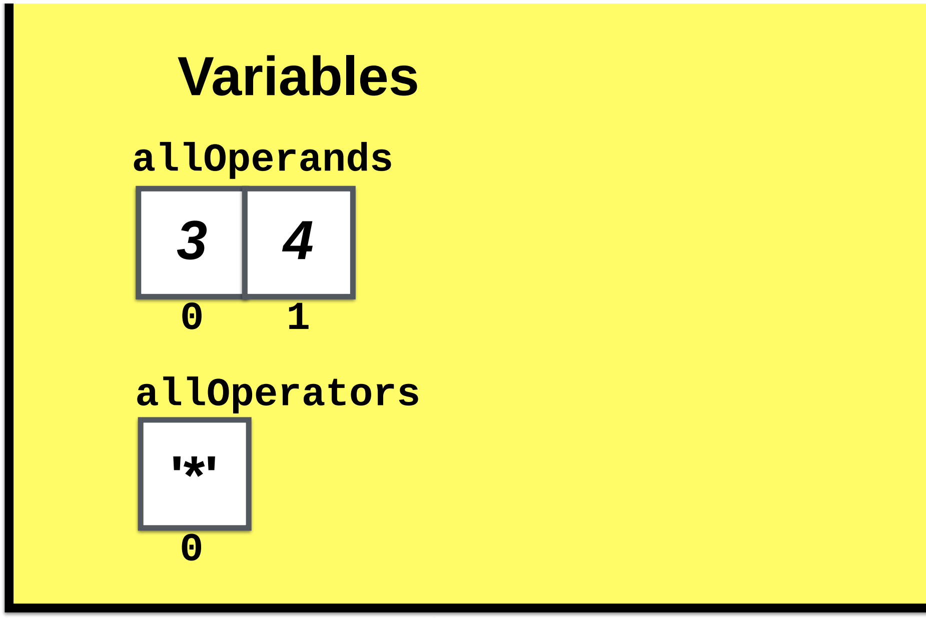
'*'

0

calculator61



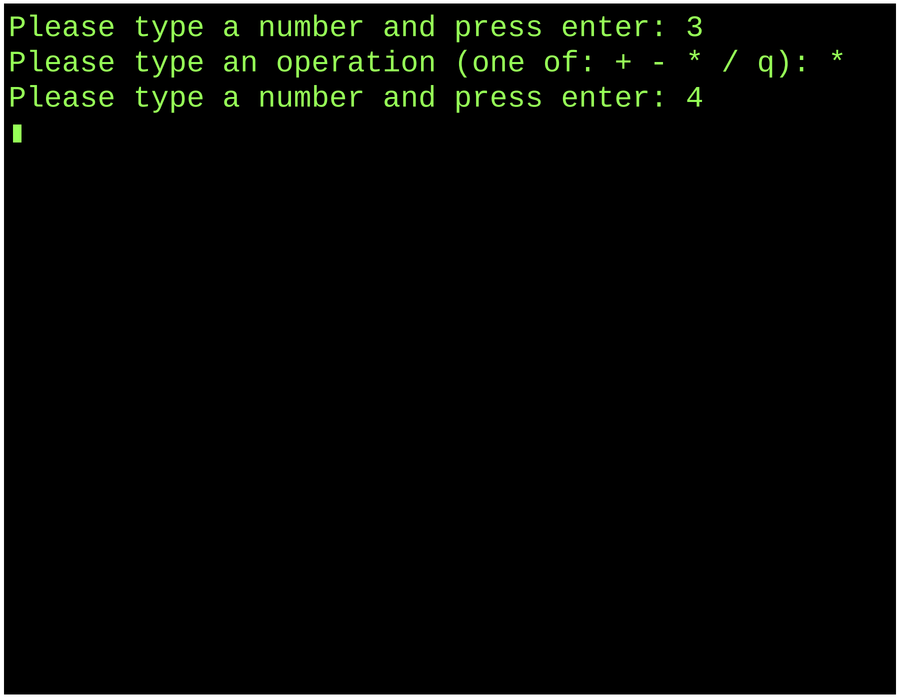
```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
```



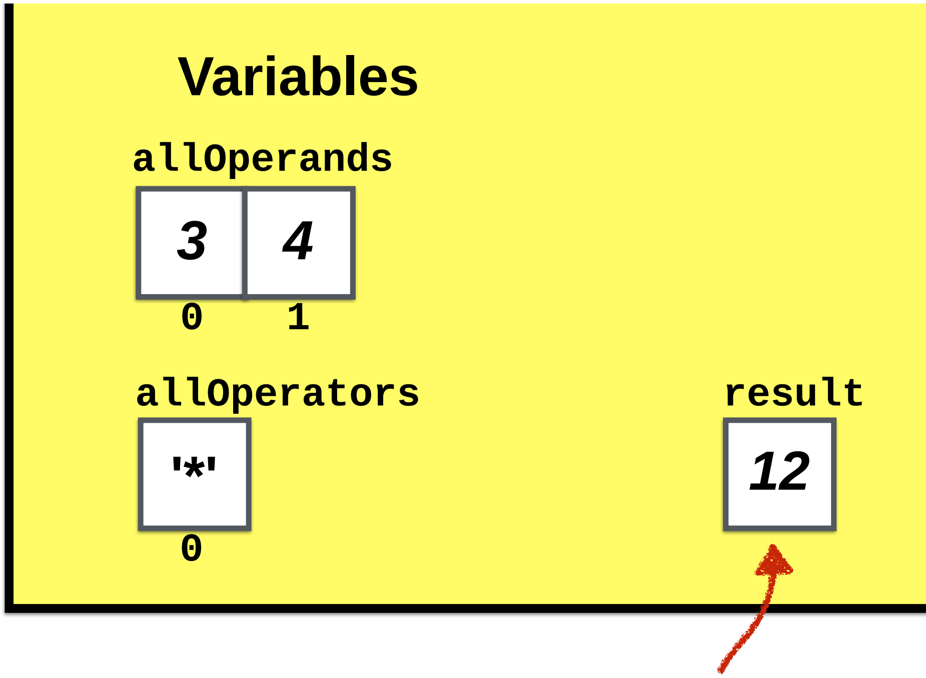
```
// Function to prompt the user to input a number that is returned in a variable
bool getNumber(float &number) {

   // Ask the user to give us a number for our next operand
   std::cout << "Please type a number and press enter: ";
   // Wait for the user to enter a number and assign it return variable
   std::cin >> number;
   // Store number at the end of vector of operands
   allOperands.push_back(number);  // push_back is a member function of vector

   return false;
}
```

Push element for operand onto vector

calculator61

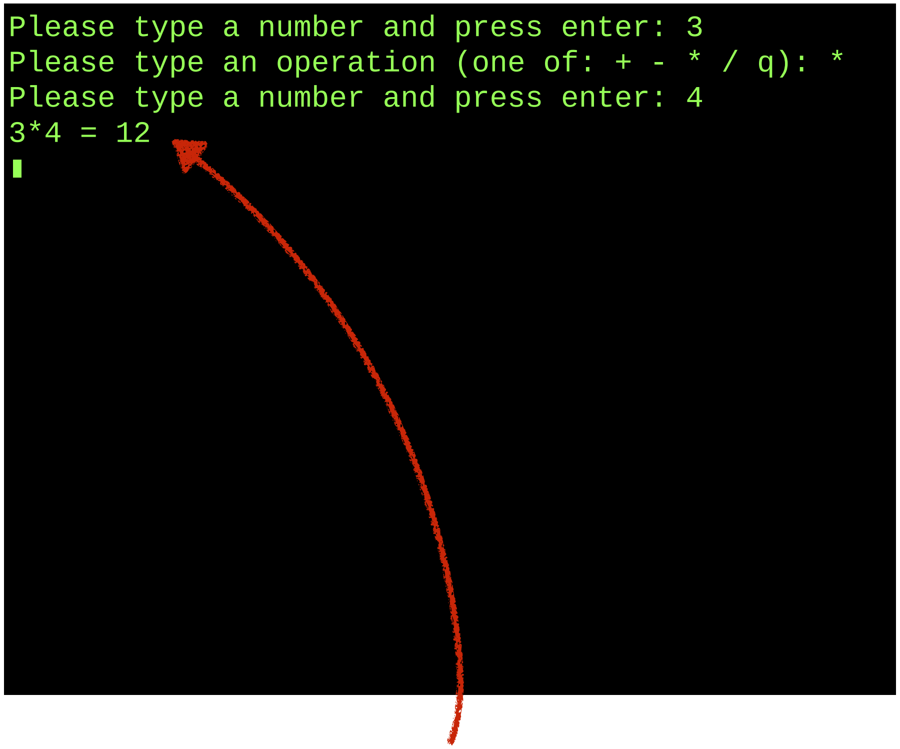


```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
▮
```



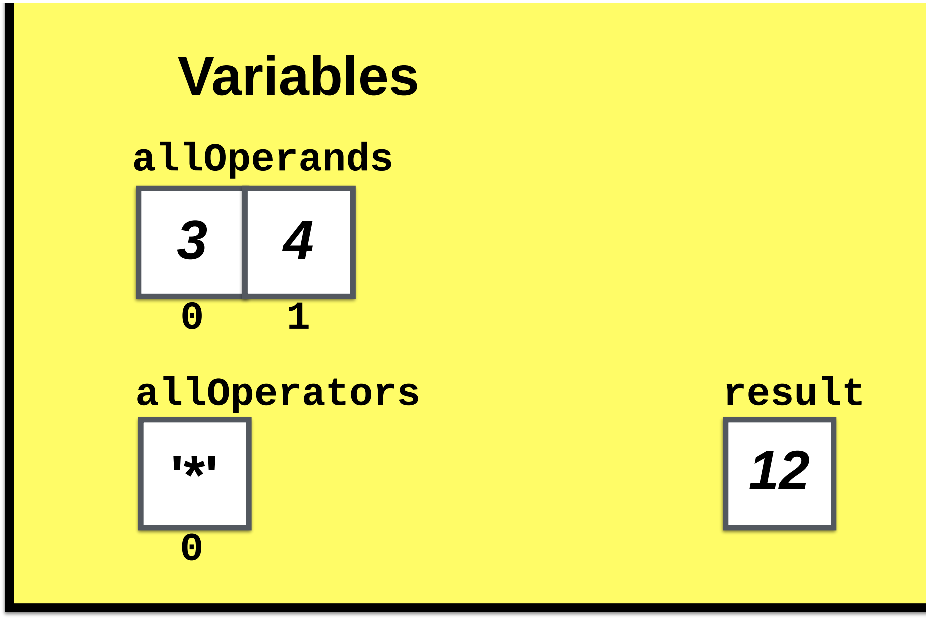
Result of the operation computed

calculator61

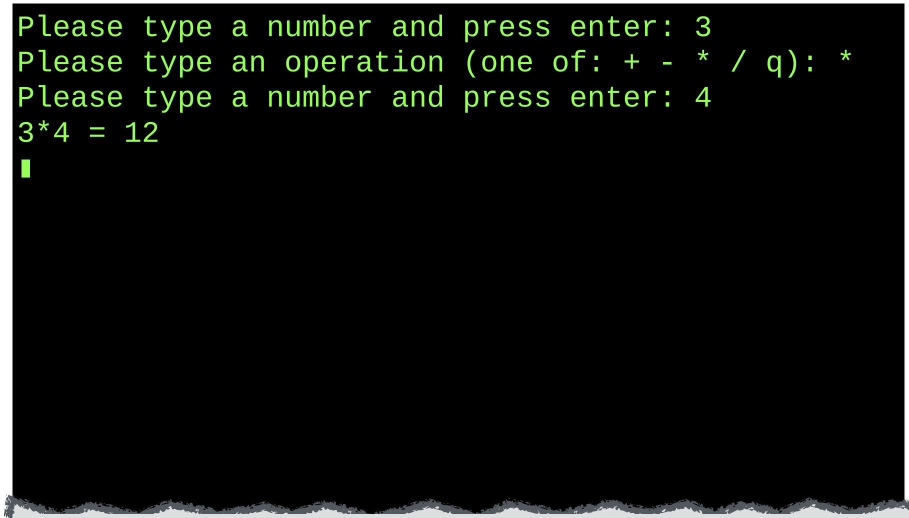


```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
```

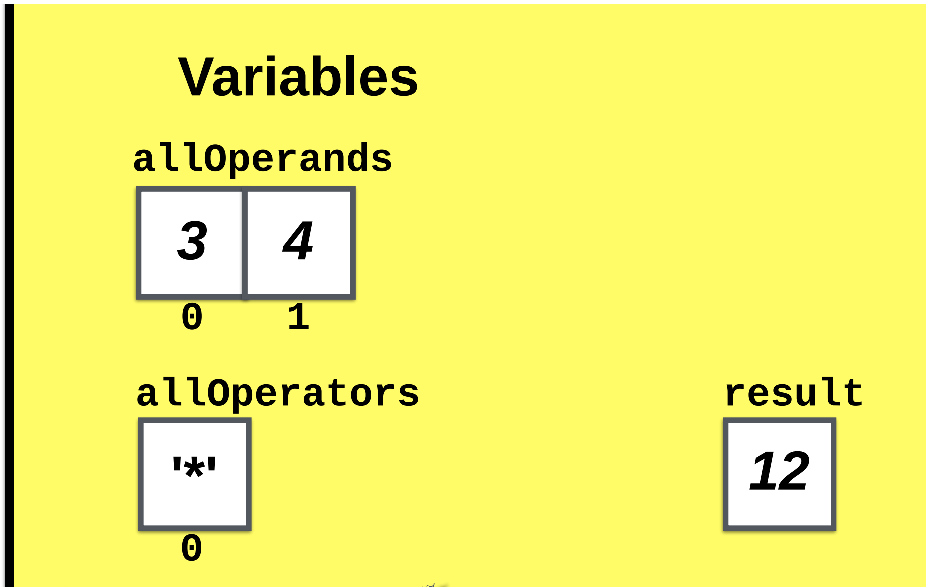
Result of the operation printed



calculator61

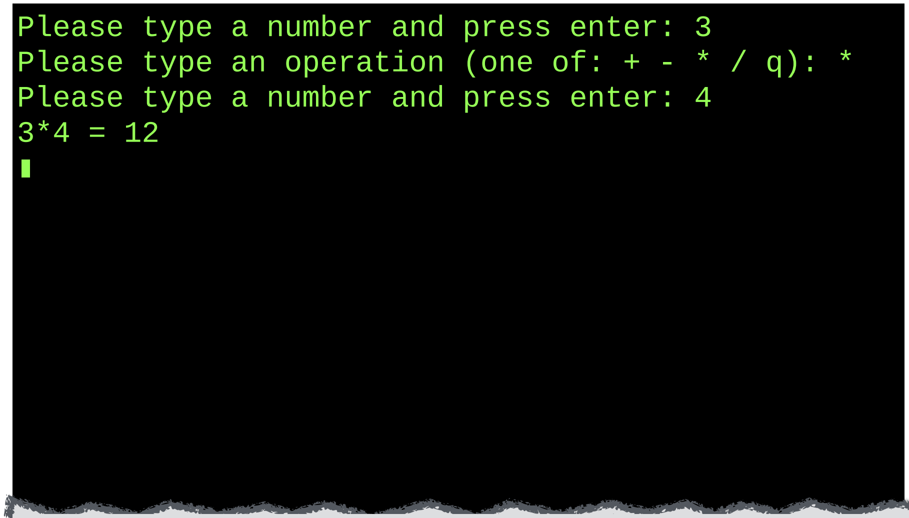


```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
```

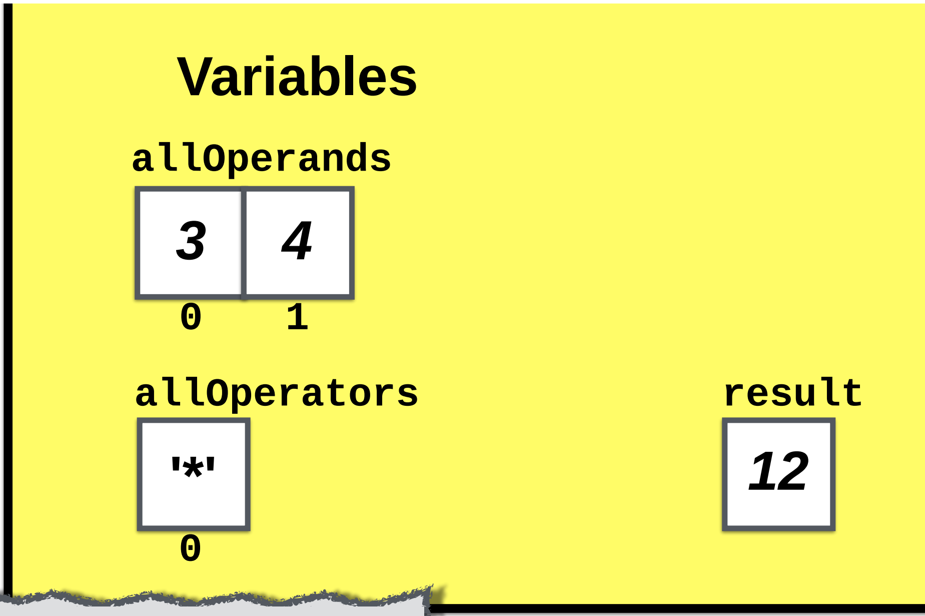


```
// Output the entire current equation to the screen
bool outputEquation(float result) {

    Output first operand

    // For loop to output the entire math expression with iteration variable i
    int i;
    for (i=0; i<allOperators.size(); i++) {
        Print current operator and operand
    }
    Print result of all operations

    return false;
}
```

calculator61



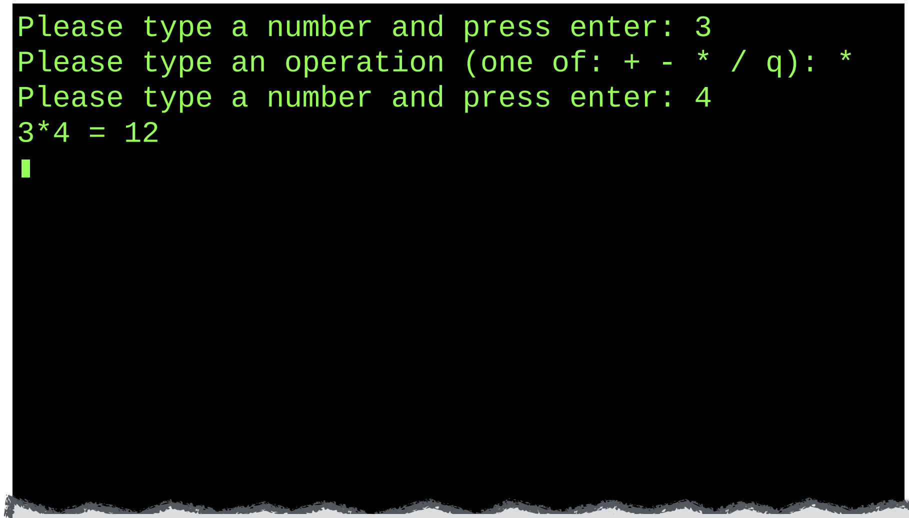
```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
▮
```



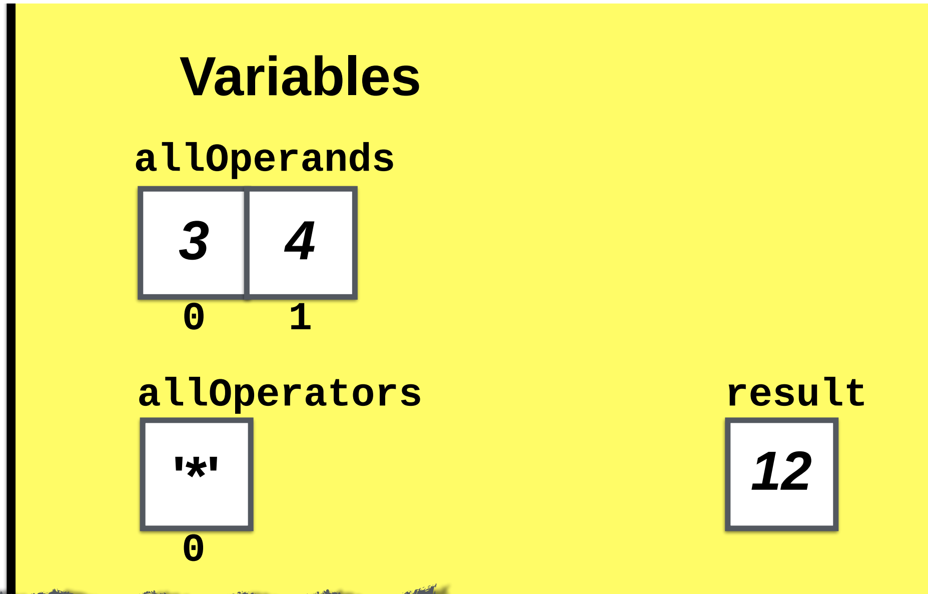
```
// Output the entire current equation to the screen
bool outputEquation(float result) {

   // Print first operand at index [0]

   // For loop to output the entire math expression with iteration variable i
   int i;
   for (i=0; i<allOperators.size(); i++) {
      // Print operator at index [i] and operand at index [i+1]
   }

   // Print current result of all operations

   return false;
}
```

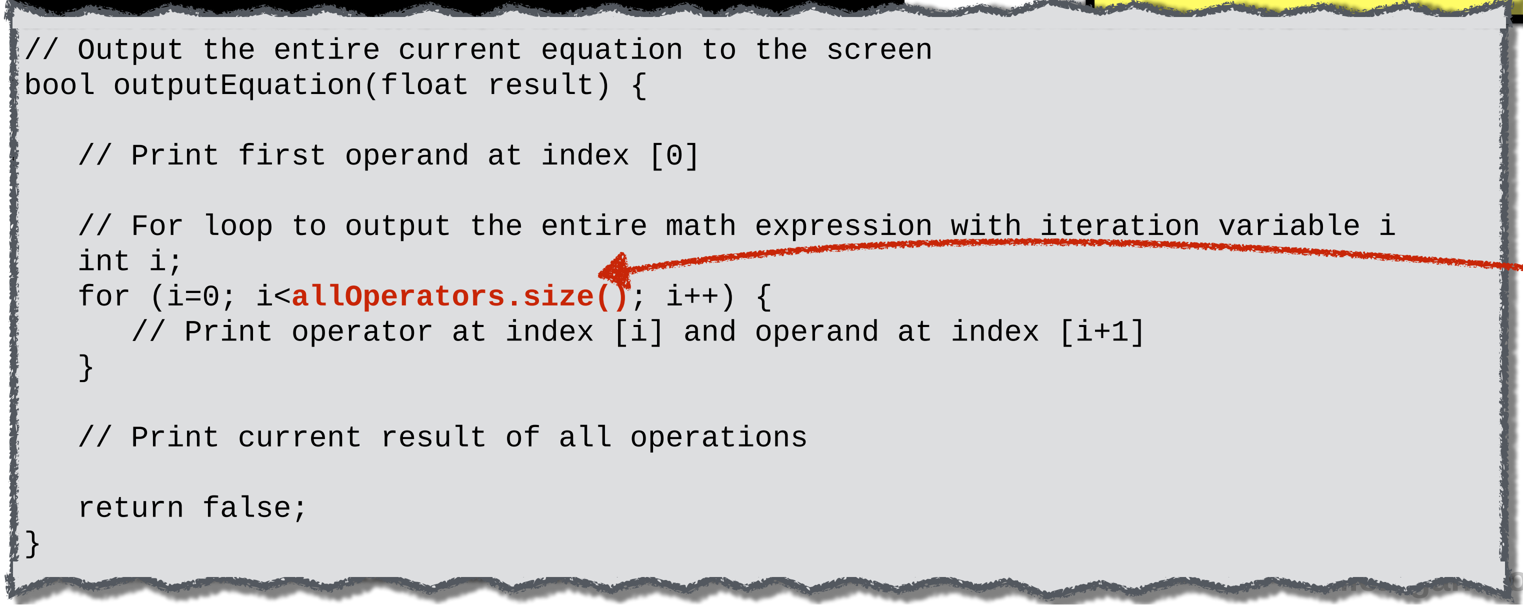
The .size() method of a vector returns the size of the vector

calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
▮
```

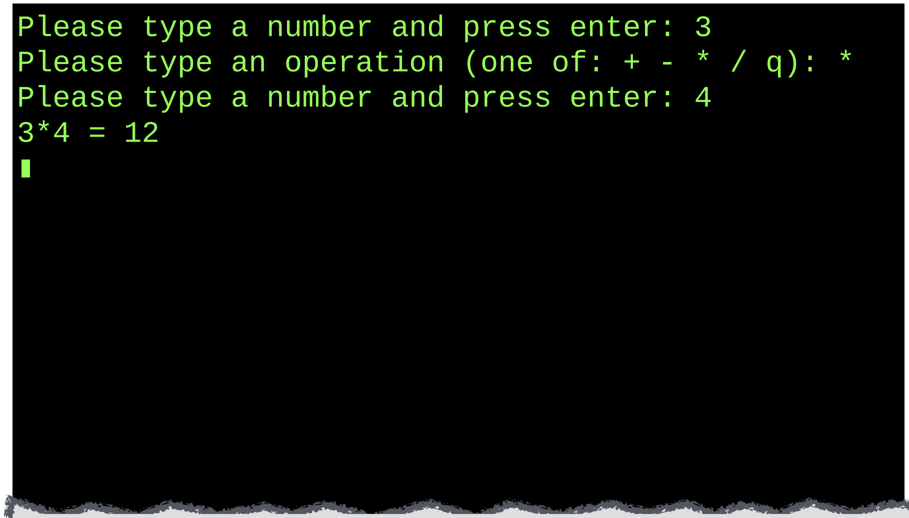




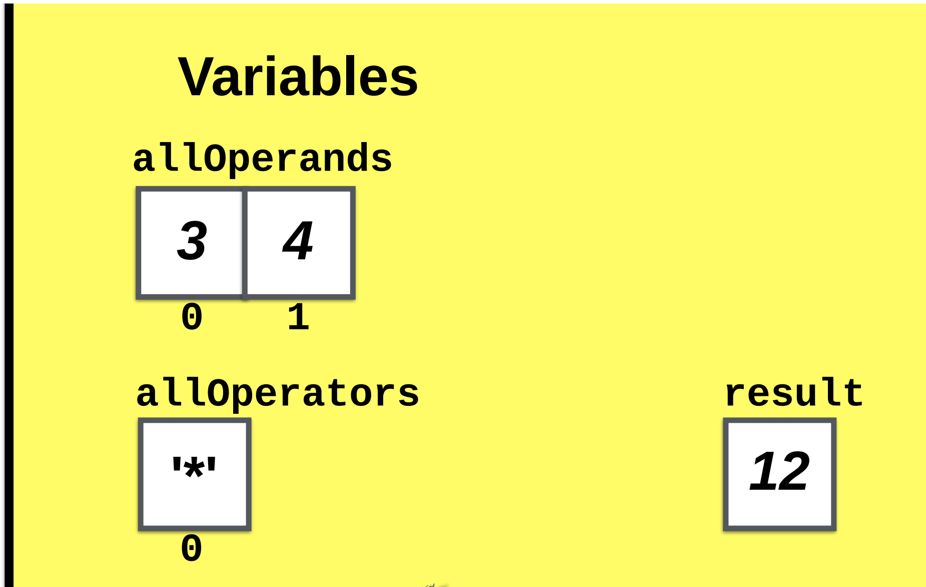
```
// Output the entire current equation to the screen
bool outputEquation(float result) {

   // Print first operand at index [0]

   // For loop to output the entire math expression with iteration variable i
   int i;
   for (i=0; i<allOperators.size(); i++) {
      // Print operator at index [i] and operand at index [i+1]
   }

   // Print current result of all operations

   return false;
}
```

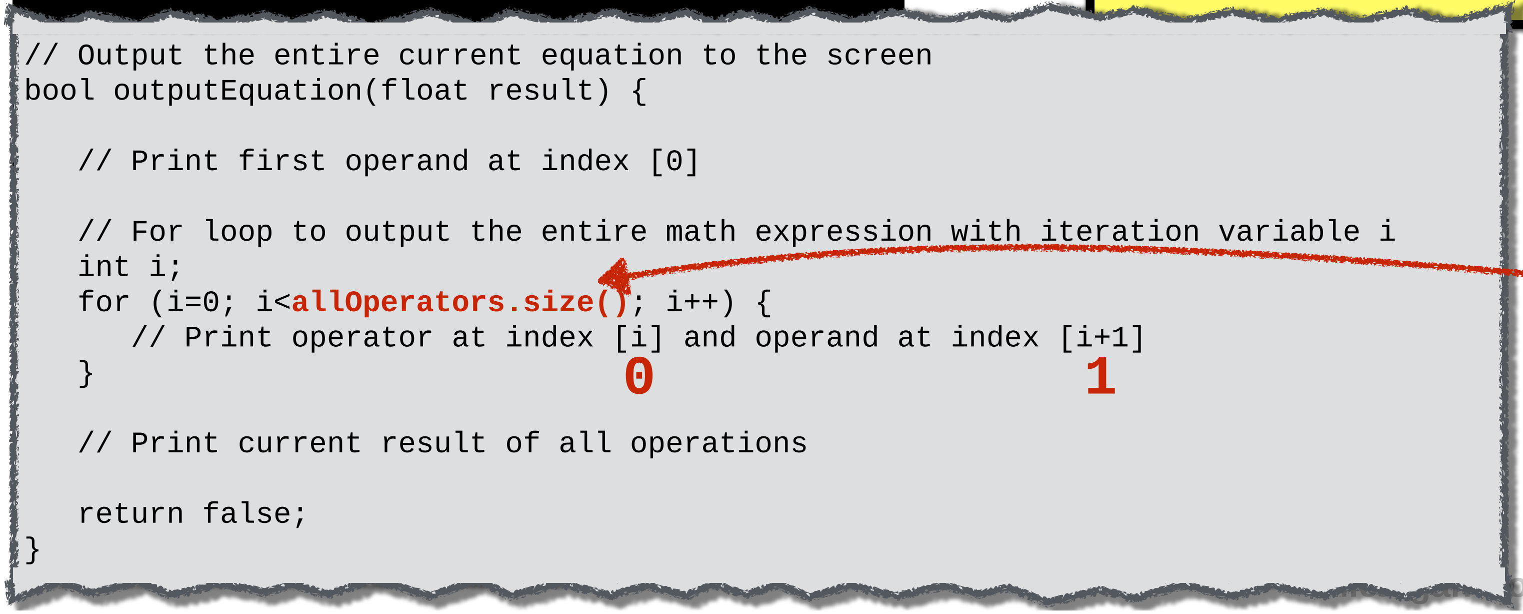
Returns 1 at this point in the program.

calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
▮
```

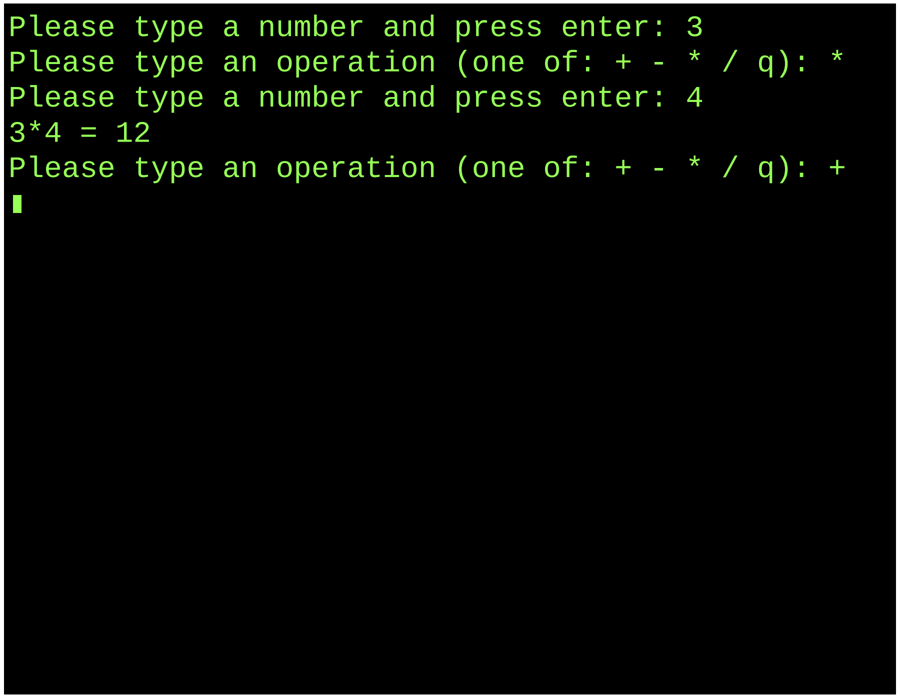




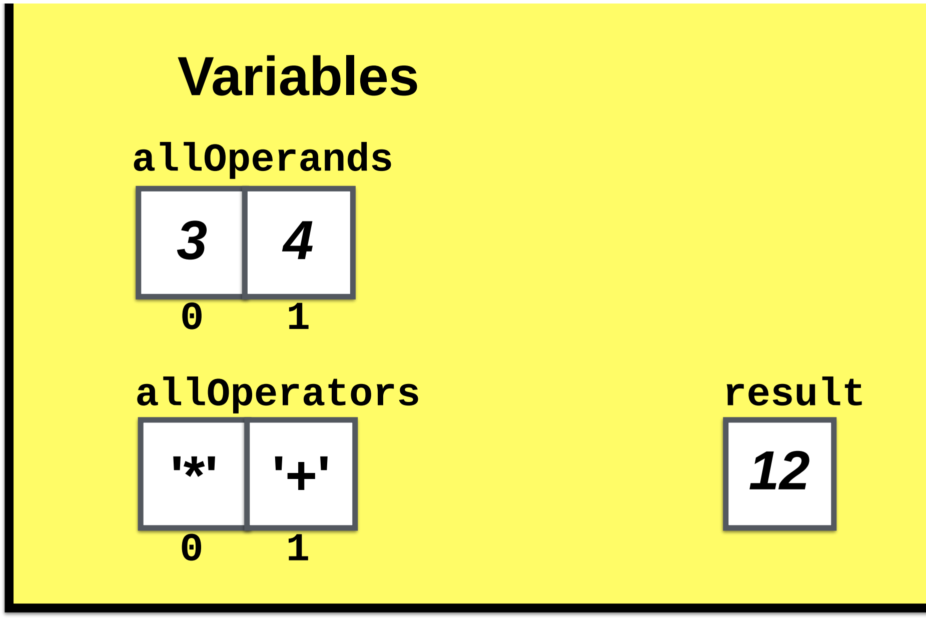
```
// Output the entire current equation to the screen
bool outputEquation(float result) {

   // Print first operand at index [0]

   // For loop to output the entire math expression with iteration variable i
   int i;
   for (i=0; i<allOperators.size(); i++) {
      // Print operator at index [i] and operand at index [i+1]
   }
                                    0                       1

   // Print current result of all operations

   return false;
}
```

Loop performs 1 iteration with `i = 0`

calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
▮
```



Variables

allOperands

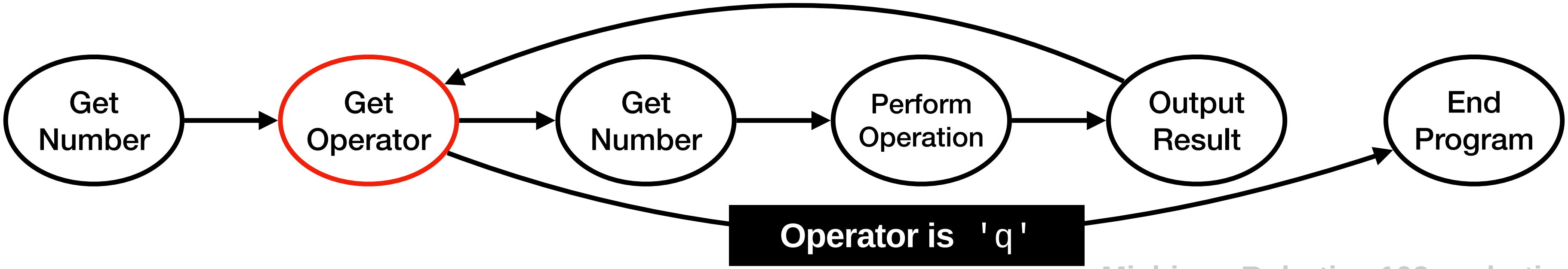
3	4
0	1

allOperators

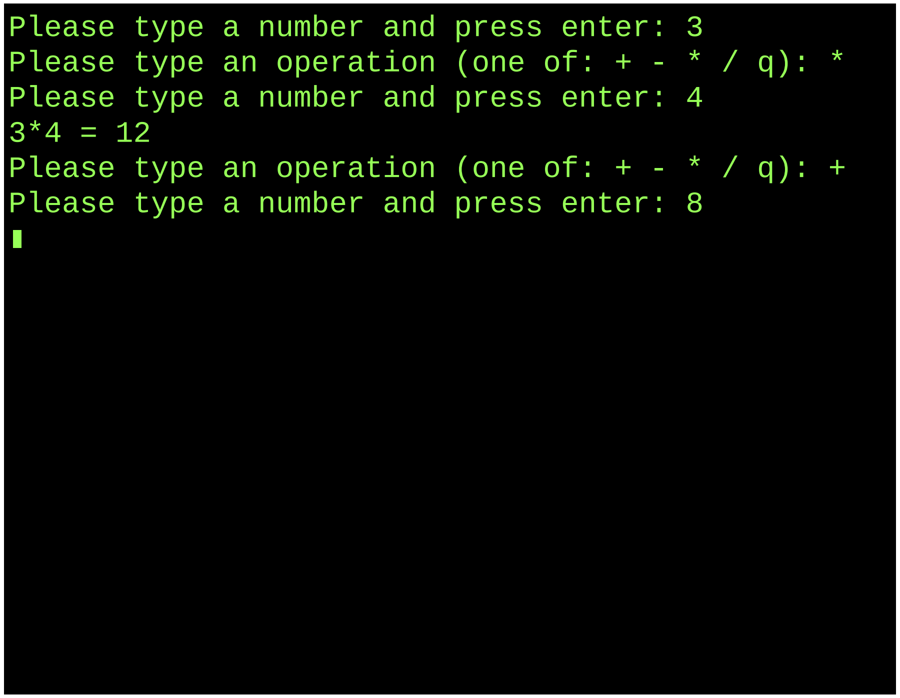
'*'	'+'
0	1

result

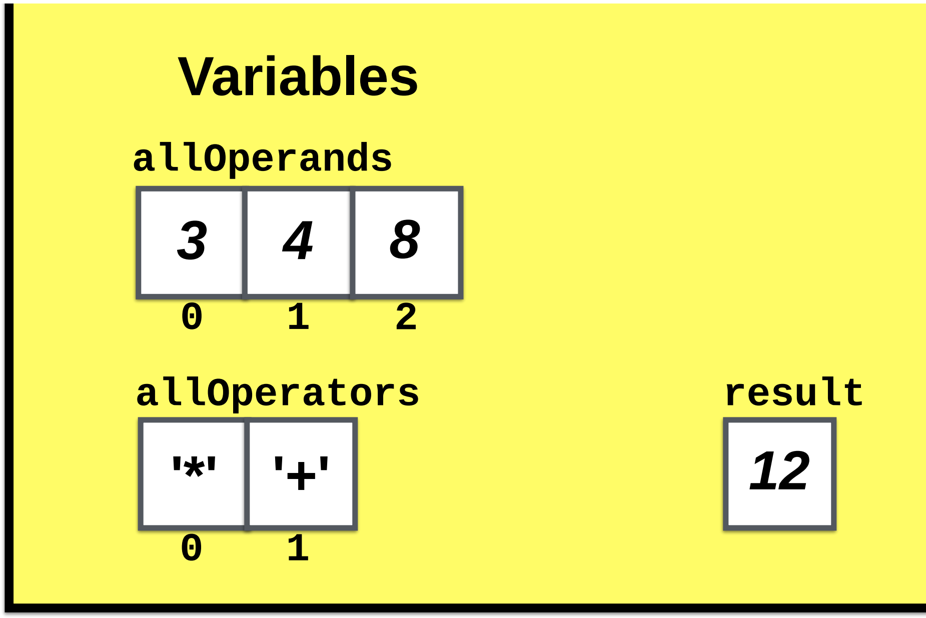
12



calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
▮
```



Variables

allOperands

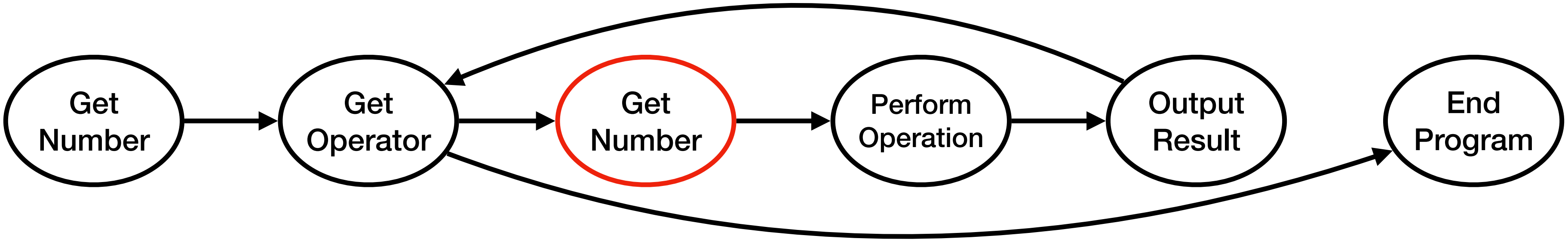
3	4	8
0	1	2

allOperators

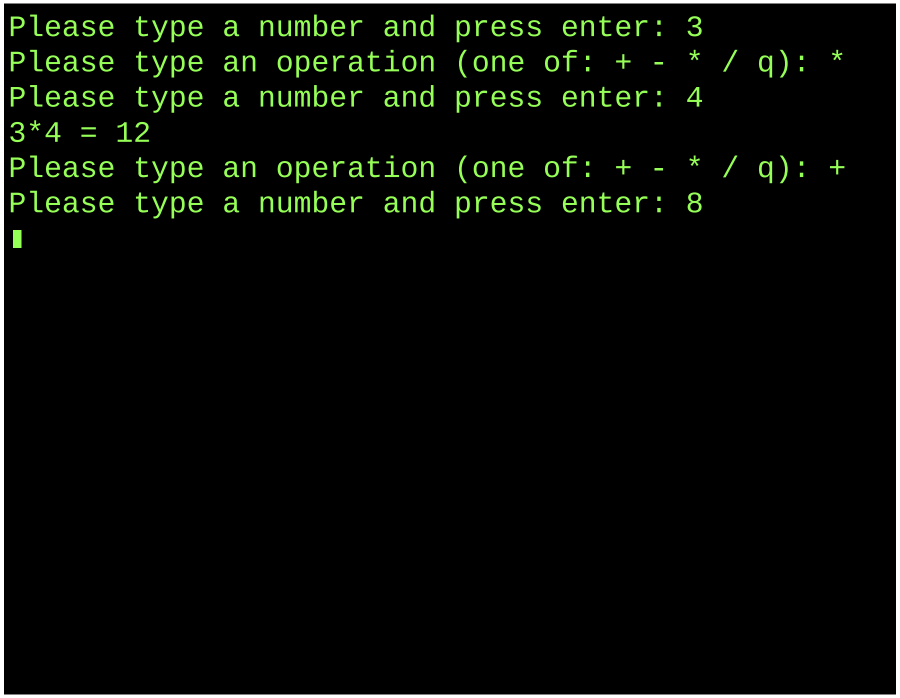
'*'	'+'
0	1

result

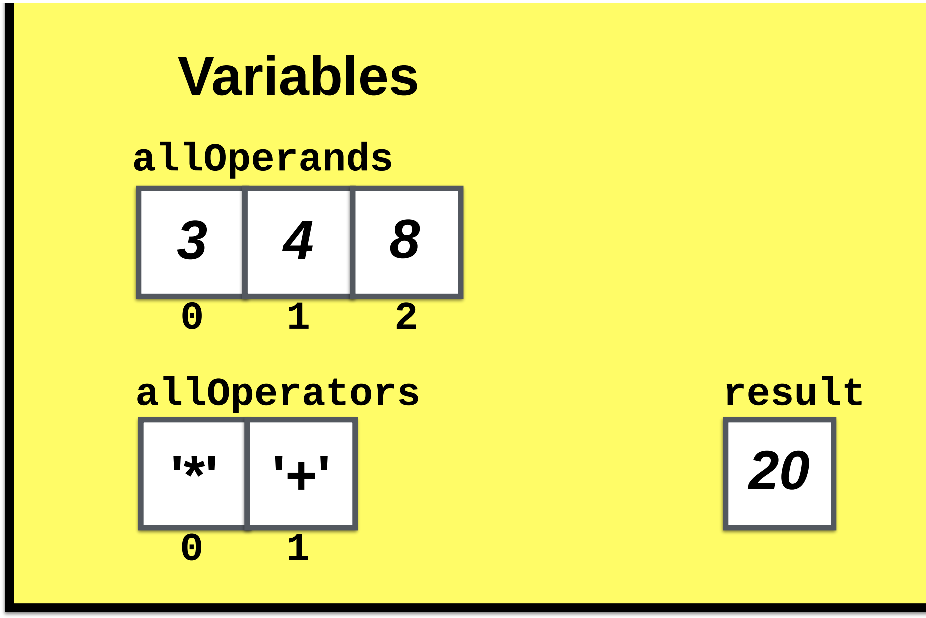
12



calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
▮
```



Variables

allOperands

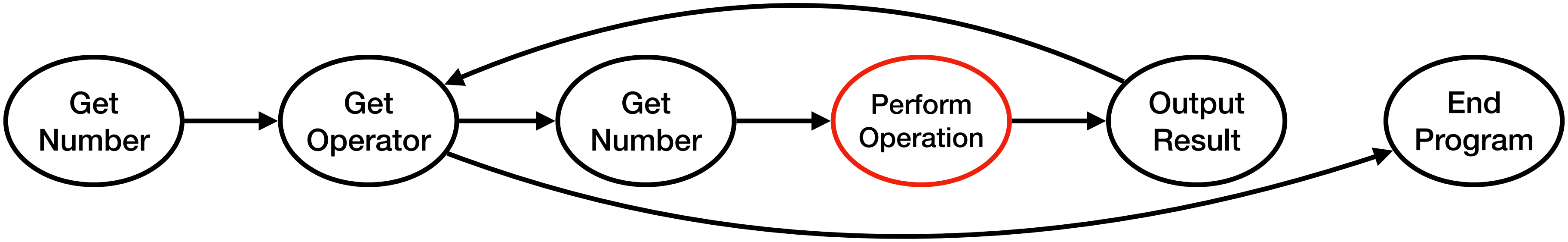
3	4	8
0	1	2

allOperators

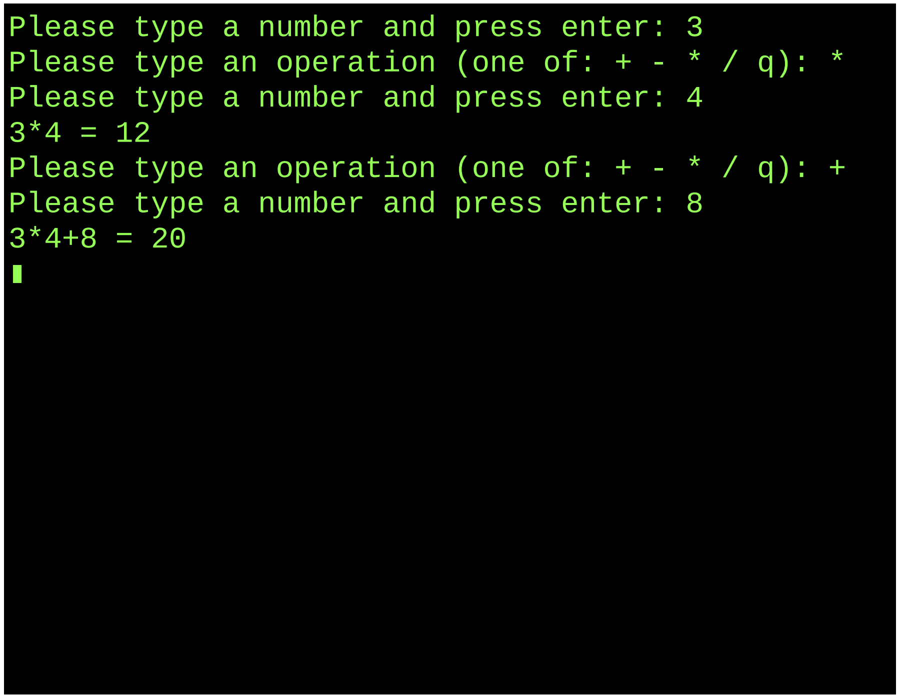
'*'	'+'
0	1

result

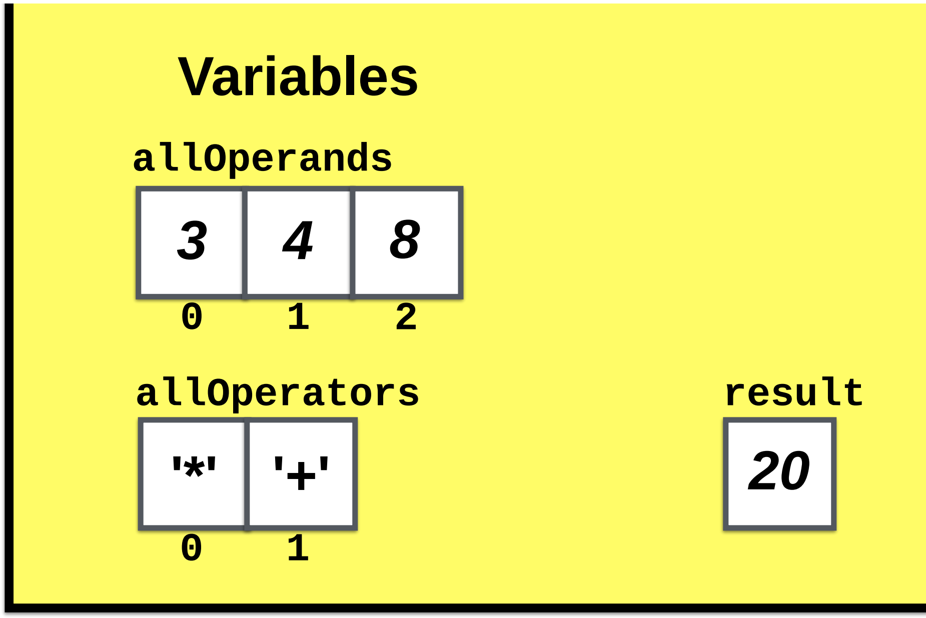
20



calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
3*4+8 = 20
▮
```



Variables

allOperands

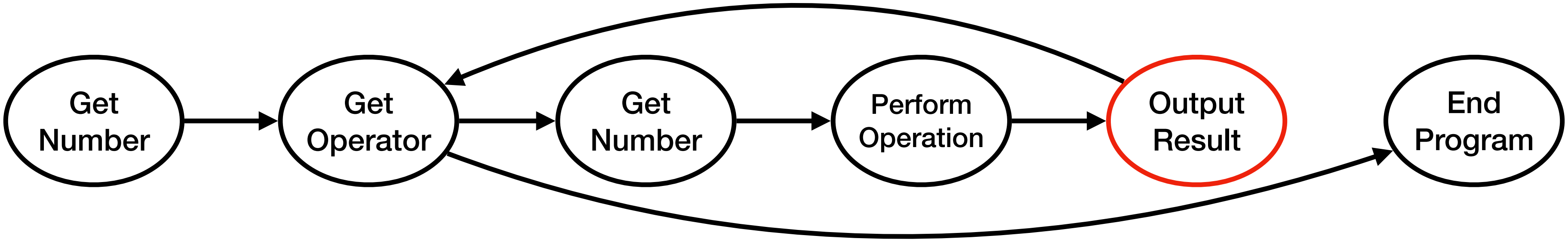
3	4	8
0	1	2

allOperators

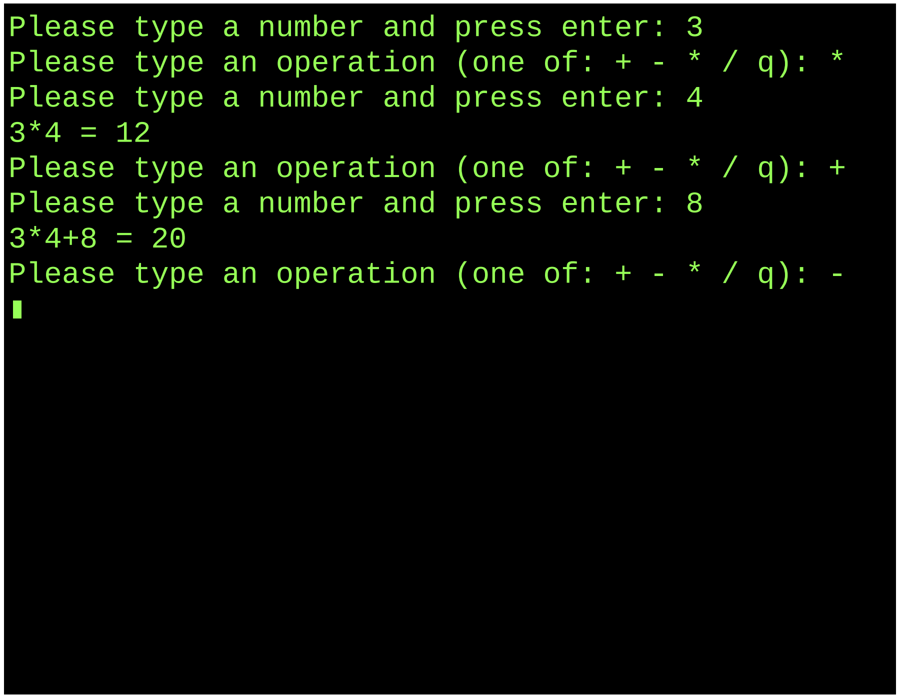
'*'	'+'
0	1

result

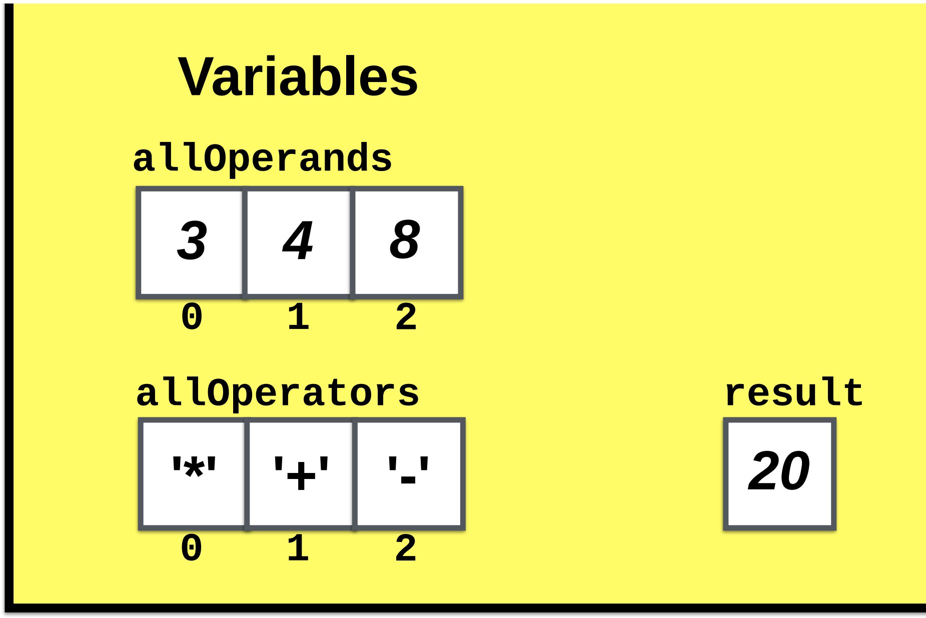
20



calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
3*4+8 = 20
Please type an operation (one of: + - * / q): -
```



Variables

allOperands

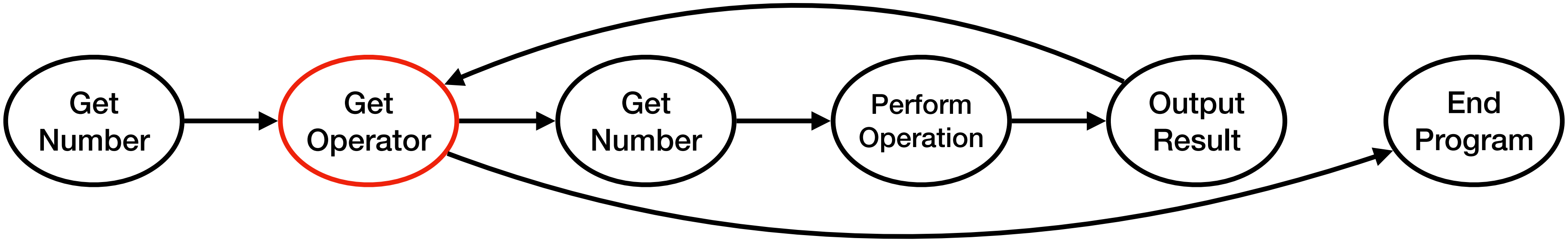
3	4	8
0	1	2

allOperators

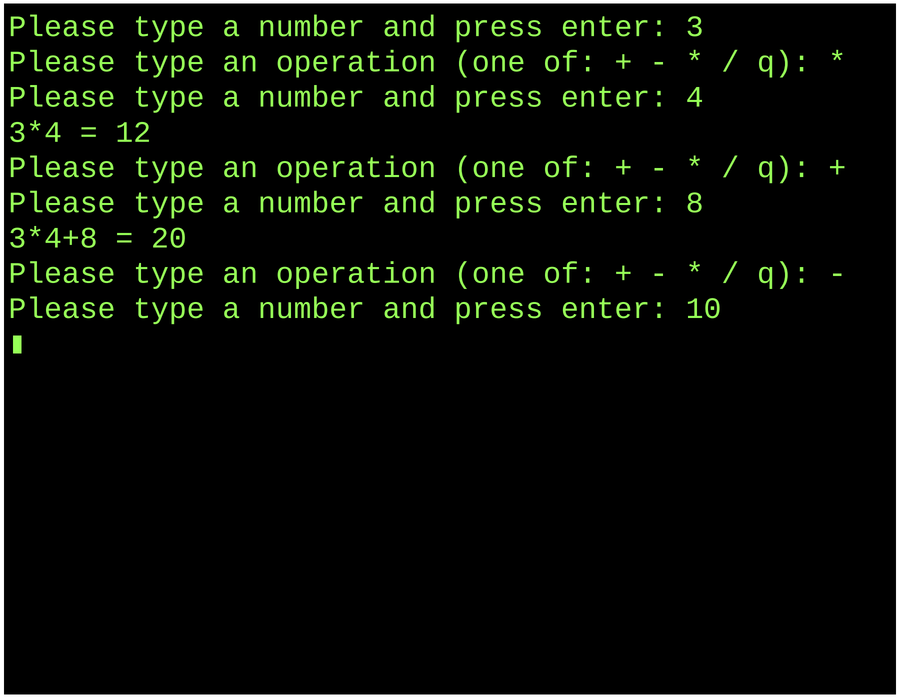
'*'	'+'	'-'
0	1	2

result

20



calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
3*4+8 = 20
Please type an operation (one of: + - * / q): -
Please type a number and press enter: 10
▮
```

Variables

allOperands

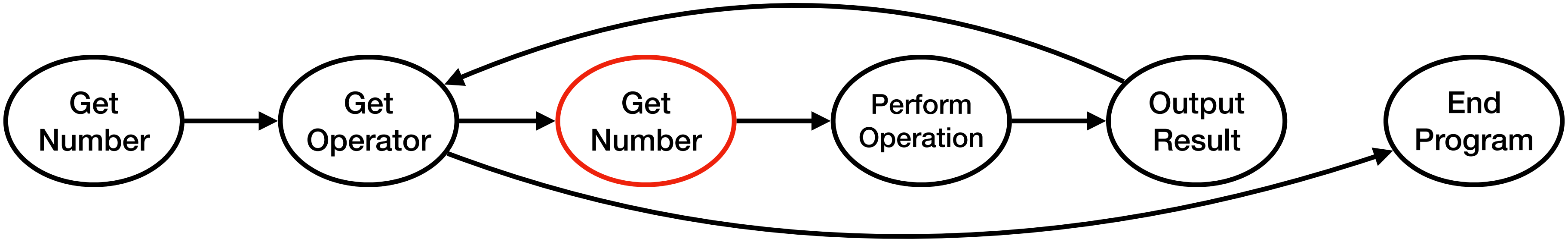
3	4	8	10
0	1	2	3

allOperators

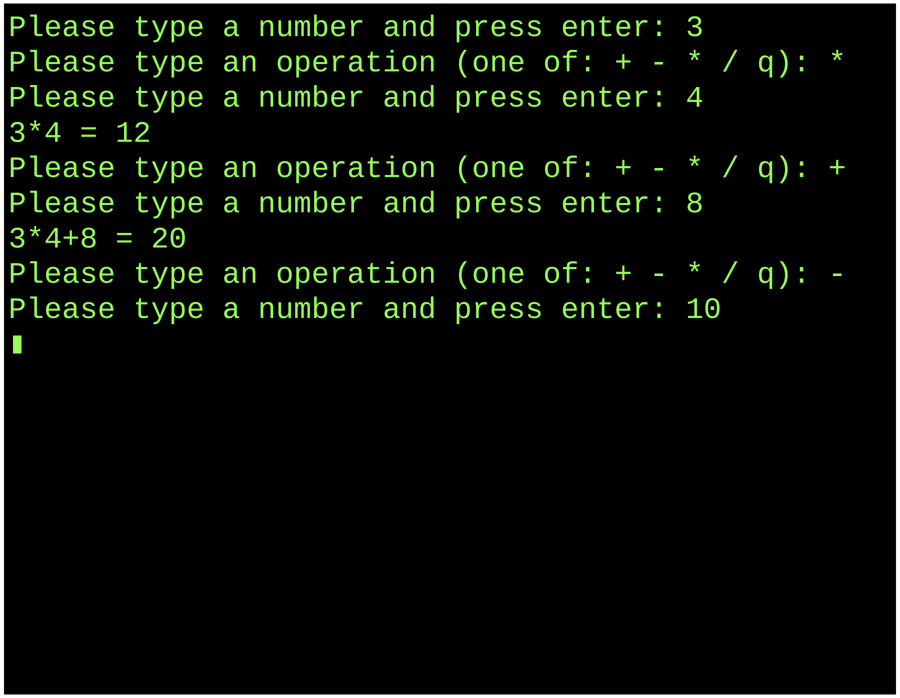
'*'	'+'	'-'
0	1	2

result

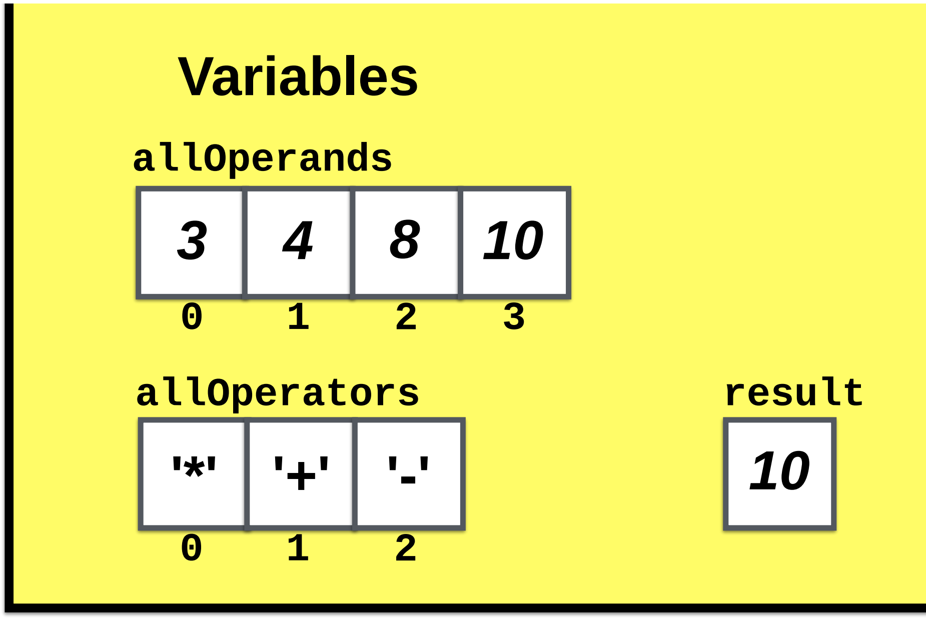
20



calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
3*4+8 = 20
Please type an operation (one of: + - * / q): -
Please type a number and press enter: 10
▮
```



Variables

allOperands

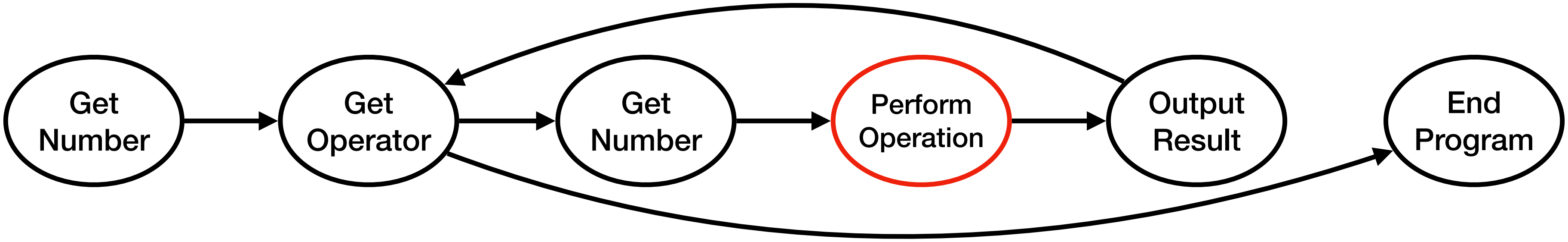
3	4	8	10
0	1	2	3

allOperators

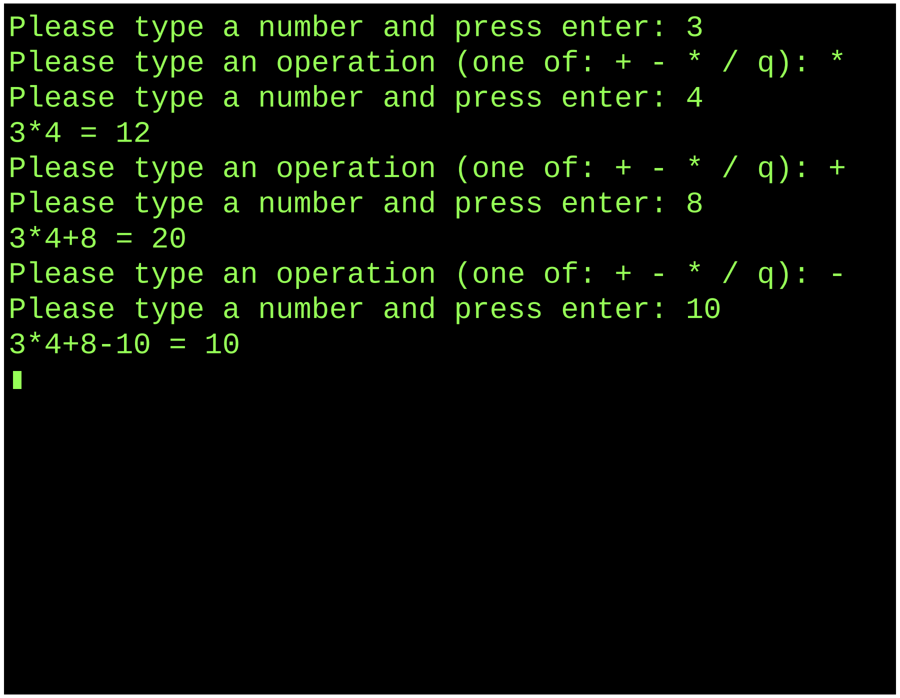
'*'	'+'	'-'
0	1	2

result

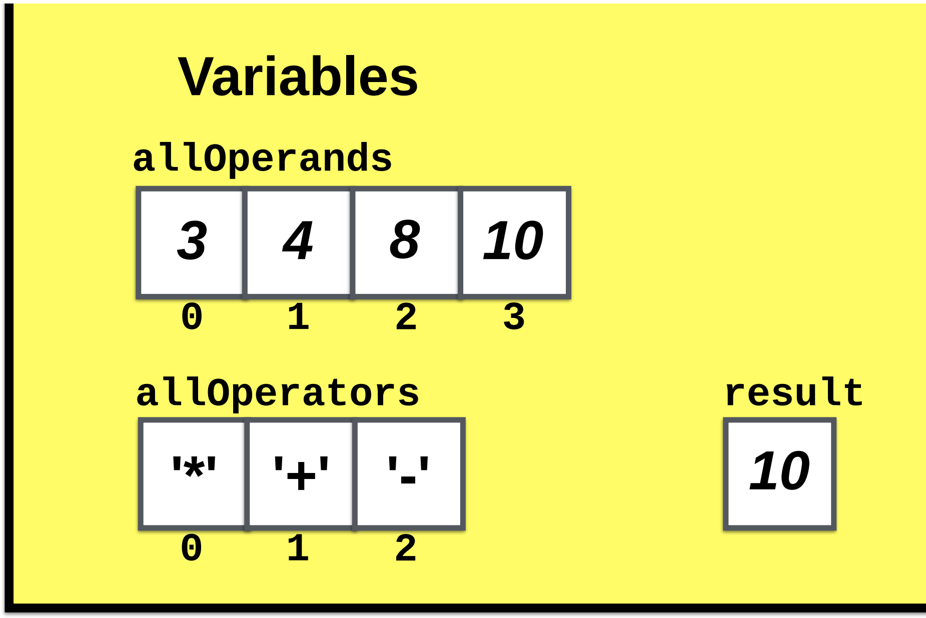
10



calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
3*4+8 = 20
Please type an operation (one of: + - * / q): -
Please type a number and press enter: 10
3*4+8-10 = 10
▮
```



Variables

allOperands

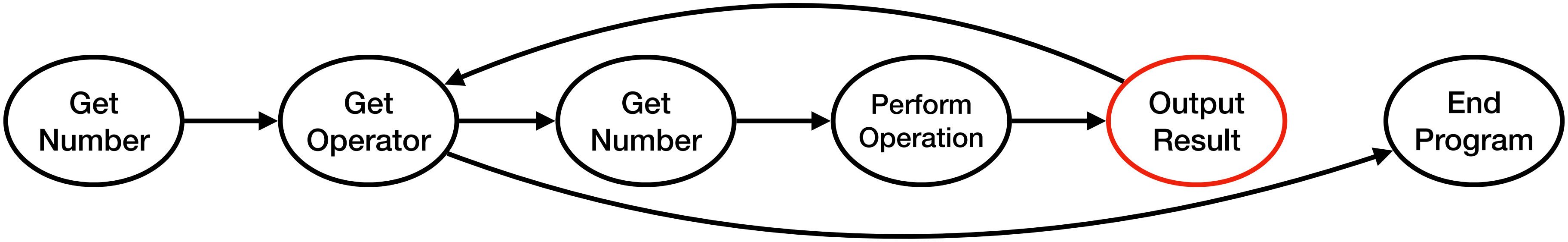
0	1	2	3
3	4	8	10

allOperators

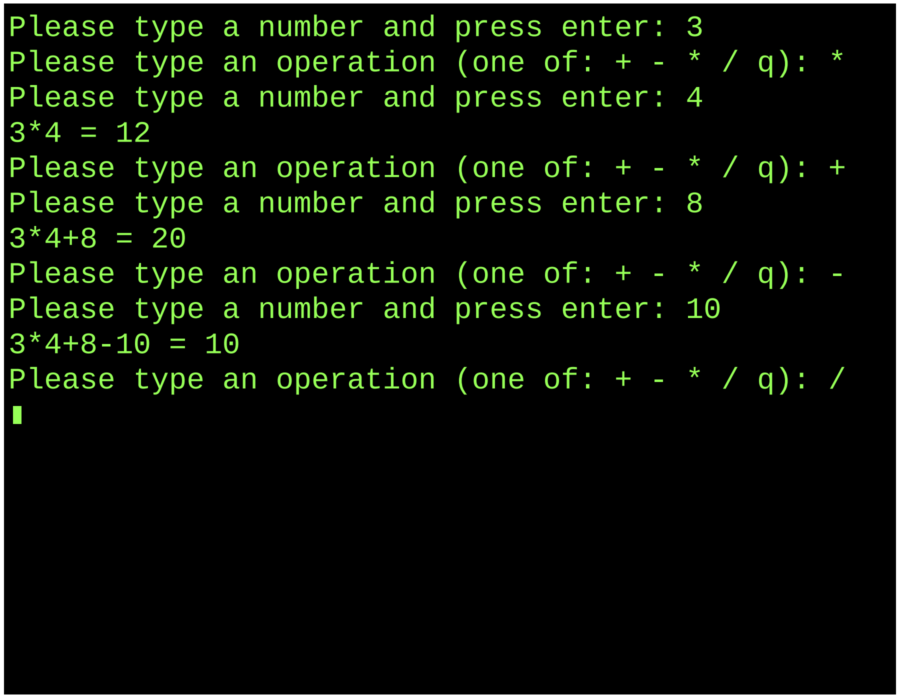
0	1	2
'*'	'+'	'-'

result

10



calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
3*4+8 = 20
Please type an operation (one of: + - * / q): -
Please type a number and press enter: 10
3*4+8-10 = 10
Please type an operation (one of: + - * / q): /
▮
```

Variables

allOperands

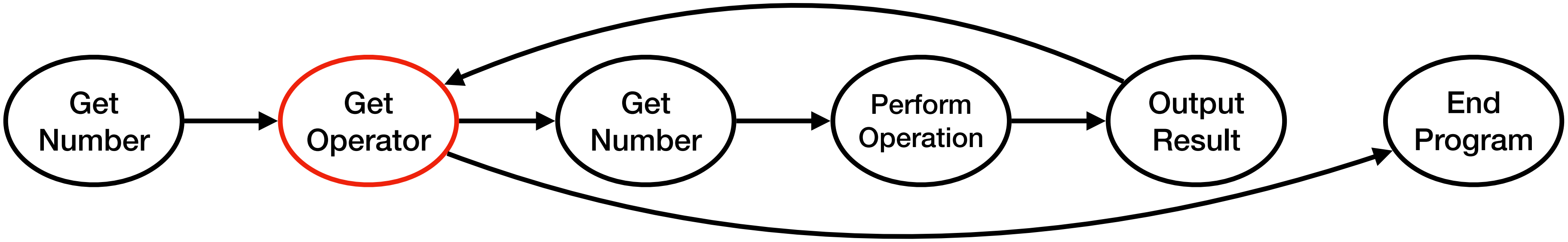
3	4	8	10
0	1	2	3

allOperators

'*'	'+'	'-'	'/ '
0	1	2	3

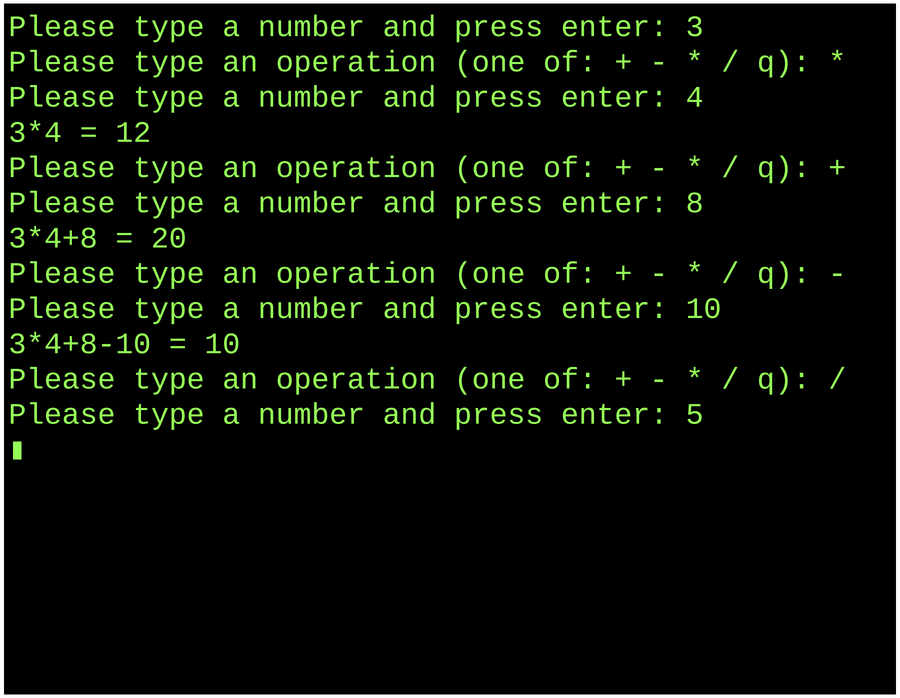
result

10



Michigan Robotics 102 - robotics102.org

calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
3*4+8 = 20
Please type an operation (one of: + - * / q): -
Please type a number and press enter: 10
3*4+8-10 = 10
Please type an operation (one of: + - * / q): /
Please type a number and press enter: 5
▮
```

Variables

allOperands

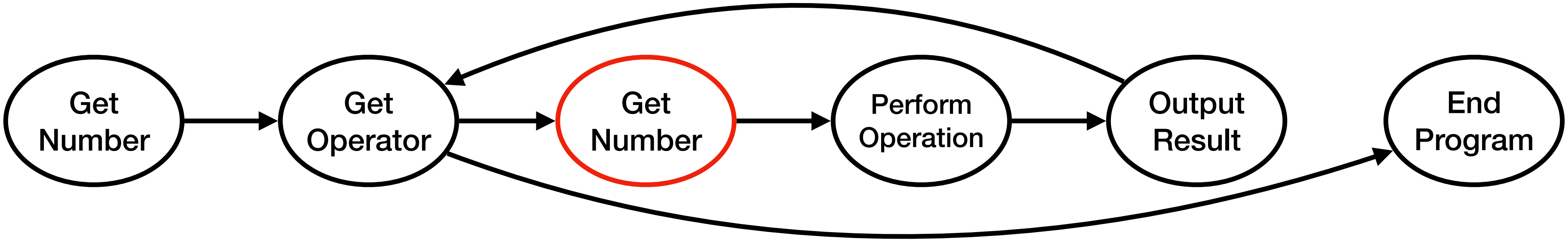
3	4	8	10	5
0	1	2	3	4

allOperators

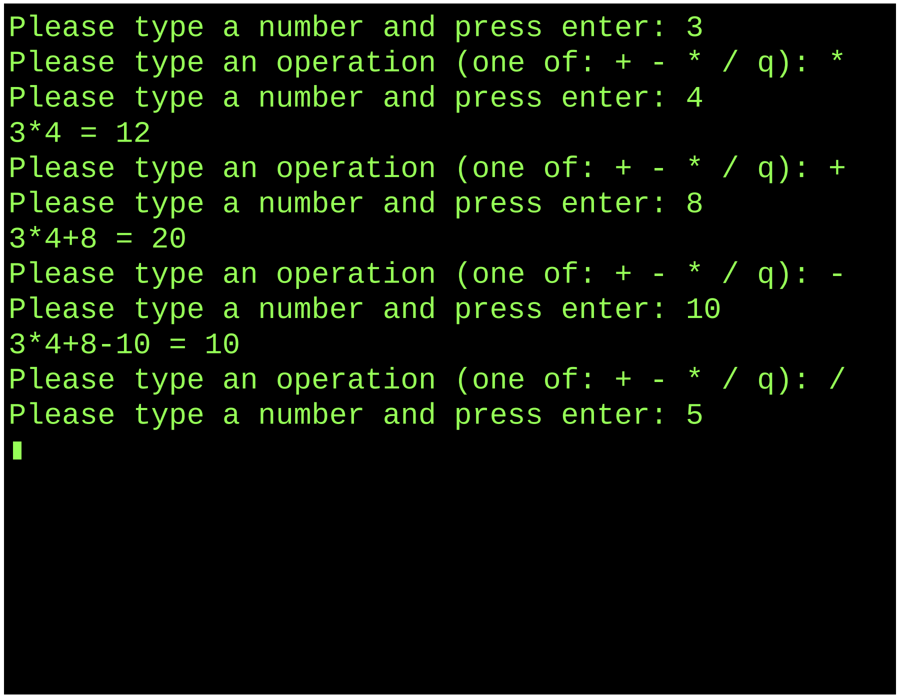
'*'	'+'	'-'	'/ '
0	1	2	3

result

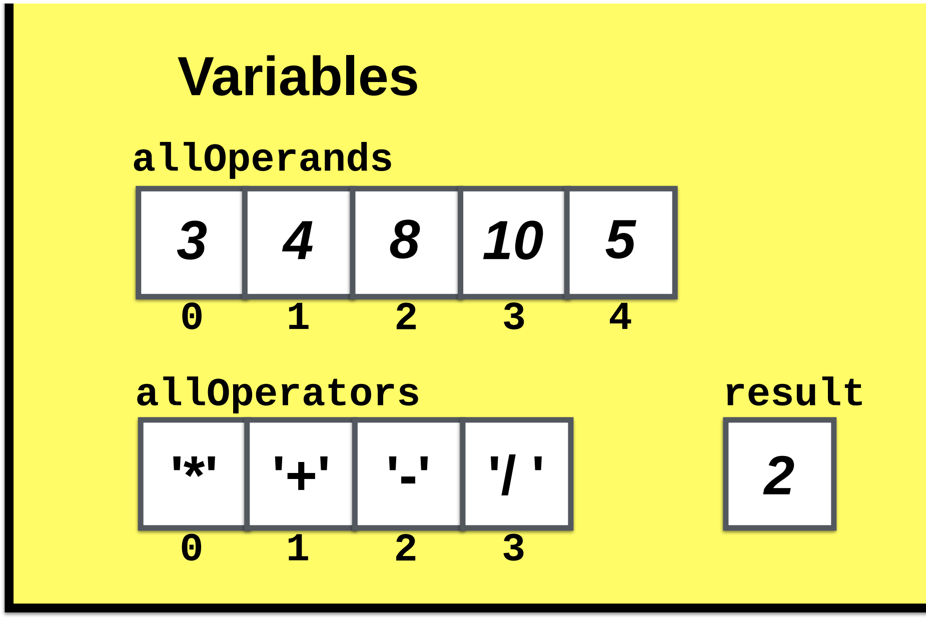
10



calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
3*4+8 = 20
Please type an operation (one of: + - * / q): -
Please type a number and press enter: 10
3*4+8-10 = 10
Please type an operation (one of: + - * / q): /
Please type a number and press enter: 5
▮
```



Variables

allOperands

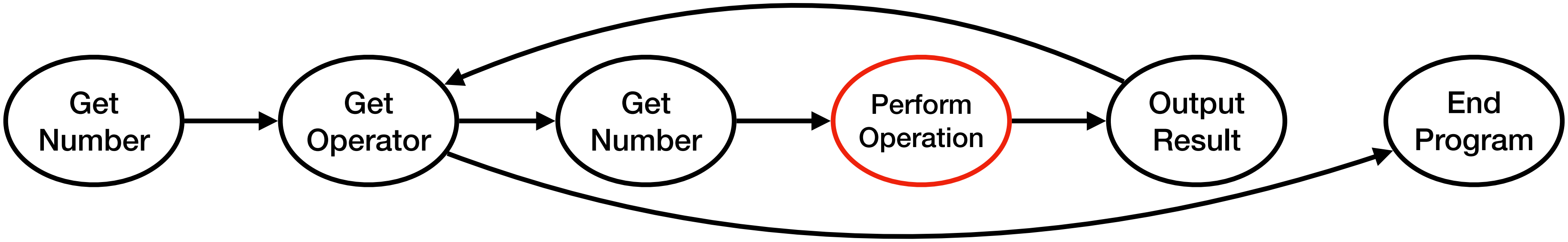
3	4	8	10	5
0	1	2	3	4

allOperators

'*'	'+'	'-'	'/ '
0	1	2	3

result

2



Get Number

Get Operator

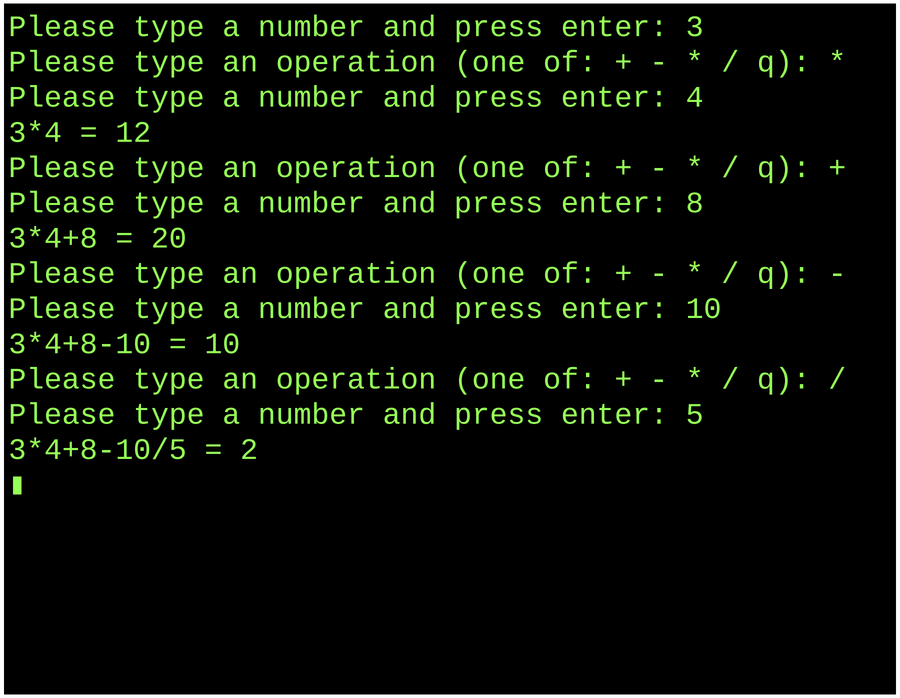
Get Number

Perform Operation

Output Result

End Program

calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
3*4+8 = 20
Please type an operation (one of: + - * / q): -
Please type a number and press enter: 10
3*4+8-10 = 10
Please type an operation (one of: + - * / q): /
Please type a number and press enter: 5
3*4+8-10/5 = 2
▮
```

Variables

allOperands

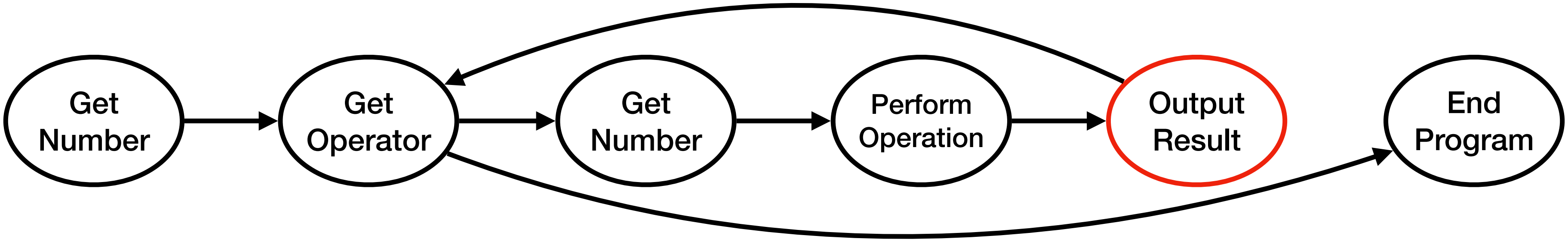
0	1	2	3	4
3	4	8	10	5

allOperators

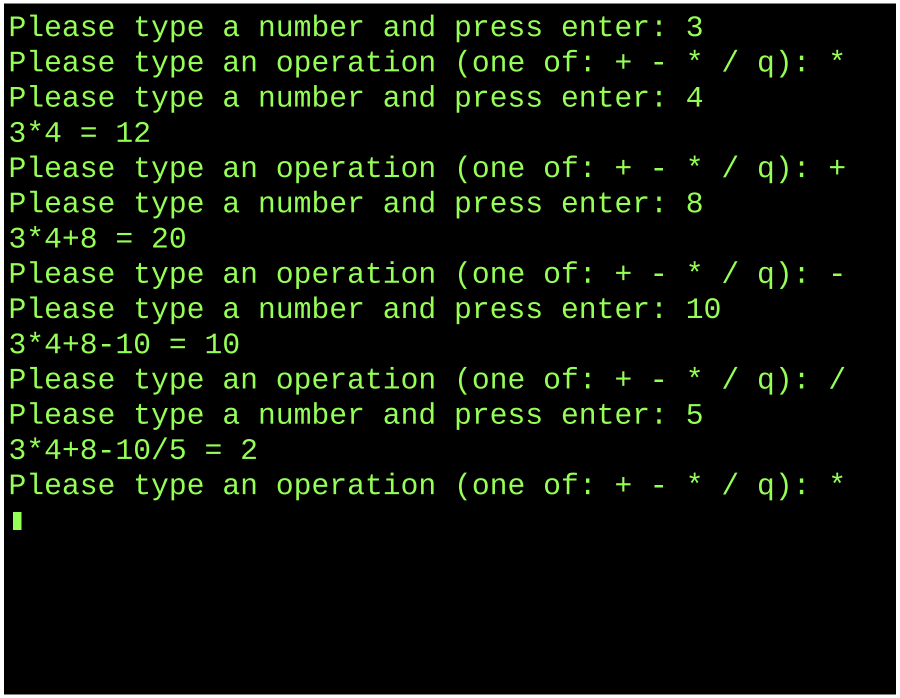
0	1	2	3
'*'	'+'	'-'	'/ '

result

2



calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
3*4+8 = 20
Please type an operation (one of: + - * / q): -
Please type a number and press enter: 10
3*4+8-10 = 10
Please type an operation (one of: + - * / q): /
Please type a number and press enter: 5
3*4+8-10/5 = 2
Please type an operation (one of: + - * / q): *
```

Variables

`allOperands`

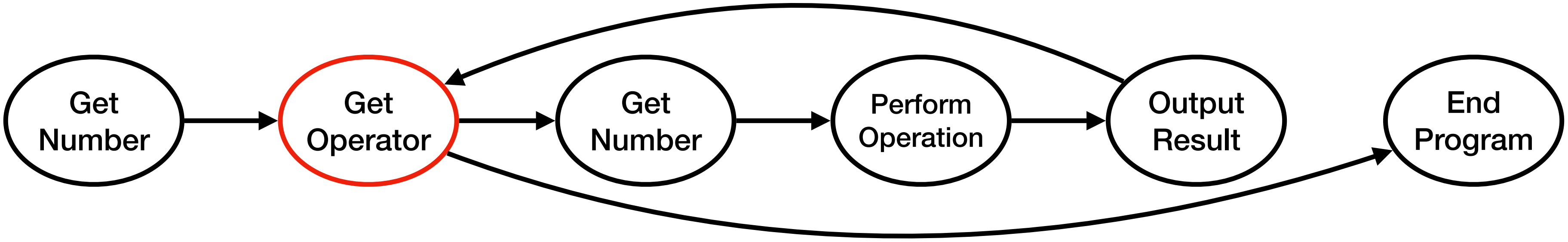
3	4	8	10	5
0	1	2	3	4

`allOperators`

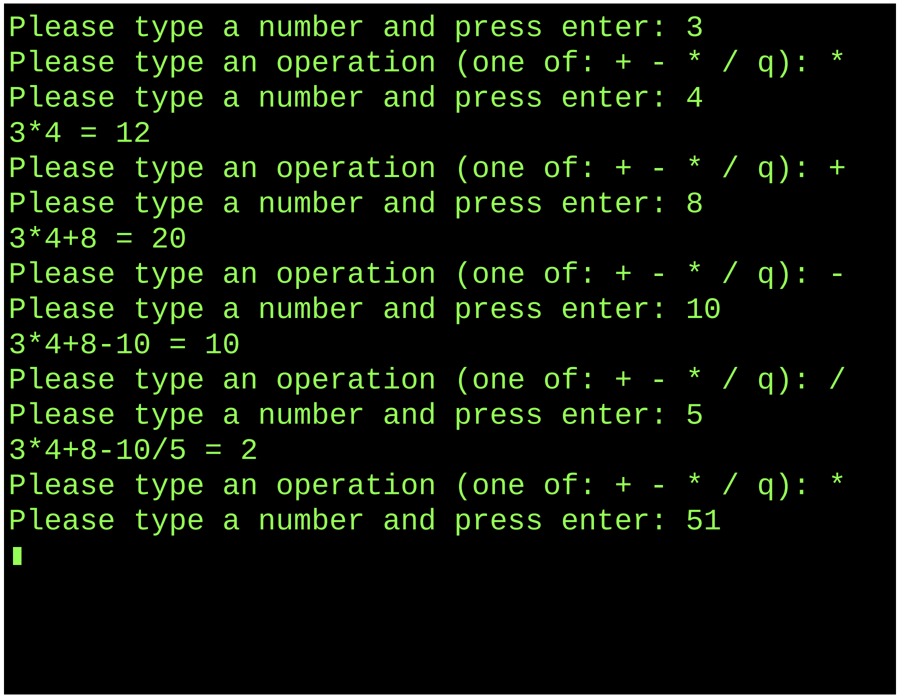
'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

`result`

2



calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
3*4+8 = 20
Please type an operation (one of: + - * / q): -
Please type a number and press enter: 10
3*4+8-10 = 10
Please type an operation (one of: + - * / q): /
Please type a number and press enter: 5
3*4+8-10/5 = 2
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 51
▮
```

Variables

allOperands

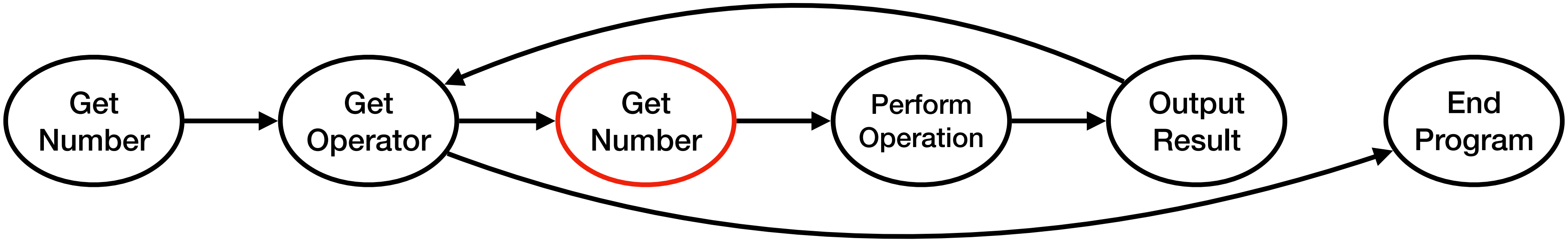
3	4	8	10	5	51
0	1	2	3	4	5

allOperators

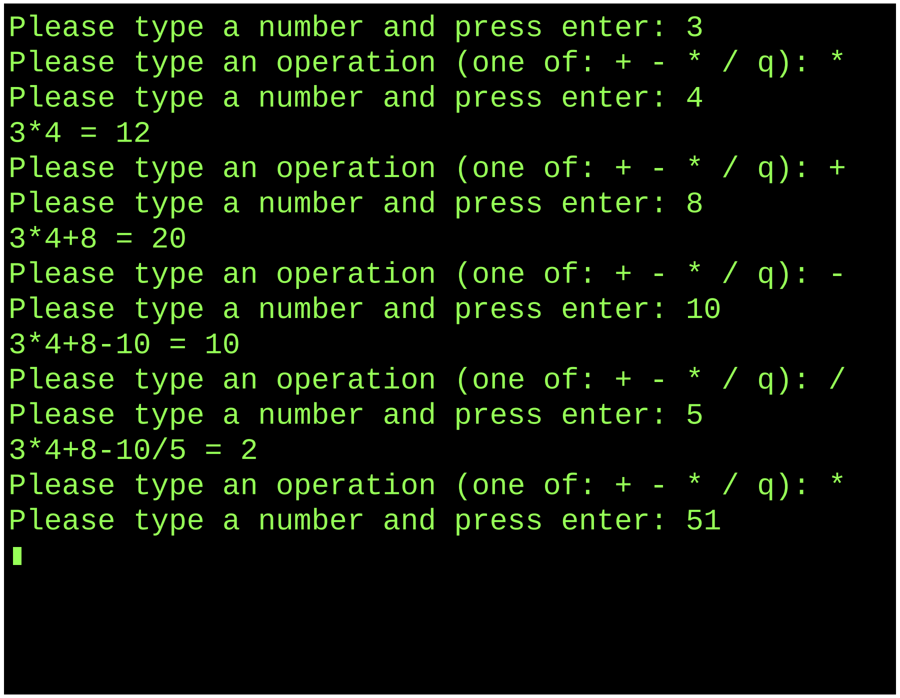
'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

result

2



calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
3*4+8 = 20
Please type an operation (one of: + - * / q): -
Please type a number and press enter: 10
3*4+8-10 = 10
Please type an operation (one of: + - * / q): /
Please type a number and press enter: 5
3*4+8-10/5 = 2
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 51
▮
```

Variables

allOperands

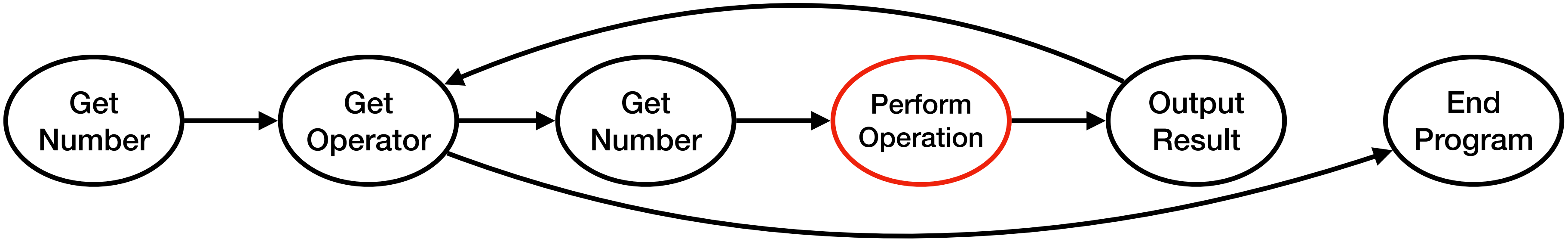
3	4	8	10	5	51
0	1	2	3	4	5

allOperators

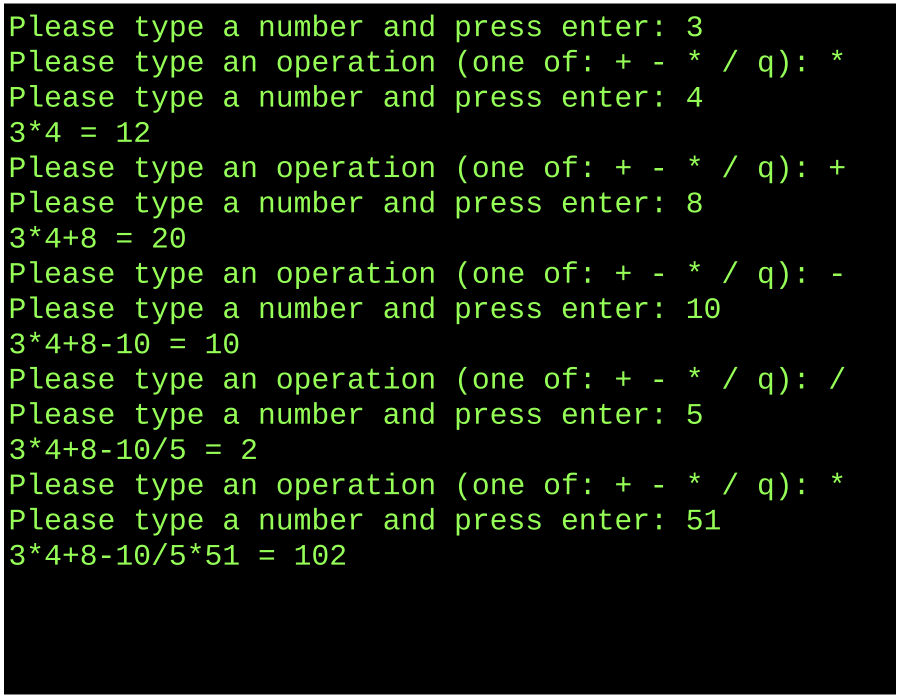
'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

result

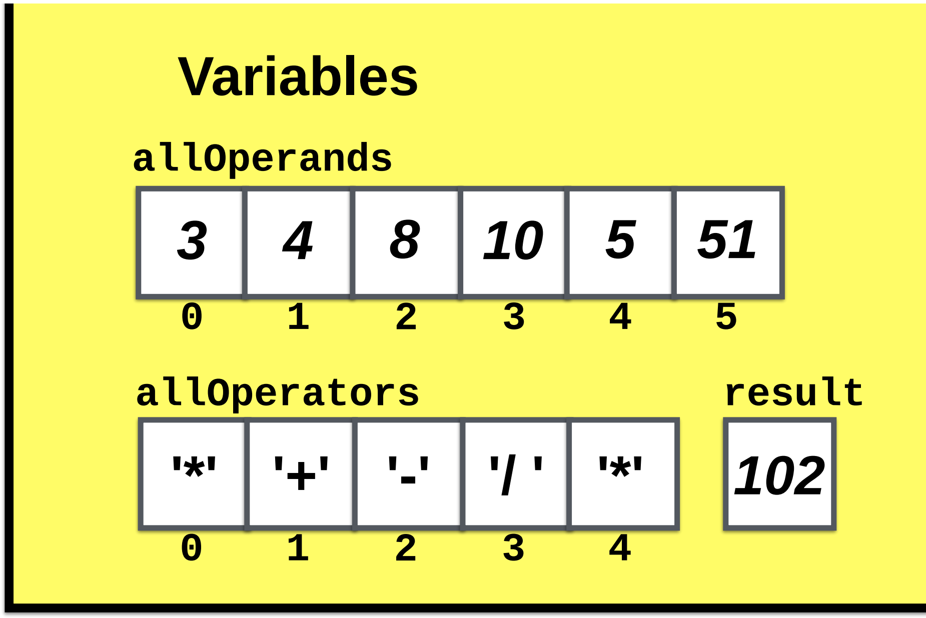
102



calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
3*4+8 = 20
Please type an operation (one of: + - * / q): -
Please type a number and press enter: 10
3*4+8-10 = 10
Please type an operation (one of: + - * / q): /
Please type a number and press enter: 5
3*4+8-10/5 = 2
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 51
3*4+8-10/5*51 = 102
```



Variables

allOperands

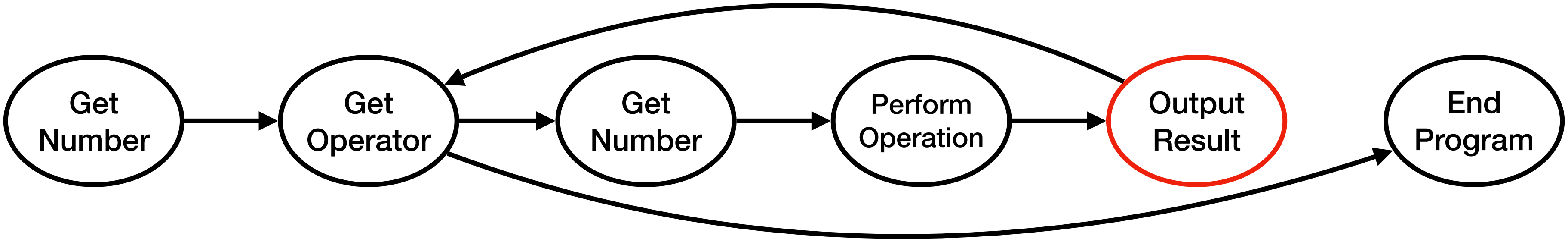
3	4	8	10	5	51
0	1	2	3	4	5

allOperators

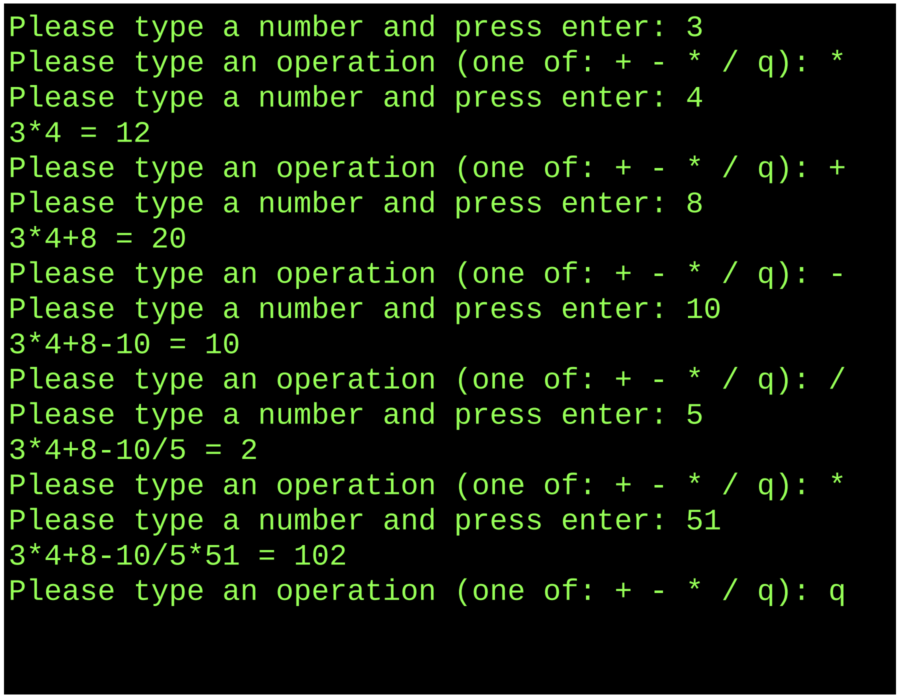
'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

result

102



calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
3*4+8 = 20
Please type an operation (one of: + - * / q): -
Please type a number and press enter: 10
3*4+8-10 = 10
Please type an operation (one of: + - * / q): /
Please type a number and press enter: 5
3*4+8-10/5 = 2
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 51
3*4+8-10/5*51 = 102
Please type an operation (one of: + - * / q): q
```

Variables

allOperands

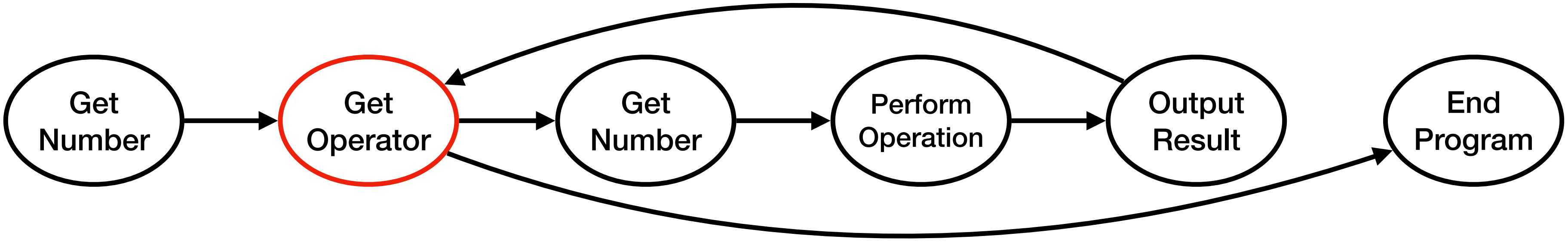
3	4	8	10	5	51
0	1	2	3	4	5

allOperators

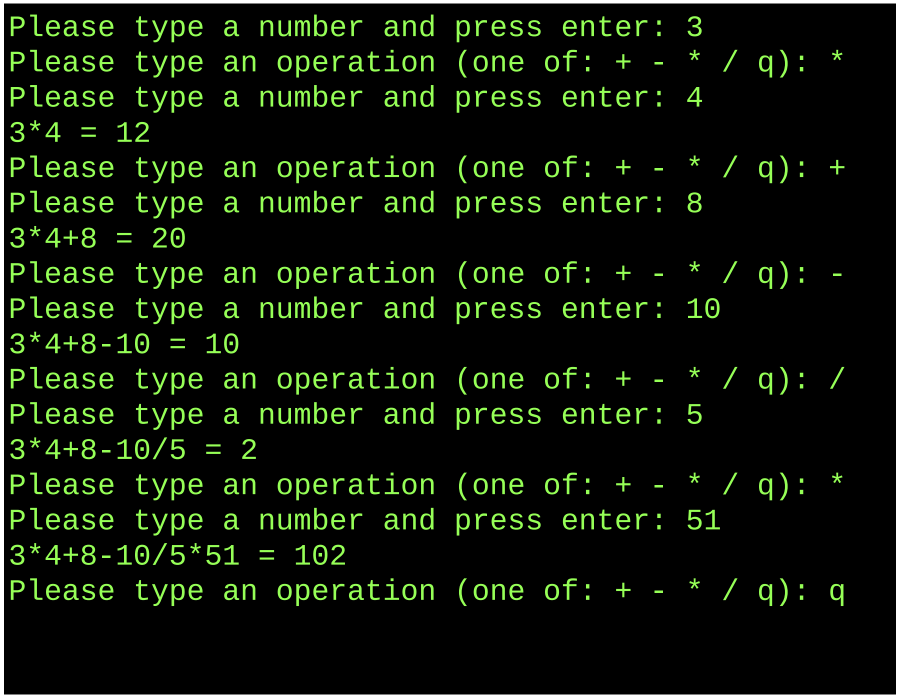
'*'	'+'	'-'	'/ '	'*'	‘q'
0	1	2	3	4	5

result

102



calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
3*4+8 = 20
Please type an operation (one of: + - * / q): -
Please type a number and press enter: 10
3*4+8-10 = 10
Please type an operation (one of: + - * / q): /
Please type a number and press enter: 5
3*4+8-10/5 = 2
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 51
3*4+8-10/5*51 = 102
Please type an operation (one of: + - * / q): q
```

Variables

allOperands

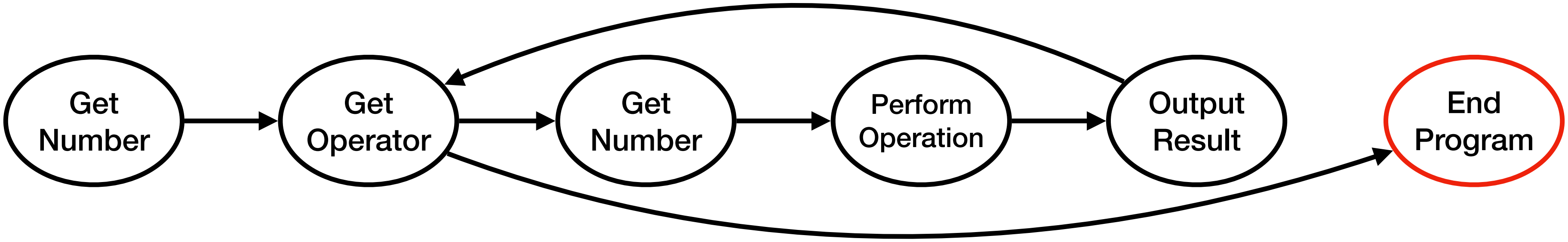
3	4	8	10	5	51
0	1	2	3	4	5

allOperators

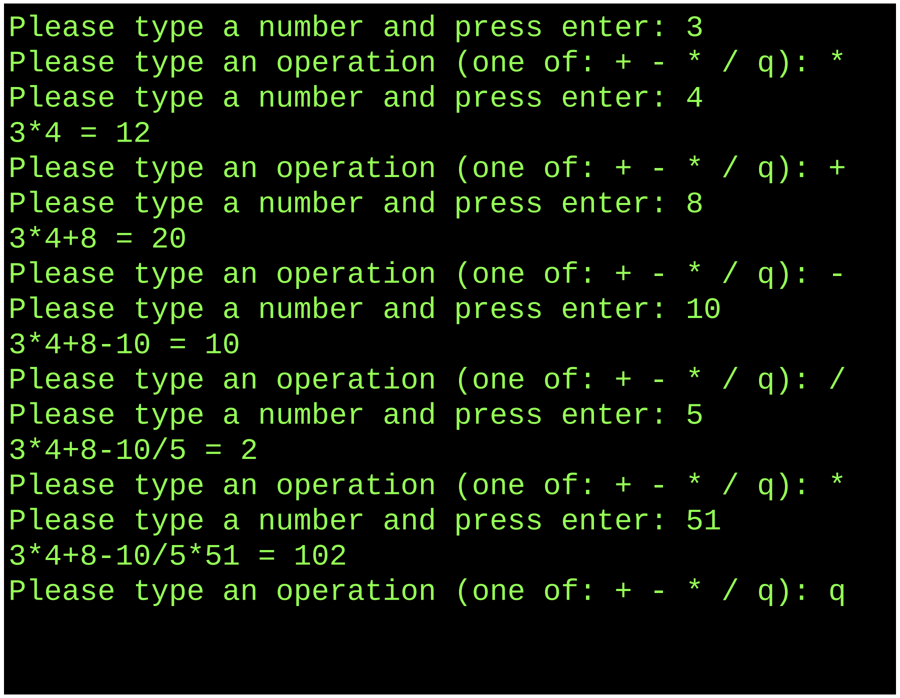
'*'	'+'	'-'	'/ '	'*'	‘q'
0	1	2	3	4	5

result

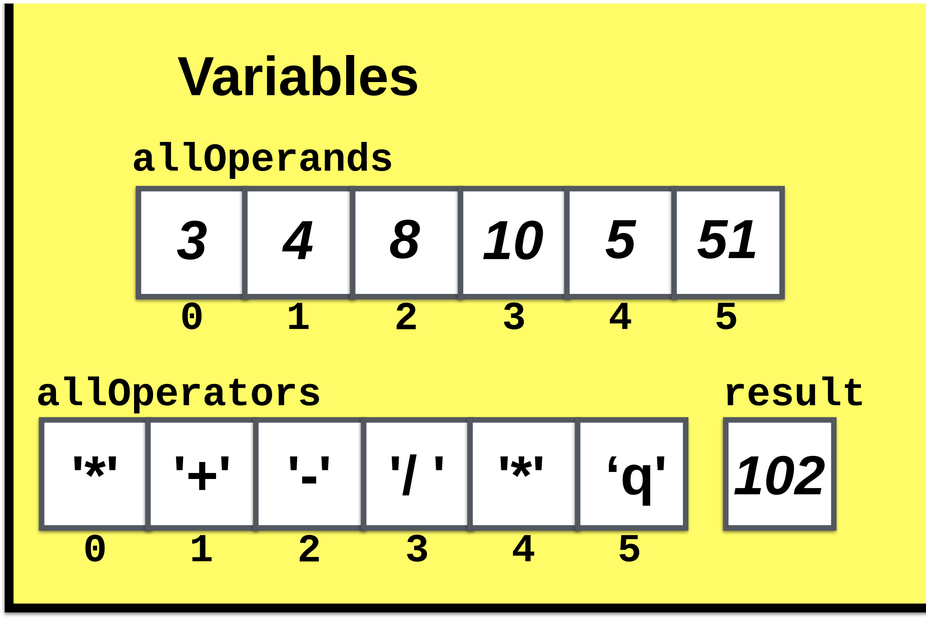
102



calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
3*4+8 = 20
Please type an operation (one of: + - * / q): -
Please type a number and press enter: 10
3*4+8-10 = 10
Please type an operation (one of: + - * / q): /
Please type a number and press enter: 5
3*4+8-10/5 = 2
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 51
3*4+8-10/5*51 = 102
Please type an operation (one of: + - * / q): q
```



Variables

allOperands

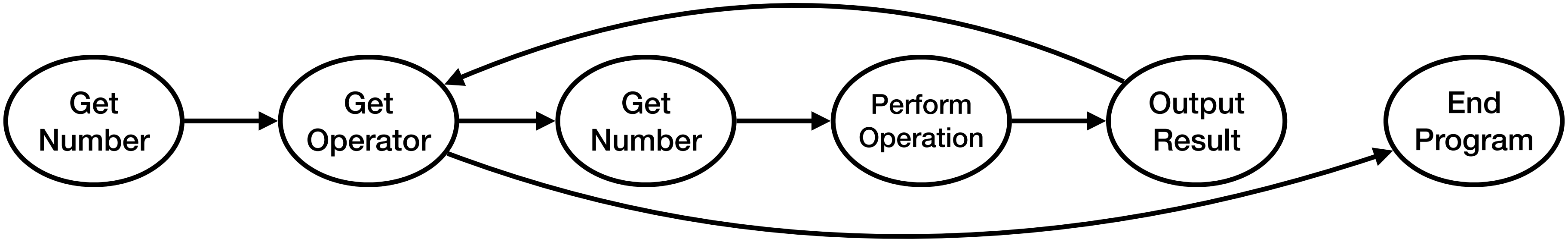
3	4	8	10	5	51
0	1	2	3	4	5

allOperators

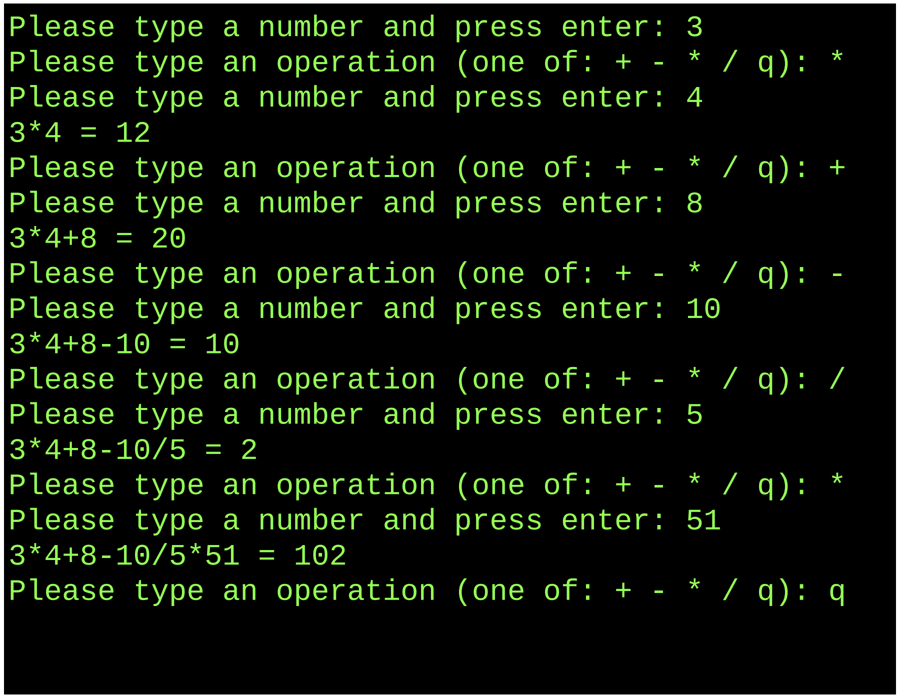
'*'	'+'	'-'	'/ '	'*'	‘q'
0	1	2	3	4	5

result

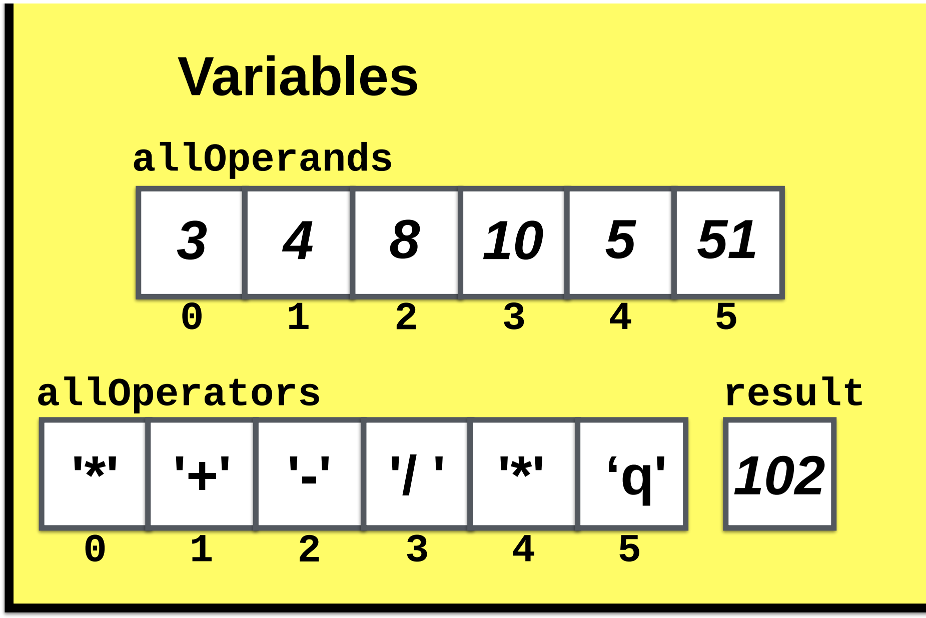
102



calculator61



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
3*4 = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
3*4+8 = 20
Please type an operation (one of: + - * / q): -
Please type a number and press enter: 10
3*4+8-10 = 10
Please type an operation (one of: + - * / q): /
Please type a number and press enter: 5
3*4+8-10/5 = 2
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 51
3*4+8-10/5*51 = 102
Please type an operation (one of: + - * / q): q
```



Variables

allOperands

3	4	8	10	5	51
0	1	2	3	4	5

allOperators

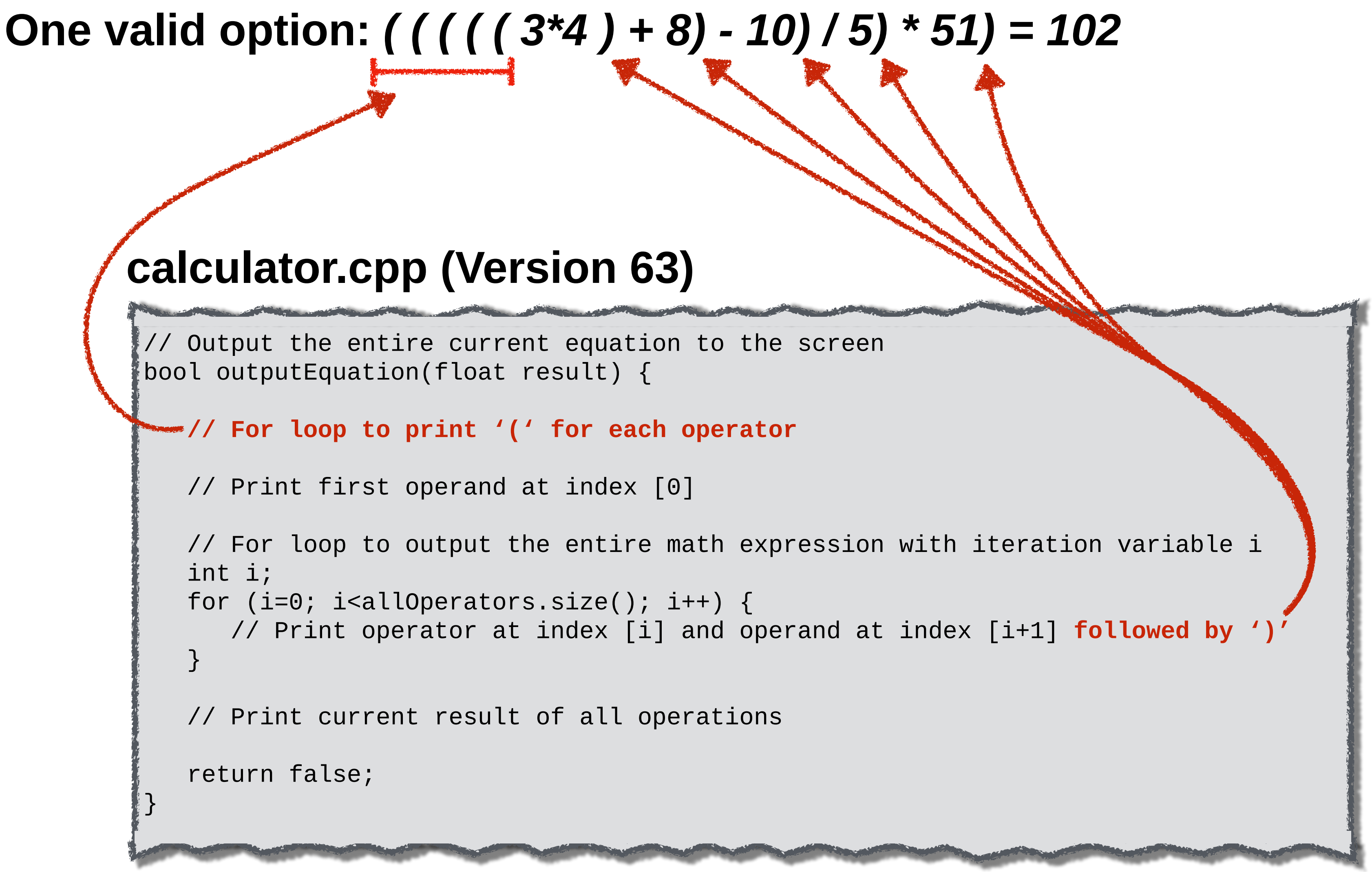
'*'	'+'	'-'	'/ '	'*'	‘q'
0	1	2	3	4	5

result

102

Output proper infix equation ?

One valid option: ***((((((3*4) + 8) - 10) / 5) * 51) = 102***



One valid option: *(((((3*4) + 8) - 10) / 5) * 51) = 102*

calculator.cpp (Version 63)

```
// Output the entire current equation to the screen
bool outputEquation(float result) {

   // For loop to print ‘(‘ for each operator

   // Print first operand at index [0]

   // For loop to output the entire math expression with iteration variable i
   int i;
   for (i=0; i<allOperators.size(); i++) {
      // Print operator at index [i] and operand at index [i+1] followed by ‘)’
   }

   // Print current result of all operations

   return false;
}
```

[redacted]

calculator.cpp (Version 63)

```
// Output the entire current equation to the screen
bool outputEquation(float result) {

   // For loop to print ‘(‘ for each operator

   // Print first operand at index [0]

   // For loop to output the entire math expression with iteration variable i
   int i;
   for (i=0; i<allOperators.size(); i++) {
      // Print operator at index [i] and operand at index [i+1] followed by ‘)’
   }

   // Print current result of all operations

   return false;
}
```

Let’s follow the program from this point

Variables

allOperands

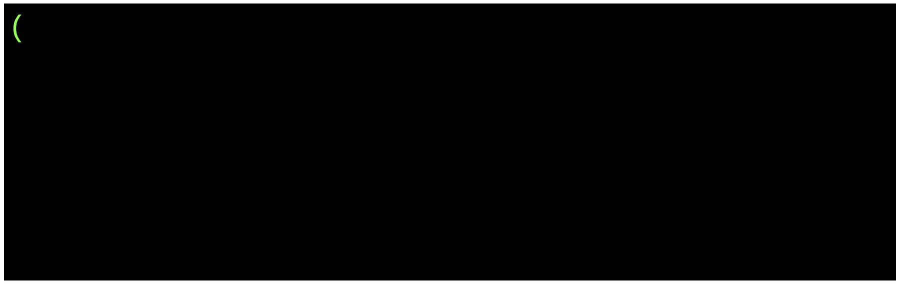
3	4	8	10	5	51
0	1	2	3	4	5

allOperators

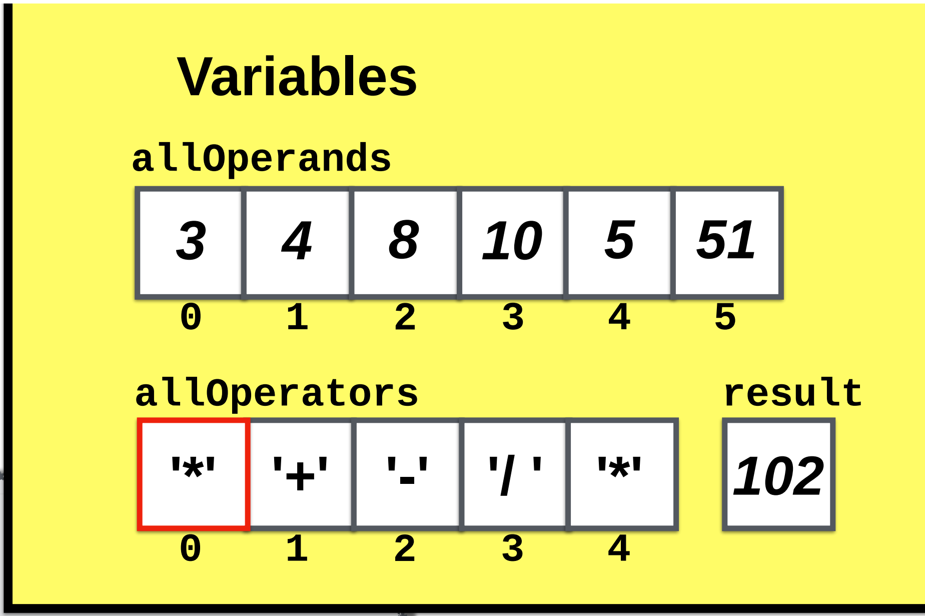
'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

result

102



```
(
```



Variables

allOperands

3	4	8	10	5	51
0	1	2	3	4	5

allOperators

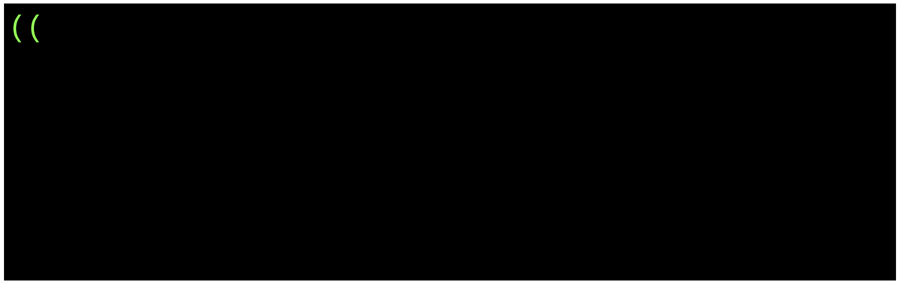
'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

result

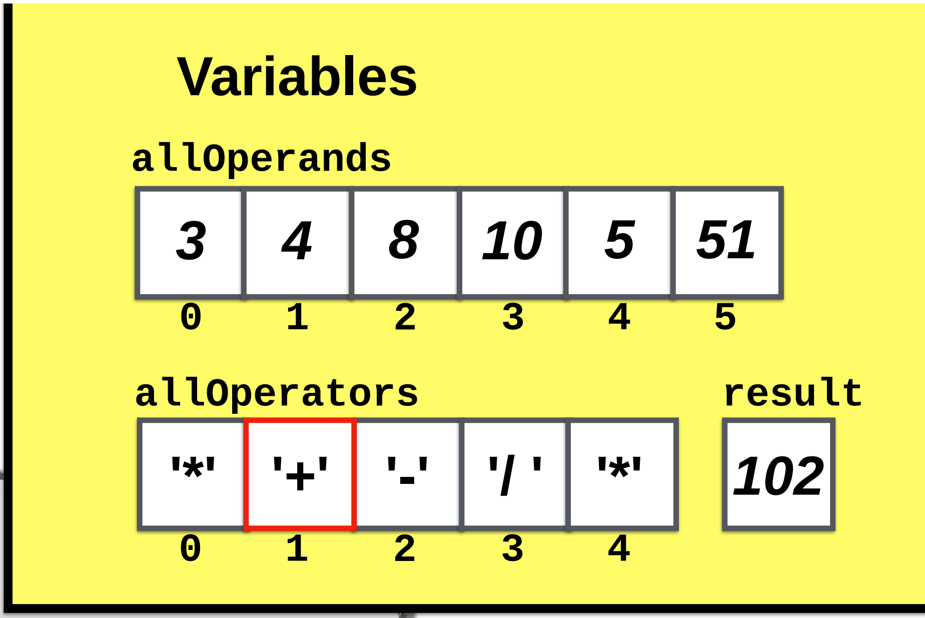
102

calculator.cpp (Version 63)

```
// Output the entire current equation to the screen
bool outputEquation(float result) {

   // For loop to print ‘(‘ for each operator

   // Print first operand at index [0]

   // For loop to output the entire math expression with iteration variable i
   int i;
   for (i=0; i<allOperators.size(); i++) {
      // Print operator at index [i] and operand at index [i+1] followed by ‘)’
   }

   // Print current result of all operations

   return false;
}
```



```
((
```



Variables

allOperands

3	4	8	10	5	51
0	1	2	3	4	5

allOperators

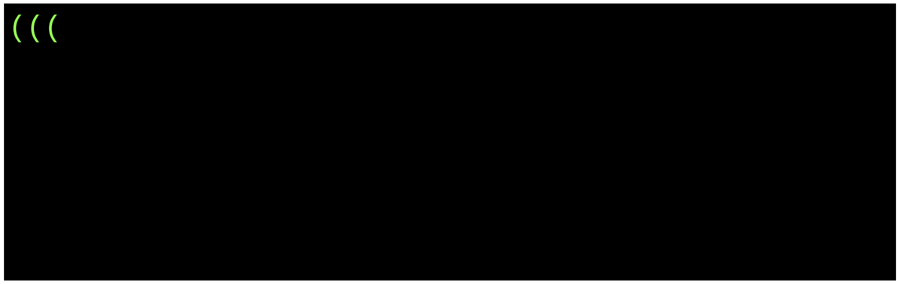
'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

result

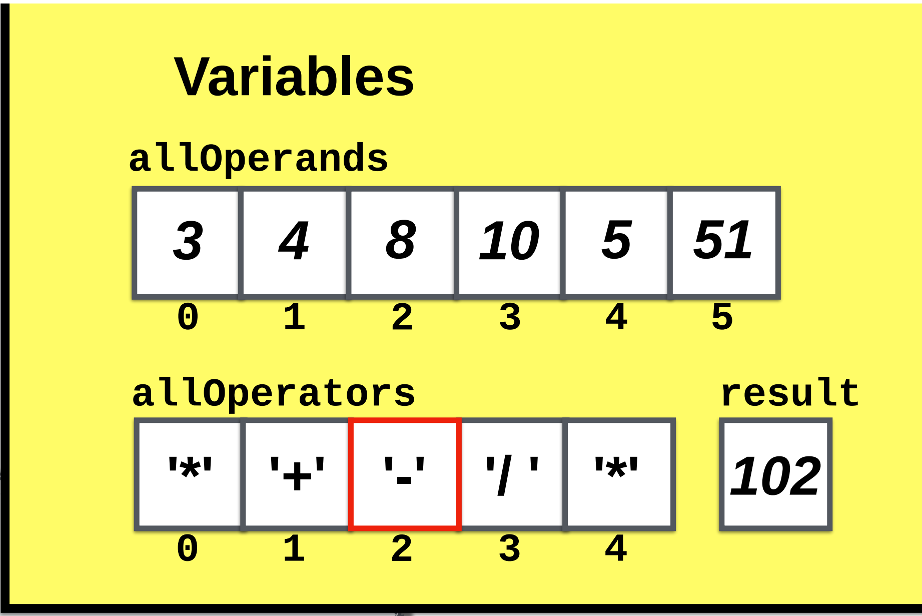
102

calculator.cpp (Version 63)

```
// Output the entire current equation to the screen
bool outputEquation(float result) {

   // For loop to print ‘(‘ for each operator

   // Print first operand at index [0]

   // For loop to output the entire math expression with iteration variable i
   int i;
   for (i=0; i<allOperators.size(); i++) {
      // Print operator at index [i] and operand at index [i+1] followed by ‘)’
   }

   // Print current result of all operations

   return false;
}
```



```
(((
```



Variables

allOperands

3	4	8	10	5	51
0	1	2	3	4	5

allOperators

'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

result

102

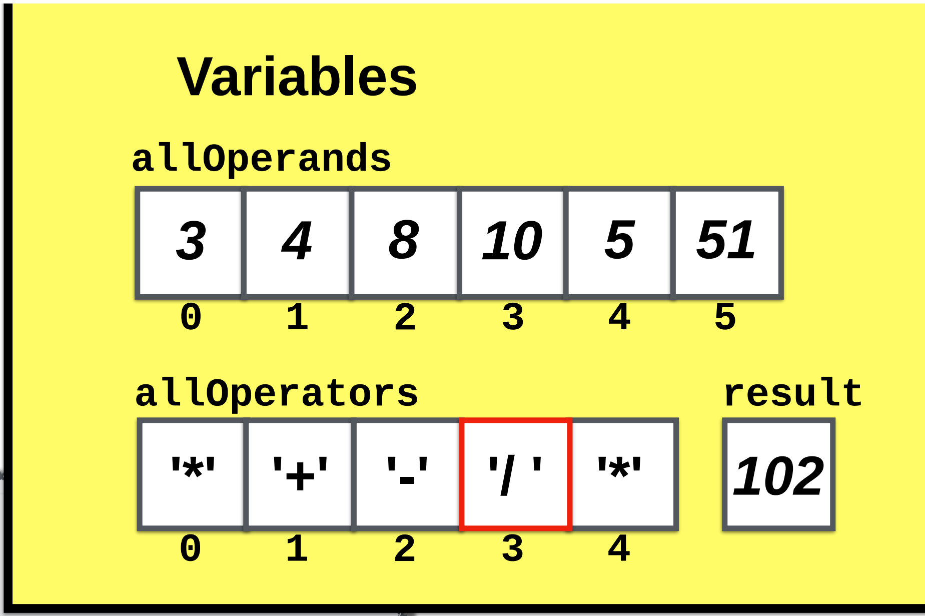
calculator.cpp (Version 63)

```
// Output the entire current equation to the screen
bool outputEquation(float result) {

   // For loop to print '(' for each operator

   // Print first operand at index [0]

   // For loop to output the entire math expression with iteration variable i
   int i;
   for (i=0; i<allOperators.size(); i++) {
      // Print operator at index [i] and operand at index [i+1] followed by ')'
   }

   // Print current result of all operations

   return false;
}
```

```
((((
```

calculator.cpp (Version 63)

```
// Output the entire current equation to the screen
bool outputEquation(float result) {

   // For loop to print ‘(‘ for each operator

   // Print first operand at index [0]

   // For loop to output the entire math expression with iteration variable i
   int i;
   for (i=0; i<allOperators.size(); i++) {
      // Print operator at index [i] and operand at index [i+1] followed by ‘)’
   }

   // Print current result of all operations

   return false;
}
```



Variables

allOperands

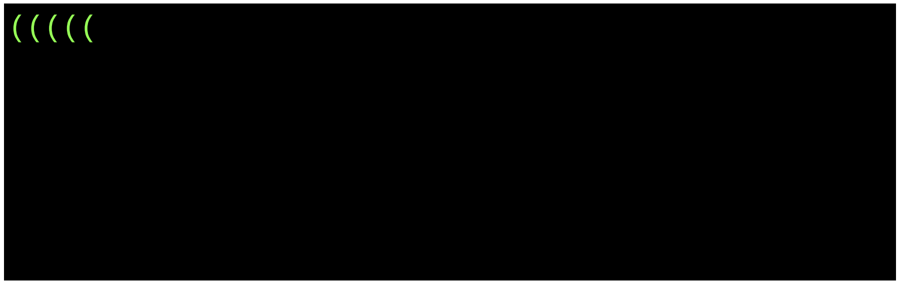
3	4	8	10	5	51
0	1	2	3	4	5

allOperators

'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

result

102



```
(((((
```

calculator.cpp (Version 63)

```
// Output the entire current equation to the screen
bool outputEquation(float result) {

   // For loop to print ‘(‘ for each operator

   // Print first operand at index [0]

   // For loop to output the entire math expression with iteration variable i
   int i;
   for (i=0; i<allOperators.size(); i++) {
      // Print operator at index [i] and operand at index [i+1] followed by ‘)’
   }

   // Print current result of all operations

   return false;
}
```

Variables

allOperands

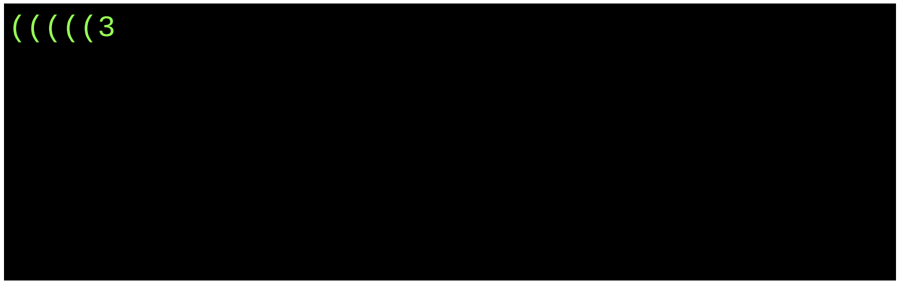
3	4	8	10	5	51
0	1	2	3	4	5

allOperators

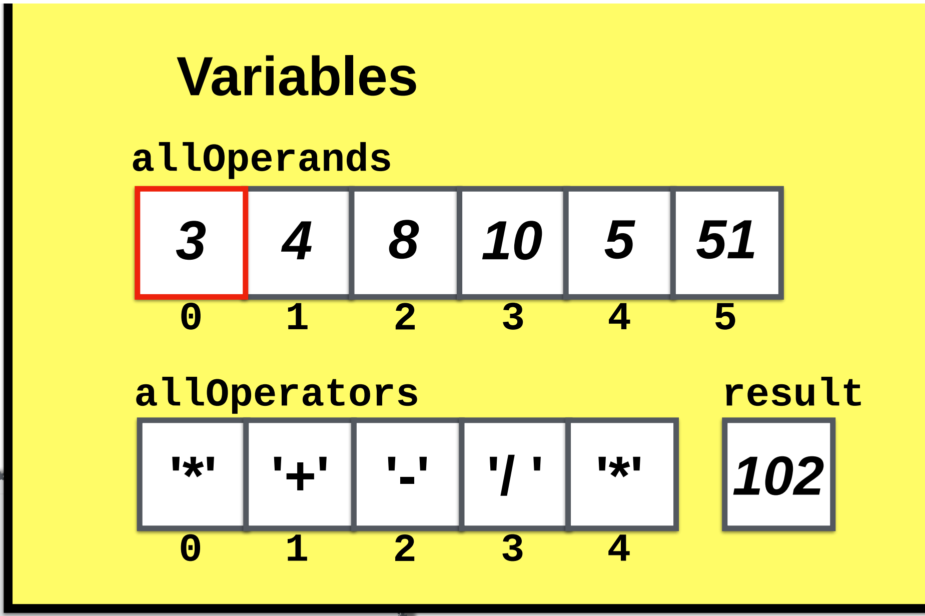
'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

result

102



```
(((((3
```



Variables

allOperands

3	4	8	10	5	51
0	1	2	3	4	5

allOperators

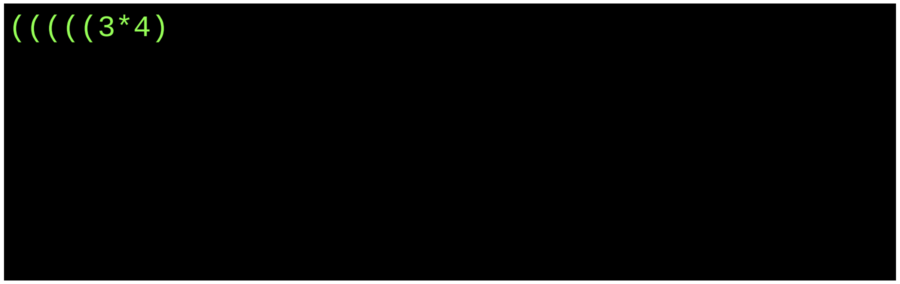
'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

result

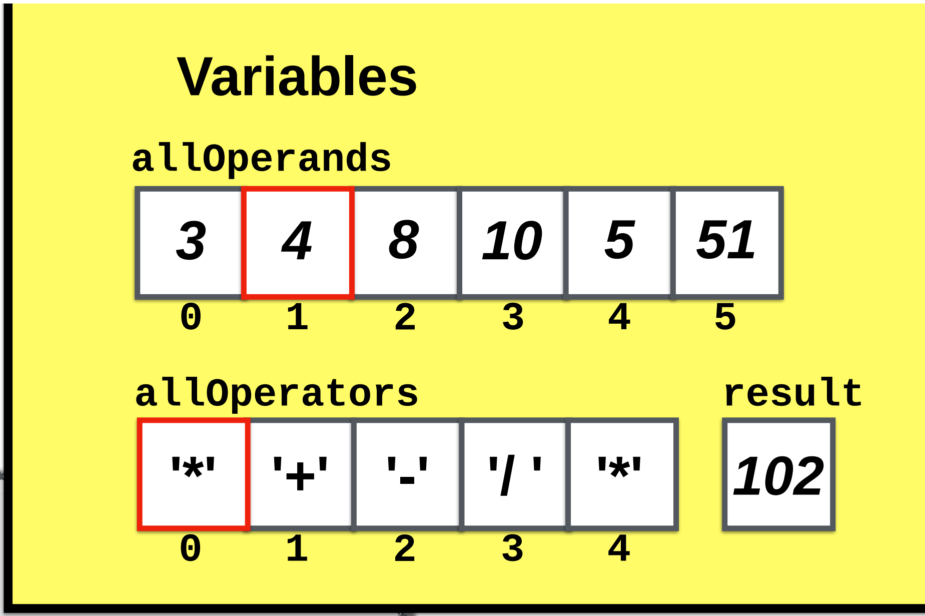
102

calculator.cpp (Version 63)

```
// Output the entire current equation to the screen
bool outputEquation(float result) {

   // For loop to print ‘(‘ for each operator

   // Print first operand at index [0]

   // For loop to output the entire math expression with iteration variable i
   int i;
   for (i=0; i<allOperators.size(); i++) {
      // Print operator at index [i] and operand at index [i+1] followed by ‘)’
   }

   // Print current result of all operations

   return false;
}
```



```
(((((3*4)
```



Variables

allOperands

3	4	8	10	5	51
0	1	2	3	4	5

allOperators

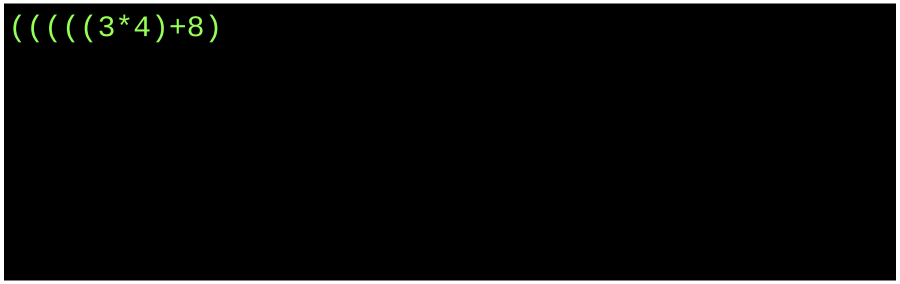
'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

result

102

calculator.cpp (Version 63)

```
// Output the entire current equation to the screen
bool outputEquation(float result) {

   // For loop to print ‘(‘ for each operator

   // Print first operand at index [0]

   // For loop to output the entire math expression with iteration variable i
   int i;
   for (i=0; i<allOperators.size(); i++) {
      // Print operator at index [i] and operand at index [i+1] followed by ‘)’
   }

   // Print current result of all operations

   return false;
}
```



```
(((((3*4)+8)
```

Variables

allOperands

3	4	8	10	5	51
0	1	2	3	4	5

allOperators

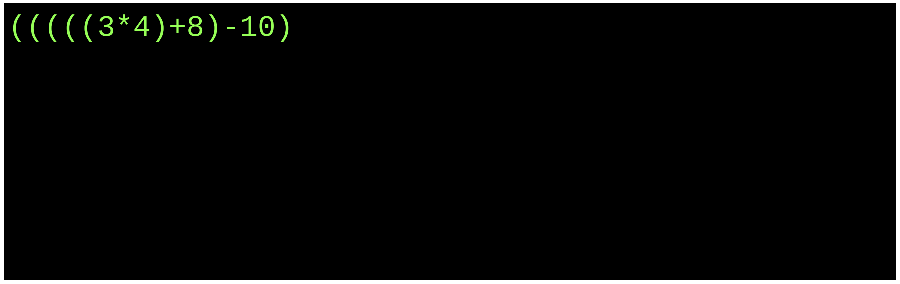
'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

result

102

calculator.cpp (Version 63)

```
// Output the entire current equation to the screen
bool outputEquation(float result) {

   // For loop to print ‘(‘ for each operator

   // Print first operand at index [0]

   // For loop to output the entire math expression with iteration variable i
   int i;
   for (i=0; i<allOperators.size(); i++) {
      // Print operator at index [i] and operand at index [i+1] followed by ‘)’
   }

   // Print current result of all operations

   return false;
}
```



```
(((((3*4)+8)-10)
```

calculator.cpp (Version 63)

```
// Output the entire current equation to the screen
bool outputEquation(float result) {

   // For loop to print ‘(‘ for each operator

   // Print first operand at index [0]

   // For loop to output the entire math expression with iteration variable i
   int i;
   for (i=0; i<allOperators.size(); i++) {
      // Print operator at index [i] and operand at index [i+1] followed by ‘)’
   }

   // Print current result of all operations

   return false;
}
```

Variables

allOperands

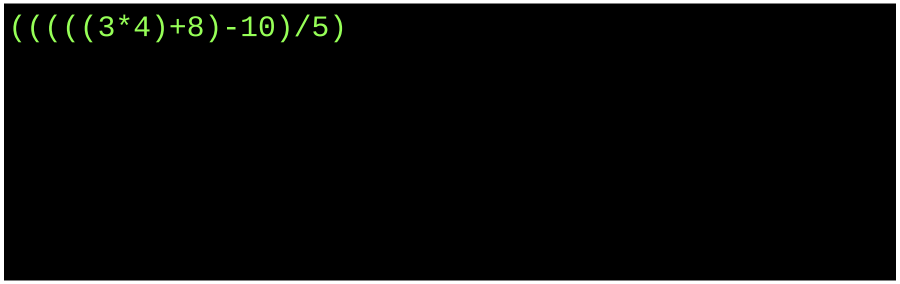
3	4	8	10	5	51
0	1	2	3	4	5

allOperators

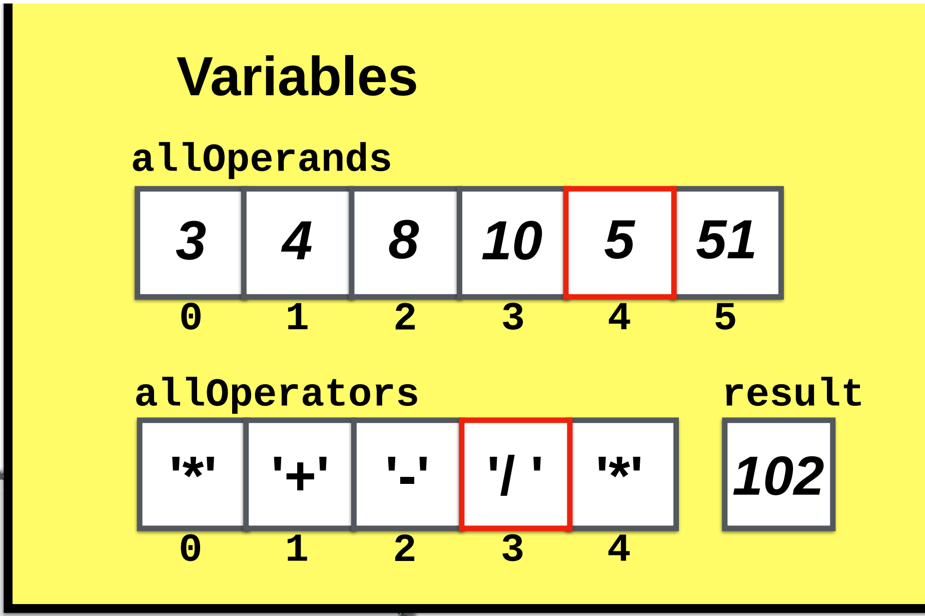
'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

result

102



```
(((((3*4)+8)-10)/5)
```



Variables

allOperands

3	4	8	10	5	51
0	1	2	3	4	5

allOperators

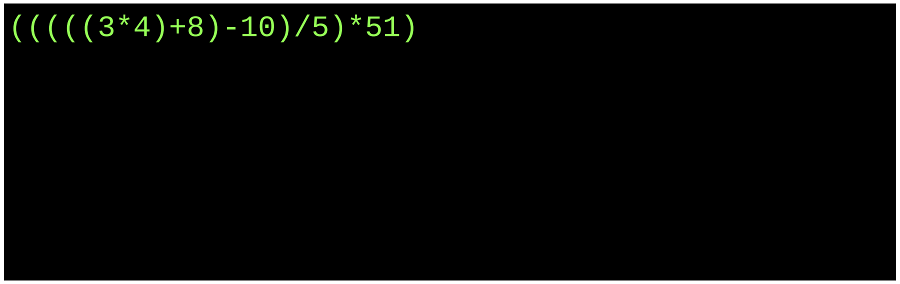
'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

result

102

calculator.cpp (Version 63)

```
// Output the entire current equation to the screen
bool outputEquation(float result) {

   // For loop to print ‘(‘ for each operator

   // Print first operand at index [0]

   // For loop to output the entire math expression with iteration variable i
   int i;
   for (i=0; i<allOperators.size(); i++) {
      // Print operator at index [i] and operand at index [i+1] followed by ‘)’
   }

   // Print current result of all operations

   return false;
}
```



```
(((((3*4)+8)-10)/5)*51)
```

calculator.cpp (Version 63)

```
// Output the entire current equation to the screen
bool outputEquation(float result) {

   // For loop to print ‘(‘ for each operator

   // Print first operand at index [0]

   // For loop to output the entire math expression with iteration variable i
   int i;
   for (i=0; i<allOperators.size(); i++) {
      // Print operator at index [i] and operand at index [i+1] followed by ‘)’
   }

   // Print current result of all operations

   return false;
}
```

Variables

allOperands

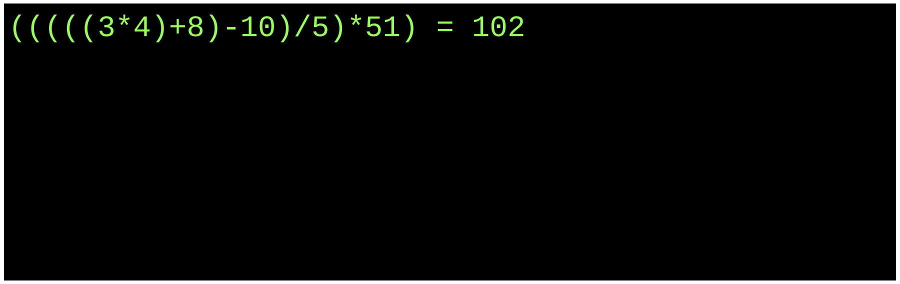
3	4	8	10	5	51
0	1	2	3	4	5

allOperators

'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

result

102



```
(((((3*4)+8)-10)/5)*51) = 102
```

calculator.cpp (Version 63)

```
// Output the entire current equation to the screen
bool outputEquation(float result) {

   // For loop to print ‘(‘ for each operator

   // Print first operand at index [0]

   // For loop to output the entire math expression with iteration variable i
   int i;
   for (i=0; i<allOperators.size(); i++) {
      // Print operator at index [i] and operand at index [i+1] followed by ‘)’
   }

   // Print current result of all operations

   return false;
}
```

Variables

allOperands

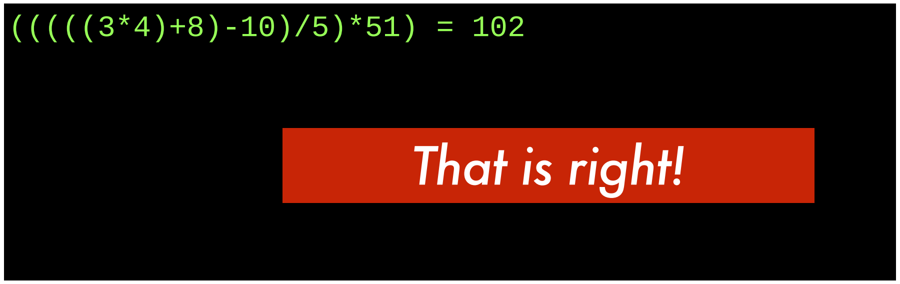
3	4	8	10	5	51
0	1	2	3	4	5

allOperators

'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

result

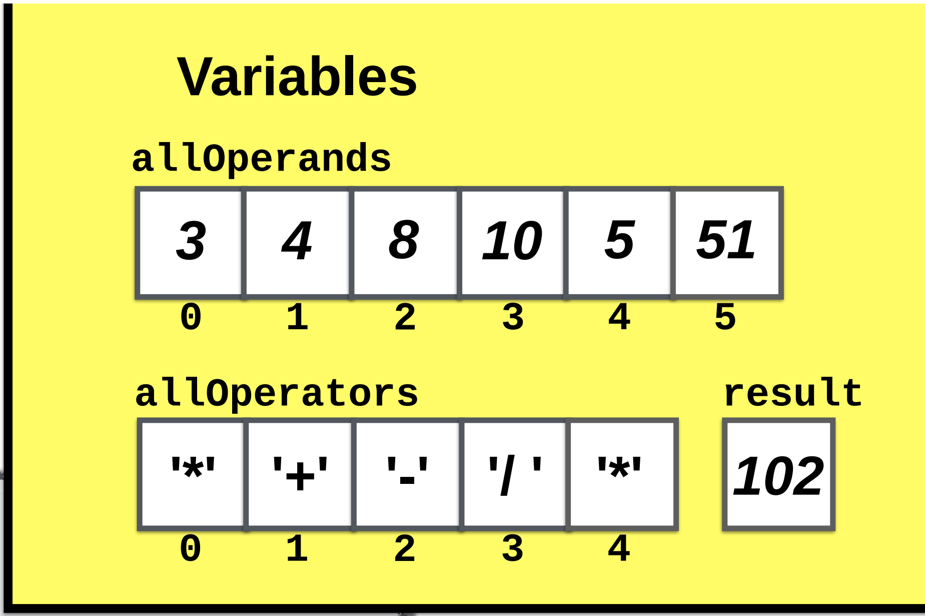
102



```
(((((3*4)+8)-10)/5)*51) = 102
```

calculator.cpp (Version 63)

```
// Output the entire current equation to the screen
bool outputEquation(float result) {

   // For loop to print ‘(‘ for each operator

   // Print first operand at index [0]

   // For loop to output the entire math expression with iteration variable i
   int i;
   for (i=0; i<allOperators.size(); i++) {
      // Print operator at index [i] and operand at index [i+1] followed by ‘)’
   }

   // Print current result of all operations

   return false;
}
```



Variables

allOperands

3	4	8	10	5	51
0	1	2	3	4	5

allOperators

'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

result

102

Our calculator is not done yet

Can we keep a history of operations?

Can we undo the last operation?

This `3*4+8-10/5*51 = 102` does not look right

Our calculator is not done yet

✓ Our calculator keeps a history of operations

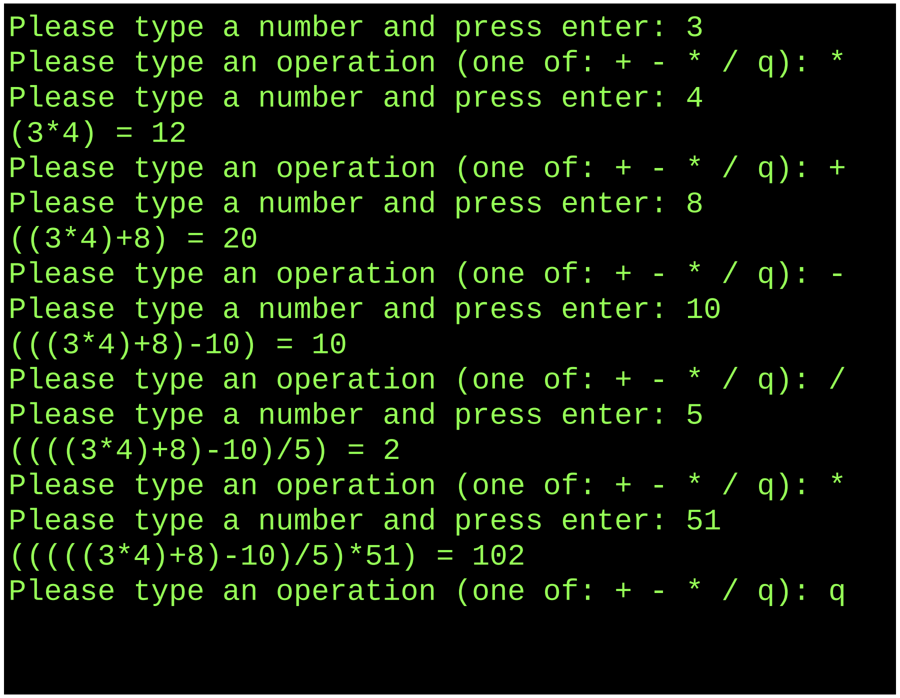
Can we undo the last operation?

✓ This `(((((3*4)+8)-10)/5)*51) = 102` looks right

Our calculator is not done yet

Can we undo the last operation?

calculator63



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / q): q
```

Variables

`allOperands`

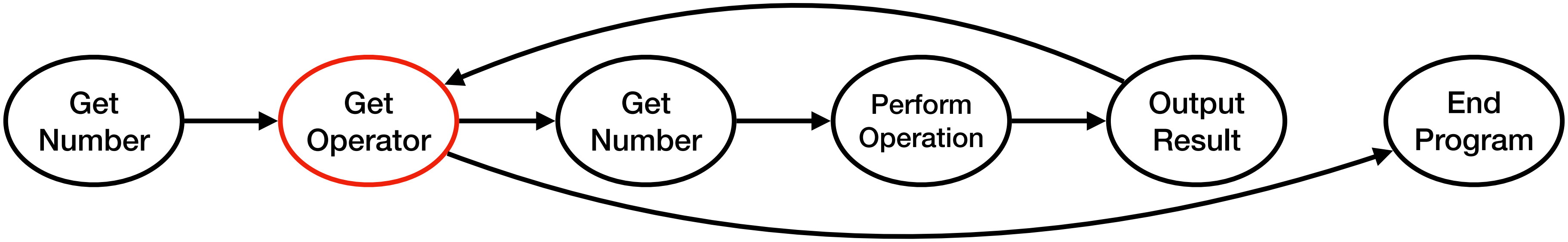
3	4	8	10	5	51
0	1	2	3	4	5

`allOperators`

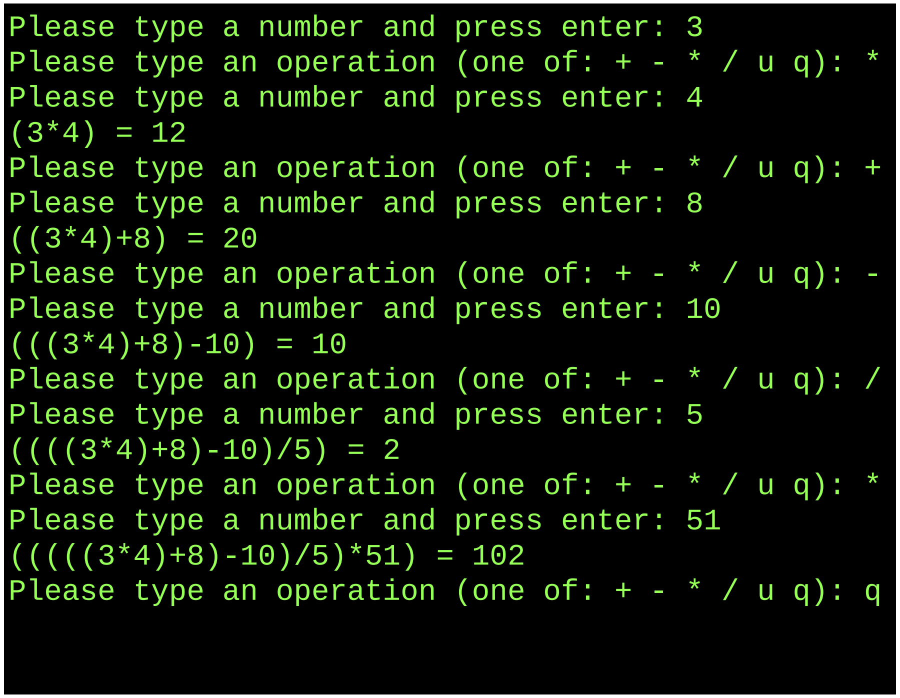
'*'	'+'	'-'	'/ '	'*'	‘q'
0	1	2	3	4	5

`result`

102



Calculator64



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): q
```

Variables

`allOperands`

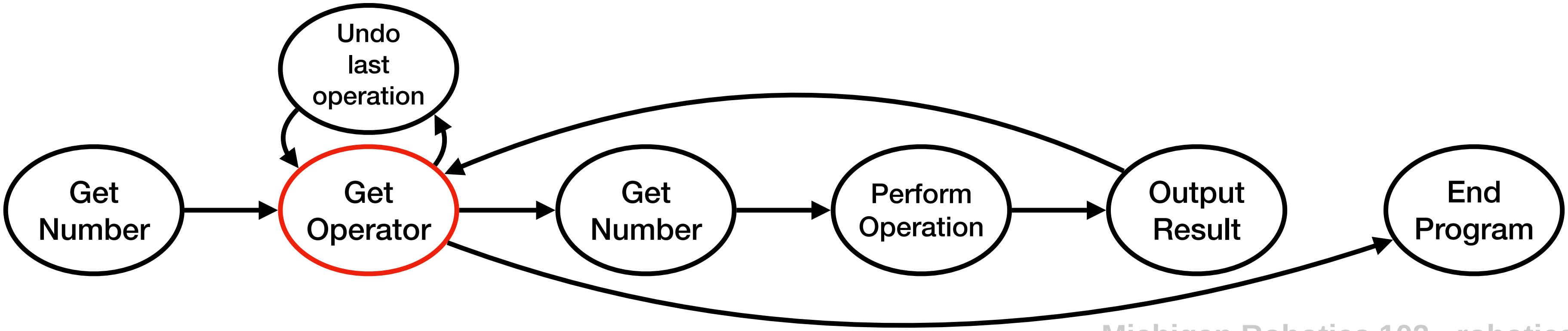
3	4	8	10	5	51
0	1	2	3	4	5

`allOperators`

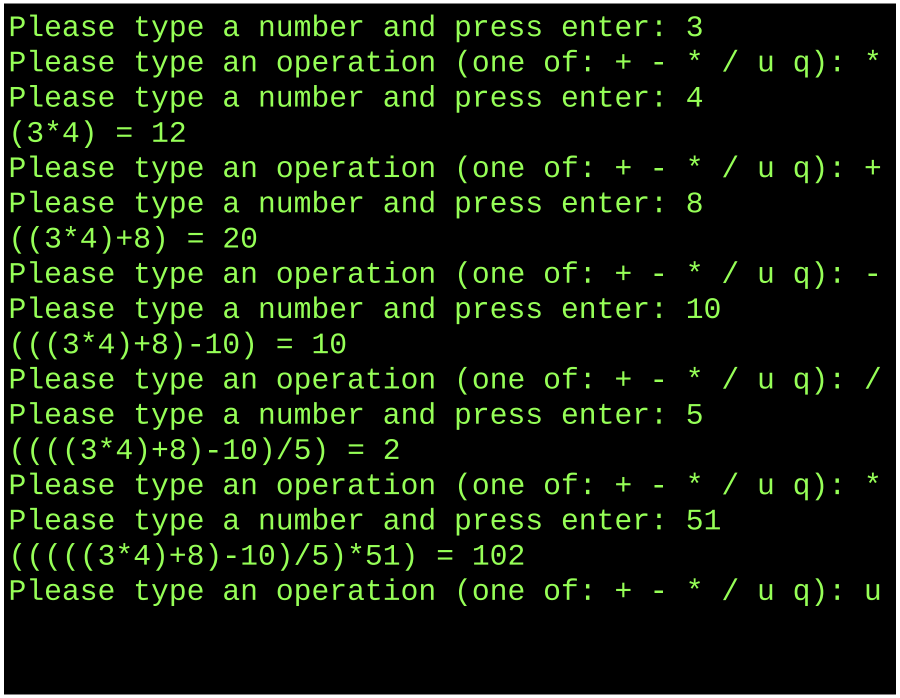
'*'	'+'	'-'	'/ '	'*'	‘q'
0	1	2	3	4	5

`result`

102



Calculator64



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
```

Variables

`allOperands`

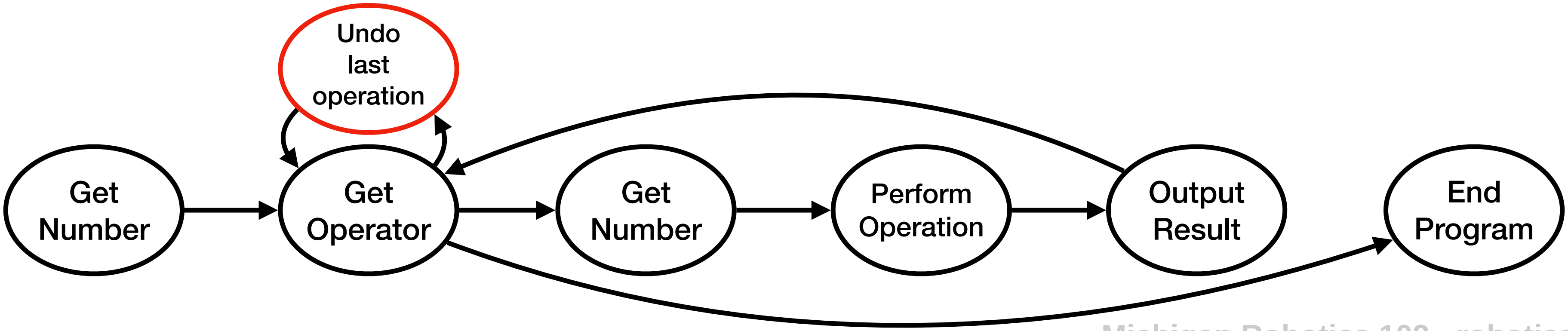
3	*4*	*8*	*10*	*5*	*51*
0	1	2	3	4	5

`allOperators`

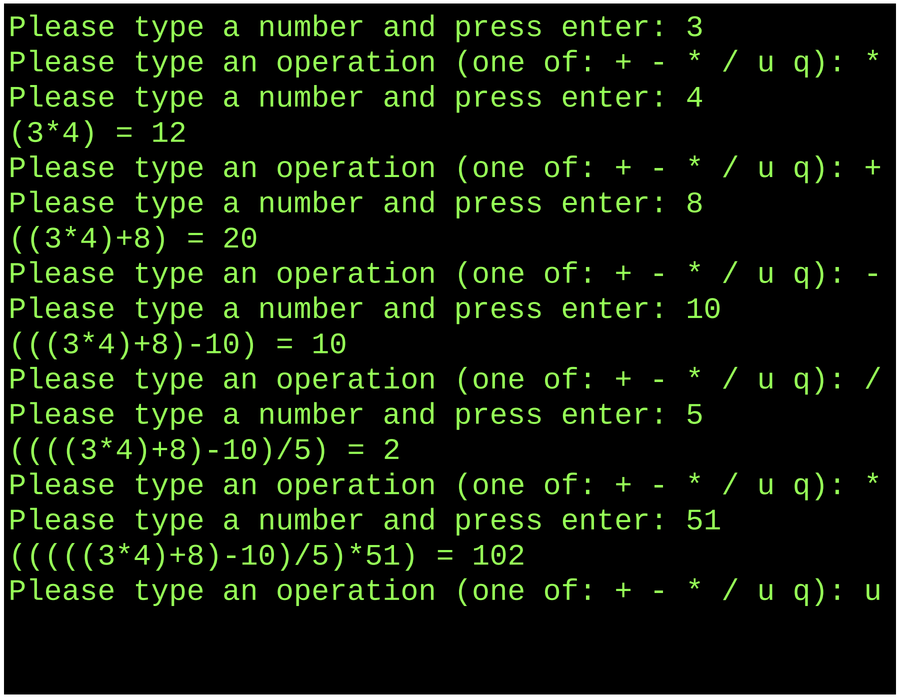
'*'	'+'	'-'	'/ '	'*'	‘u’
0	1	2	3	4	5

`result`

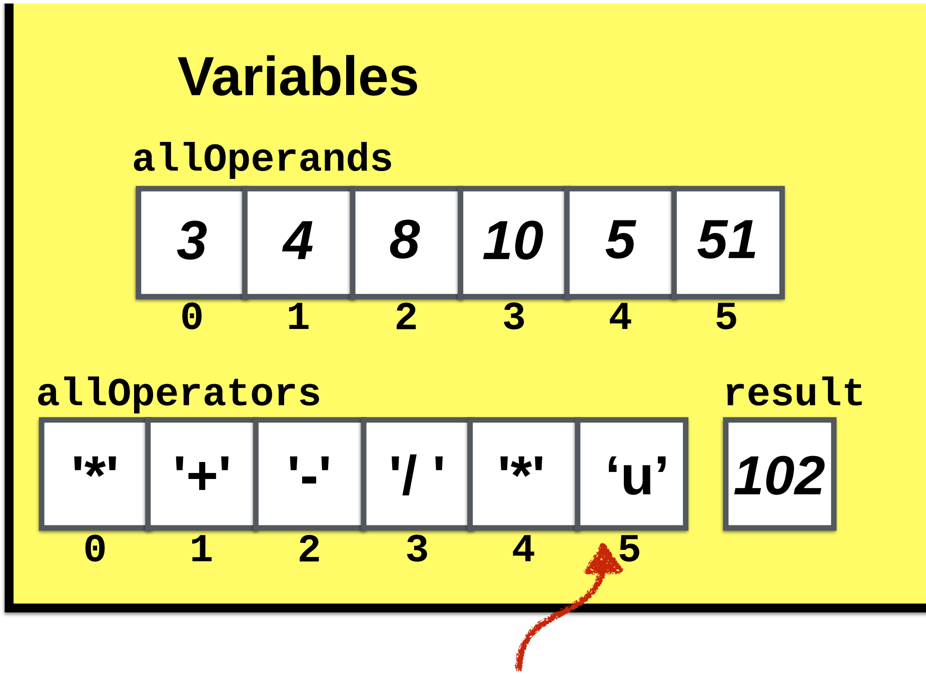
102



Calculator64



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
```



Variables

allOperands

3	4	8	10	5	51
0	1	2	3	4	5

allOperators

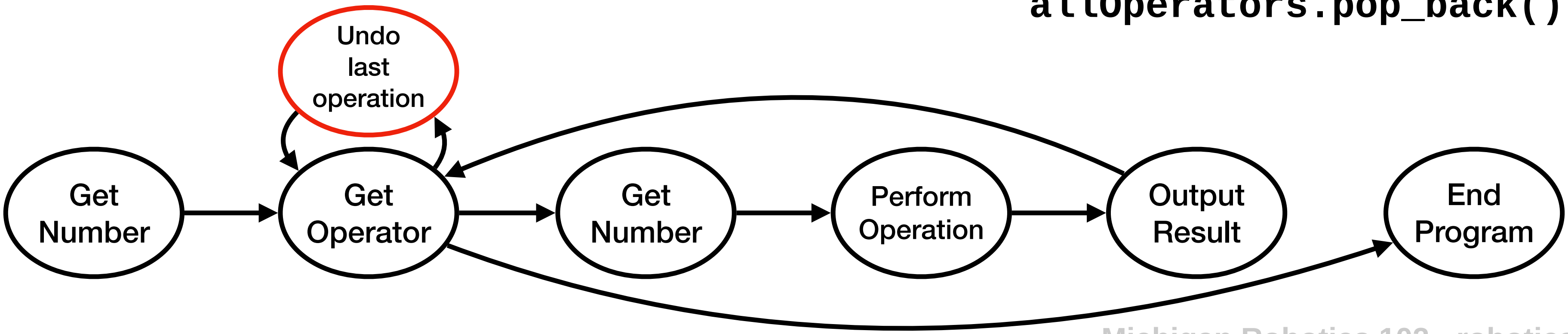
'*'	'+'	'-'	'/ '	'*'	‘u’
0	1	2	3	4	5

result

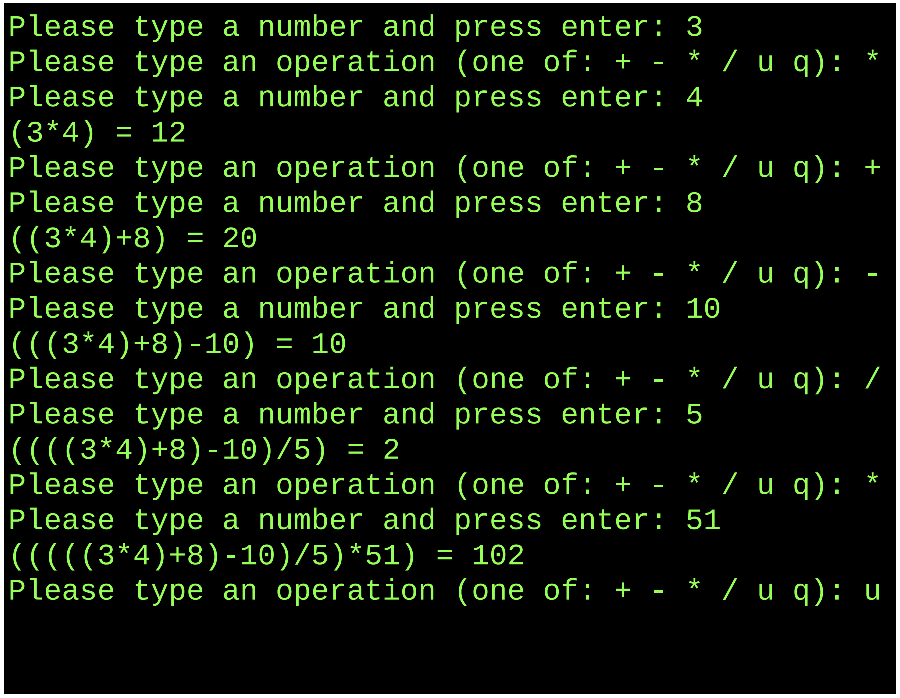
102

Remove last element of vector

`allOperators.pop_back();`



Calculator64



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
```

Variables

`allOperands`

3	4	8	10	5	51
0	1	2	3	4	5

`allOperators`

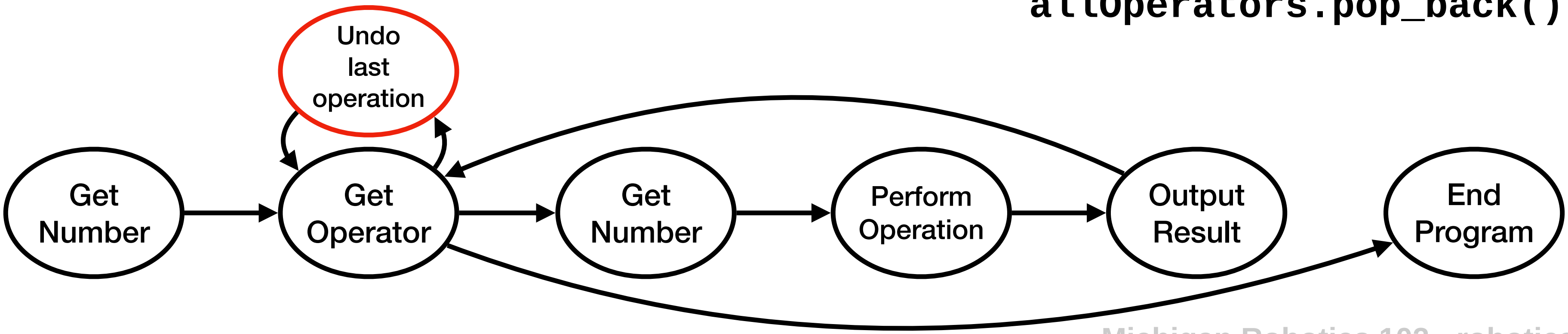
'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

`result`

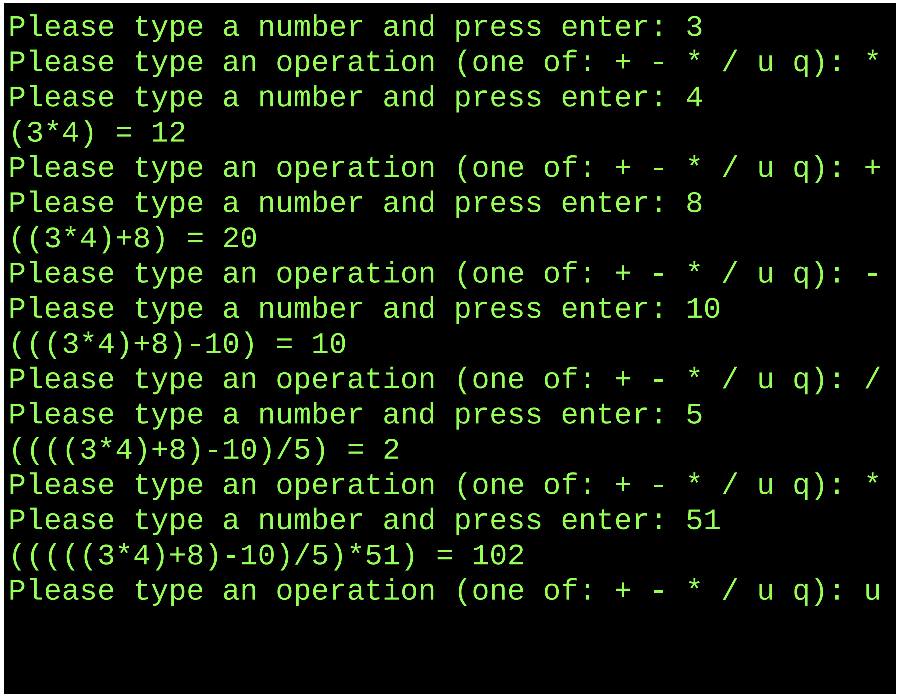
102

Remove last element of vector

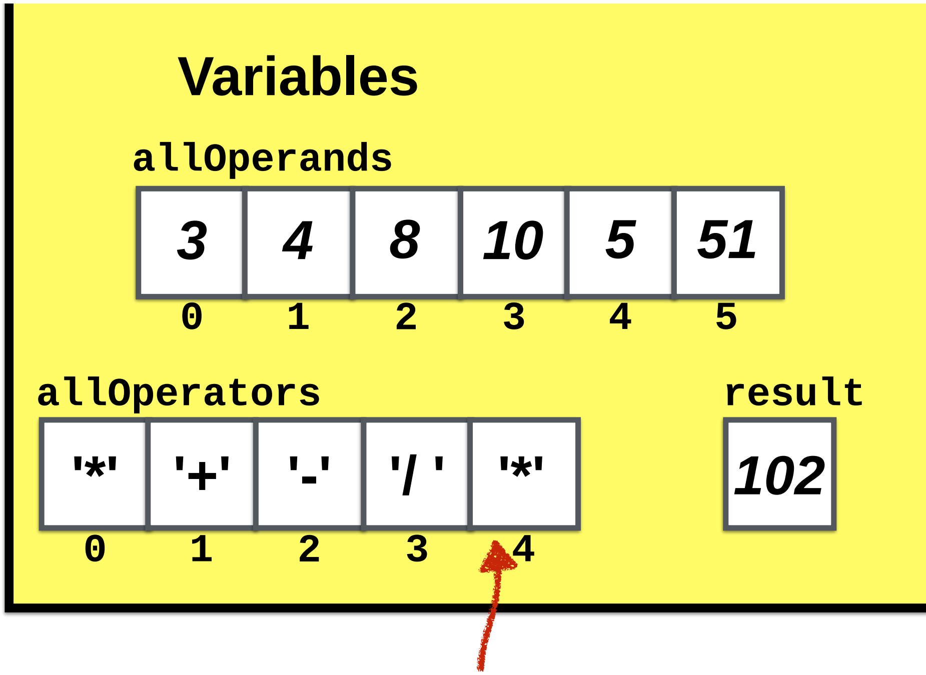
```
allOperators.pop_back();
```



Calculator64



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
```



Variables

`allOperands`

3	4	8	10	5	51
0	1	2	3	4	5

`allOperators`

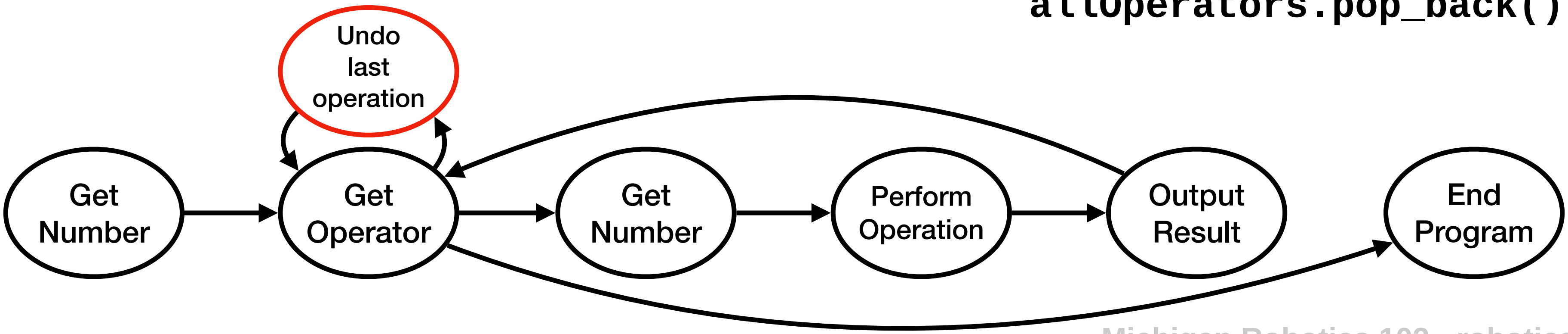
'*'	'+'	'-'	'/ '	'*'
0	1	2	3	4

`result`

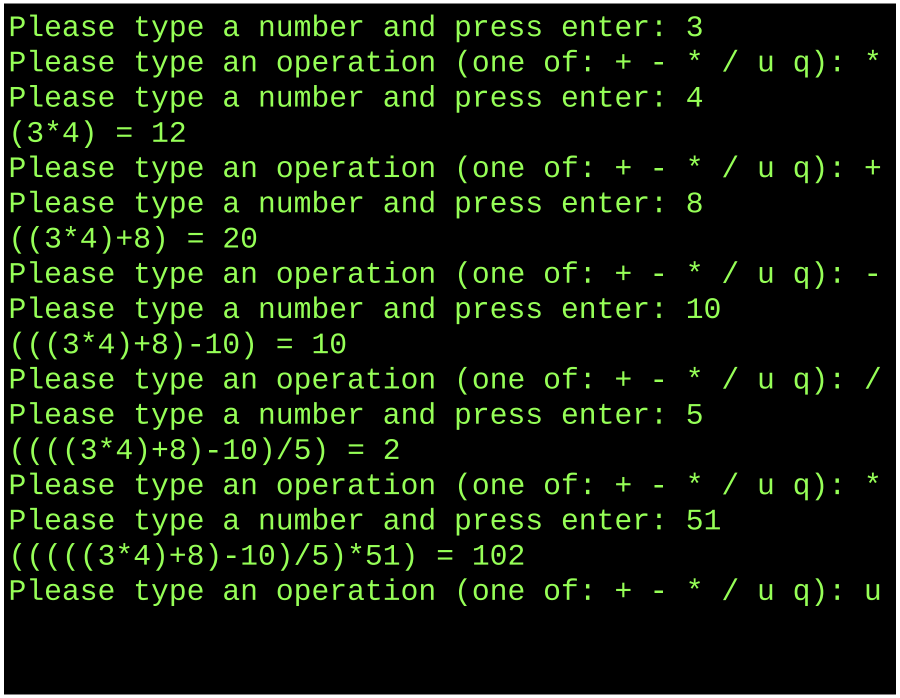
102

Remove last element of vector again

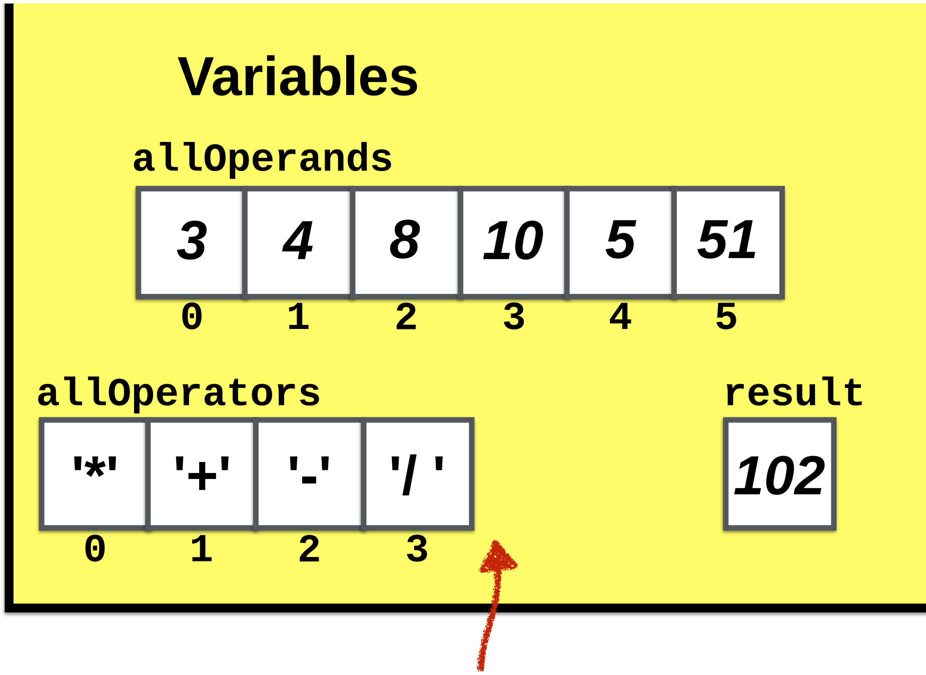
```
allOperators.pop_back();
```



Calculator64



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
```



Variables

allOperands

3	4	8	10	5	51
0	1	2	3	4	5

allOperators

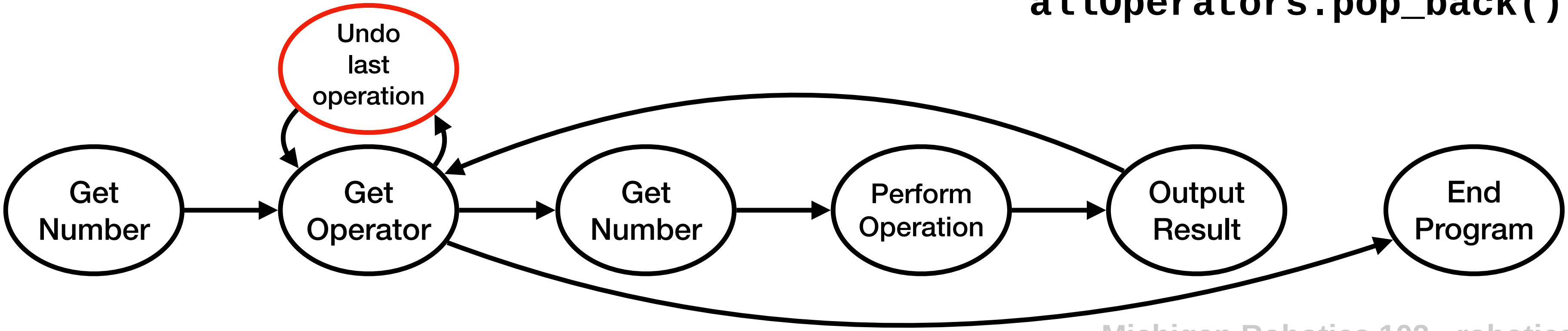
'*'	'+'	'-'	'/ '
0	1	2	3

result

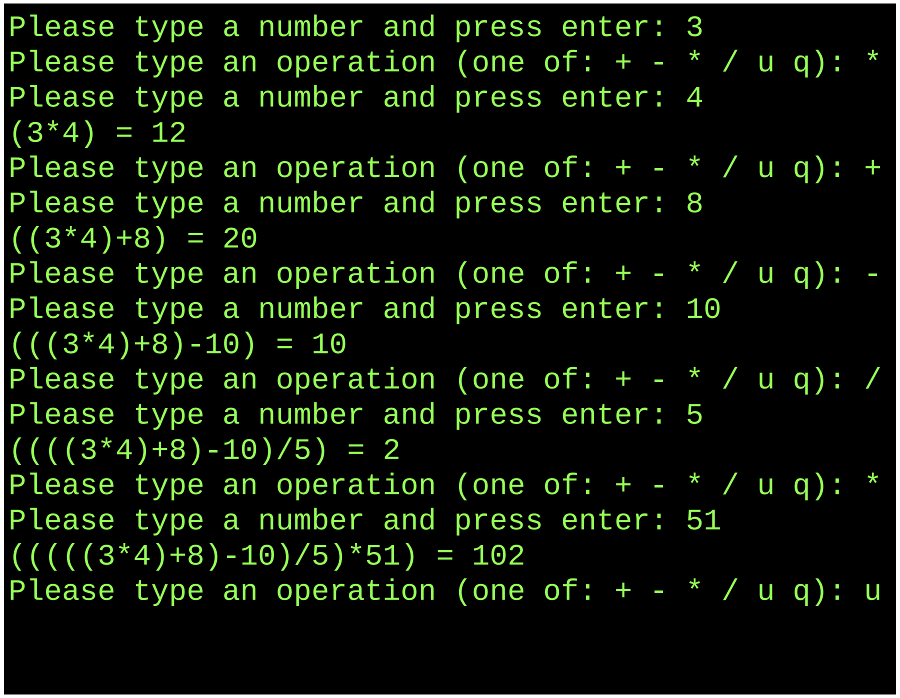
102

Remove last element of vector again

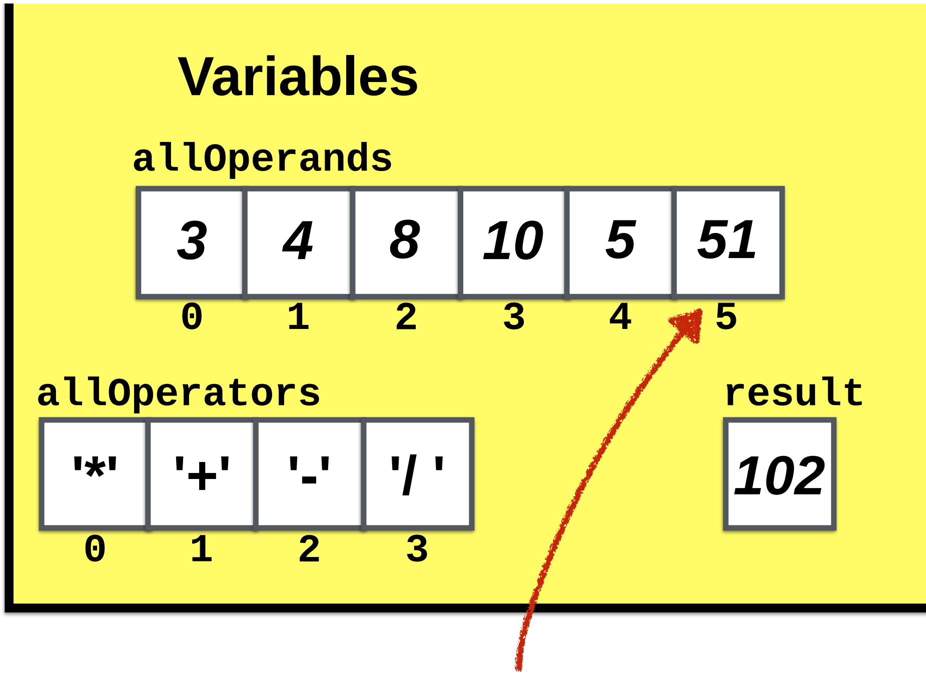
```
allOperators.pop_back();
```



Calculator64



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
```



Variables

allOperands

0	1	2	3	4	5
3	4	8	10	5	51

allOperators

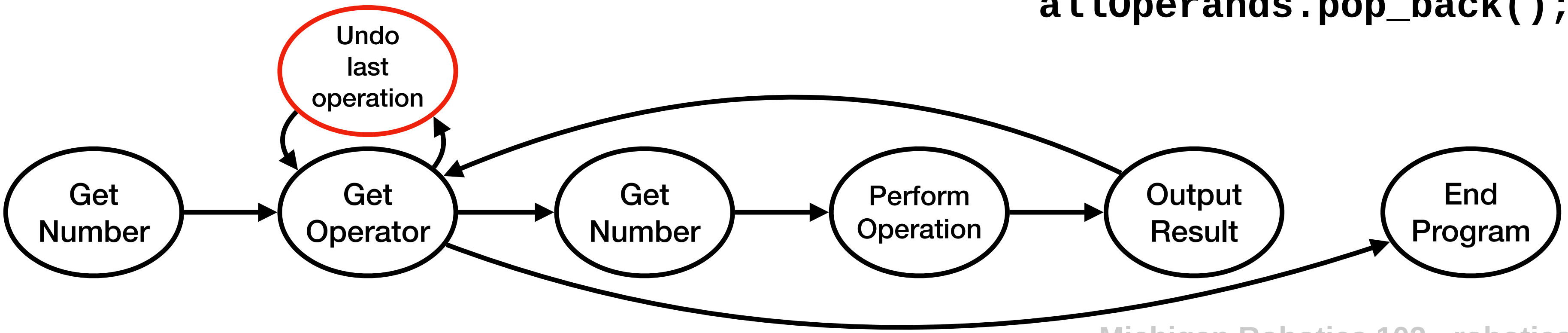
0	1	2	3
'*'	'+'	'-'	'/ '

result

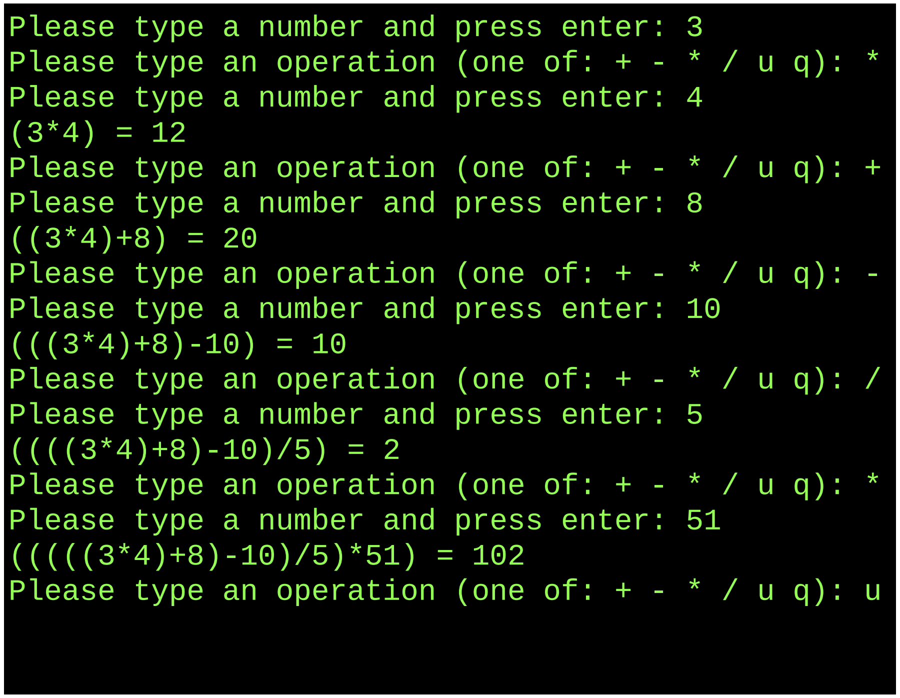
102

Remove last element of other vector

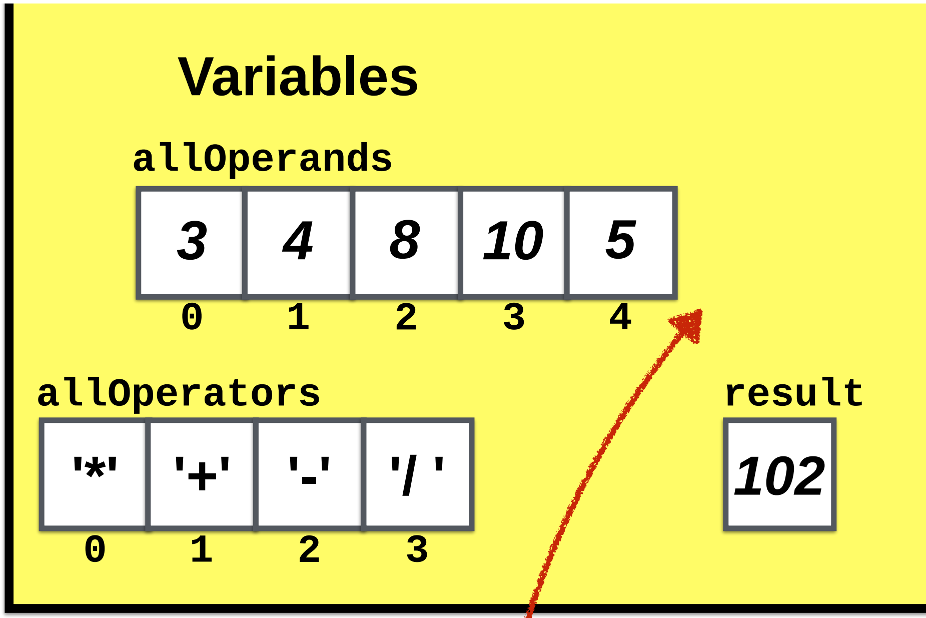
```
allOperands.pop_back();
```



Calculator64

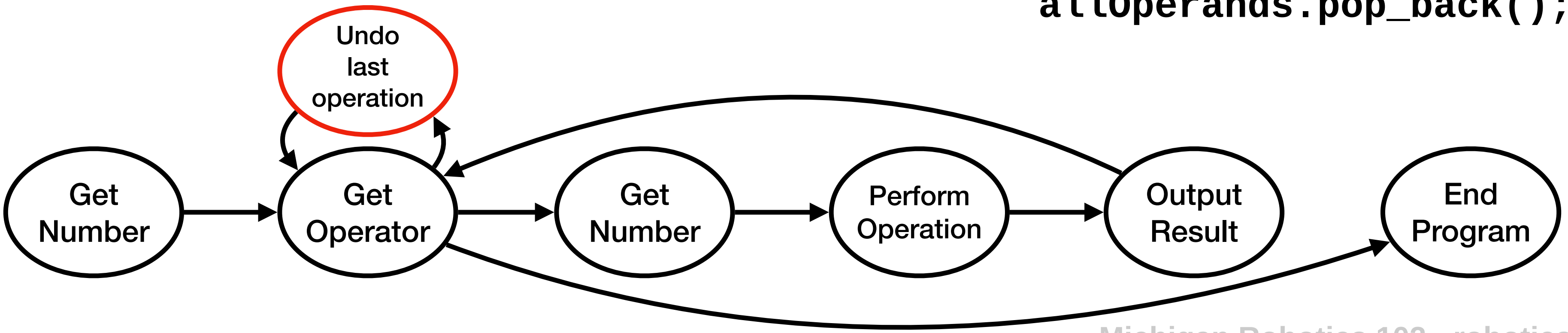


```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
```

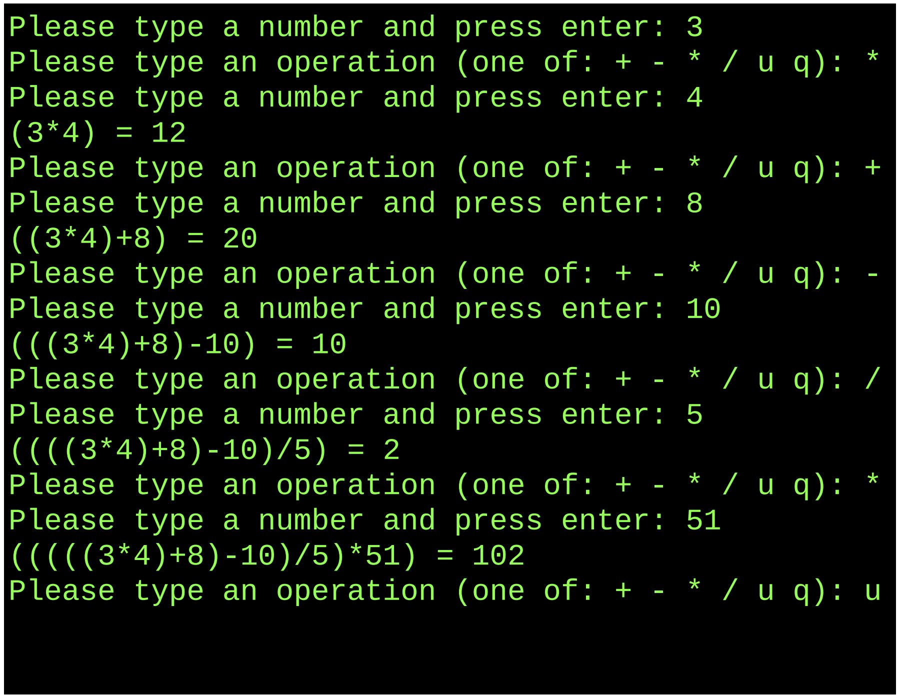


Remove last element of other vector

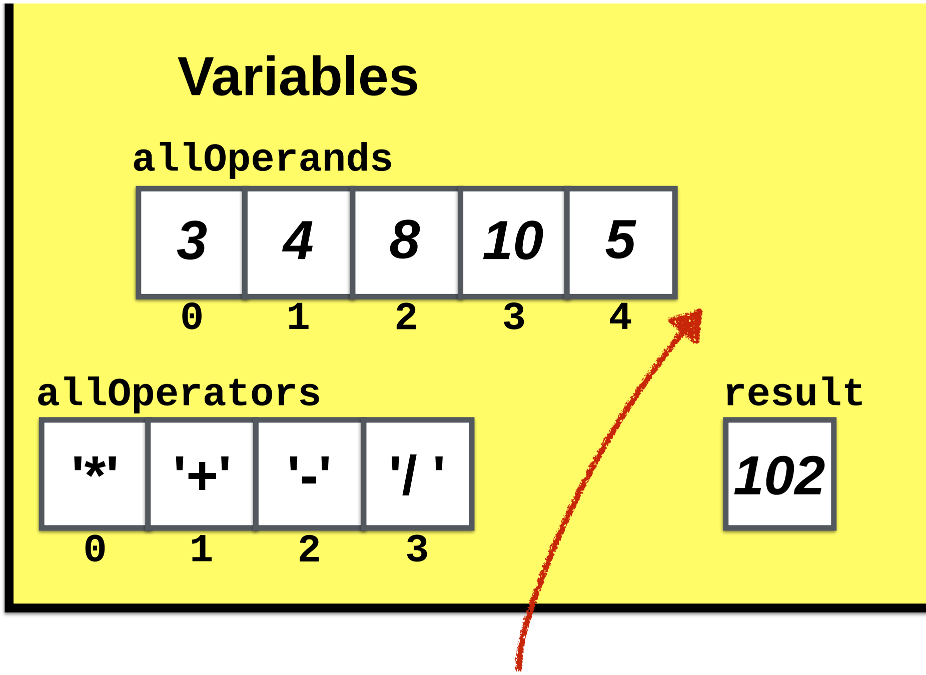
```
allOperands.pop_back();
```



Calculator64



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
```



Variables

allOperands

3	4	8	10	5
0	1	2	3	4

allOperators

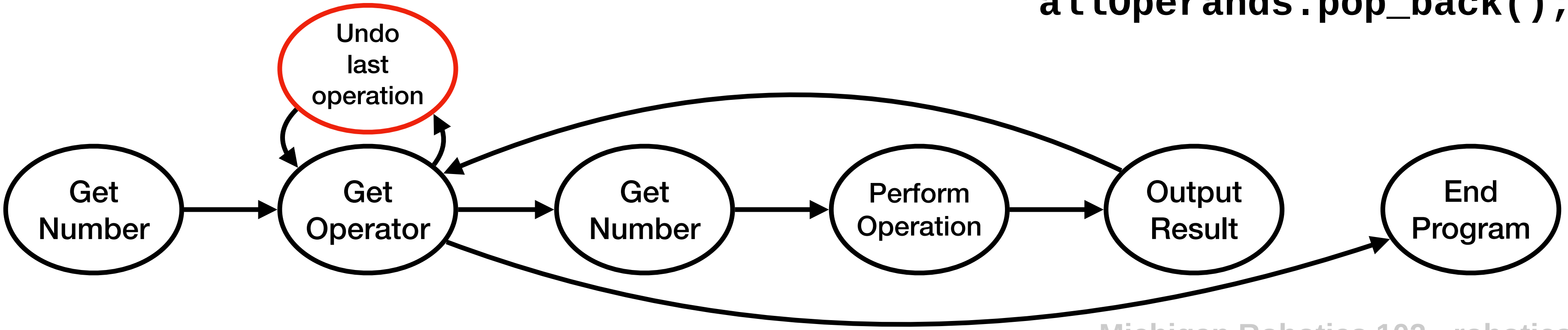
'*'	'+'	'-'	'/ '
0	1	2	3

result

102

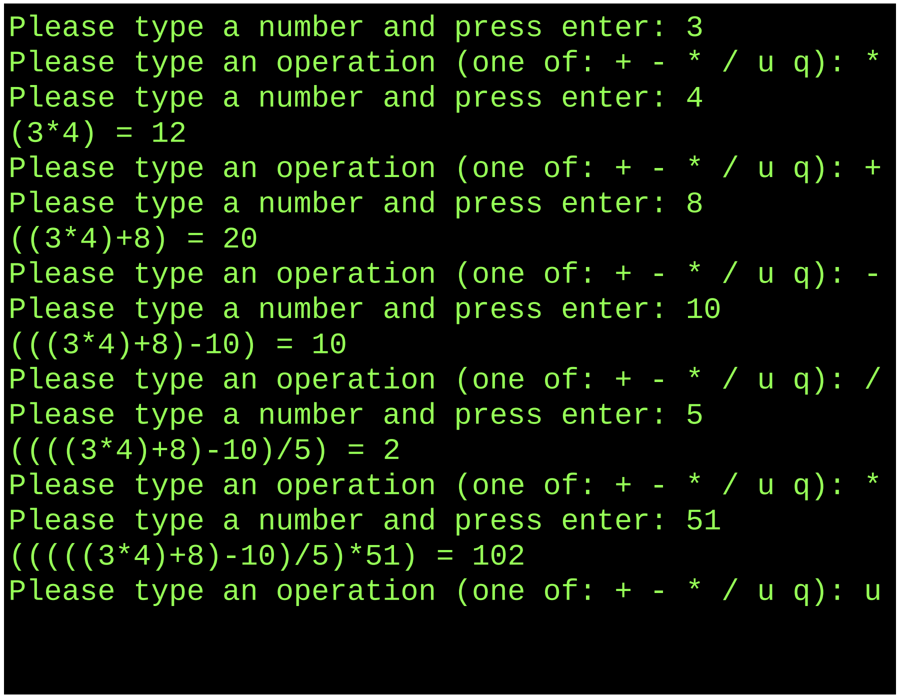
Remove last element of other vector

```
allOperands.pop_back();
```

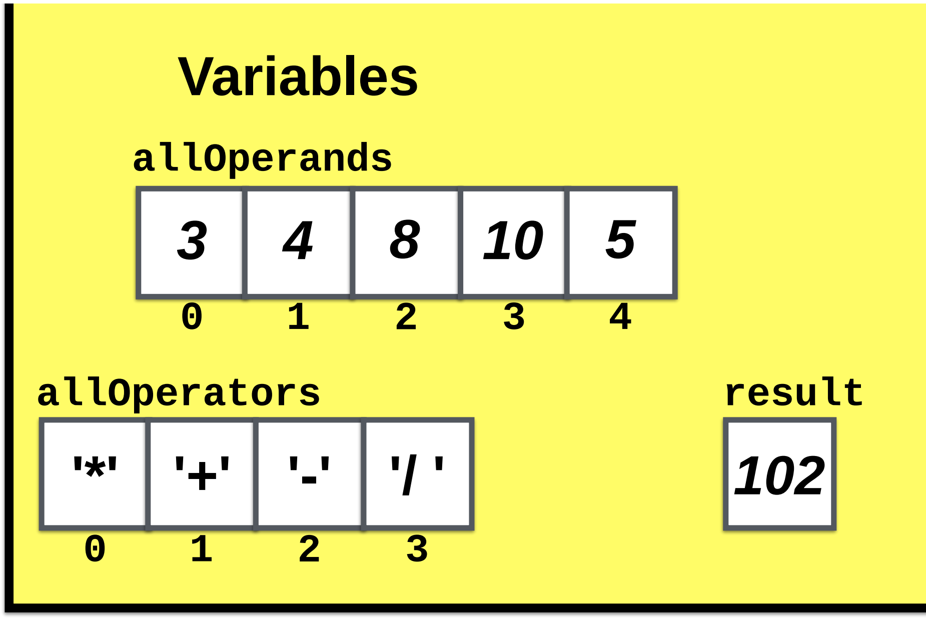


Michigan Robotics 102 - robotics102.org

Calculator64



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
```



Variables

allOperands

3	4	8	10	5
0	1	2	3	4

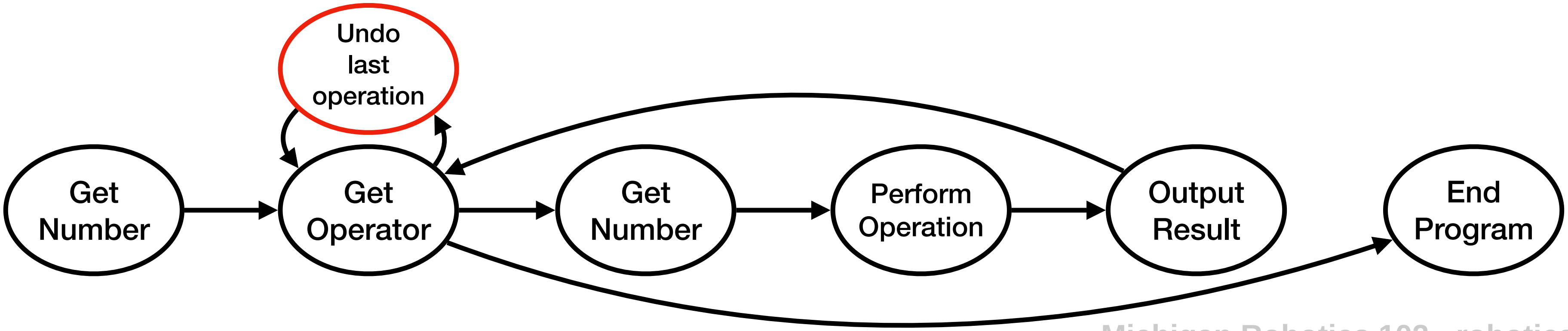
allOperators

'*'	'+'	'-'	'/ '
0	1	2	3

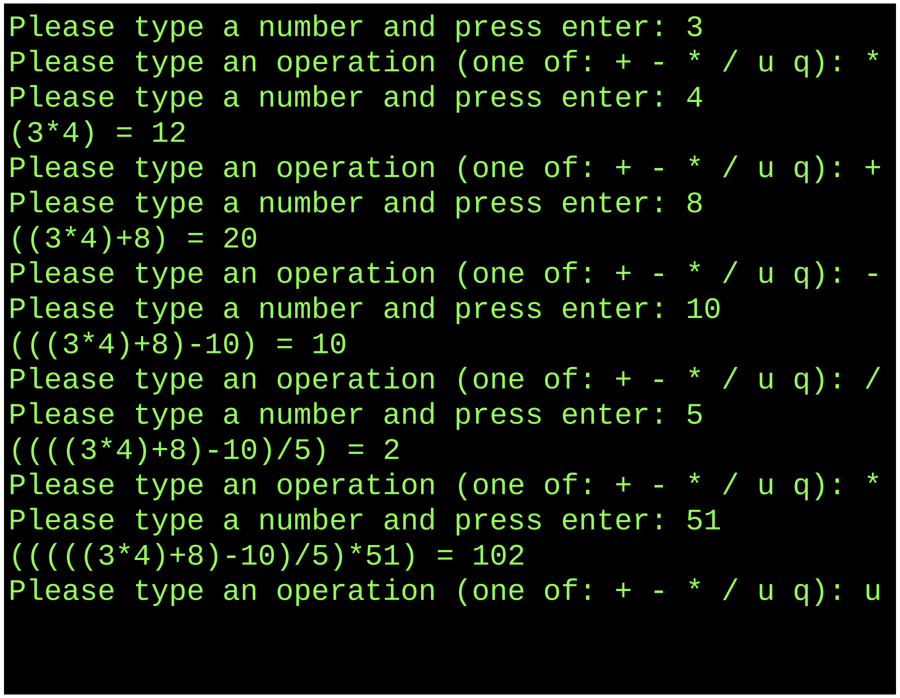
result

102

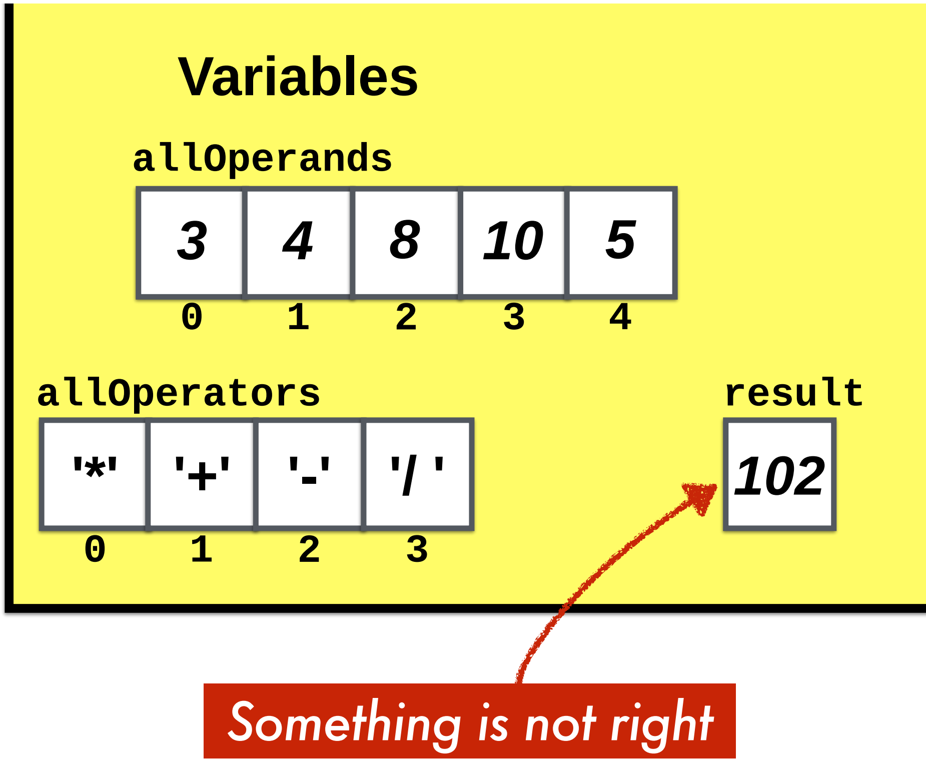
Something is not right



Calculator64



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
```



Variables

allOperands

0	1	2	3	4
3	4	8	10	5

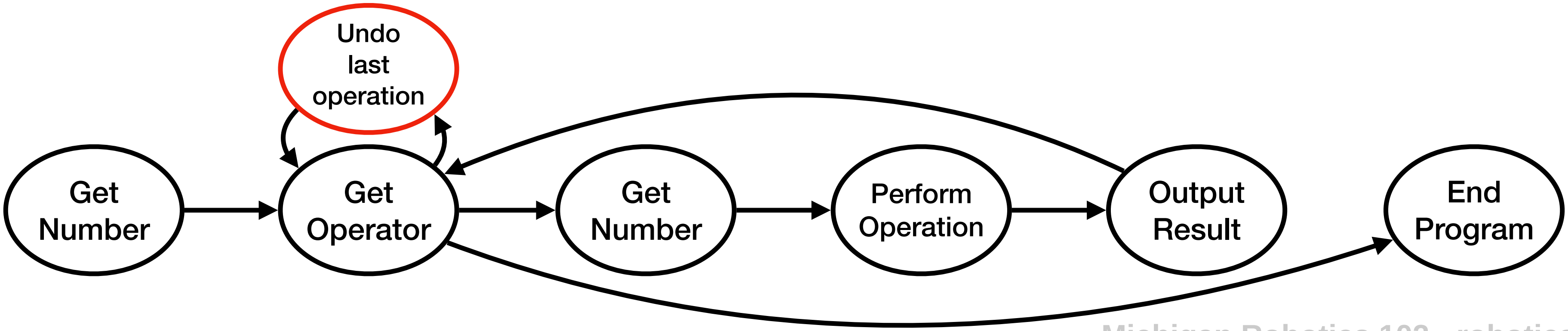
allOperators

0	1	2	3
'*'	'+'	'-'	'/ '

result

102

Something is not right



Undo last operation

Get Number

Get Operator

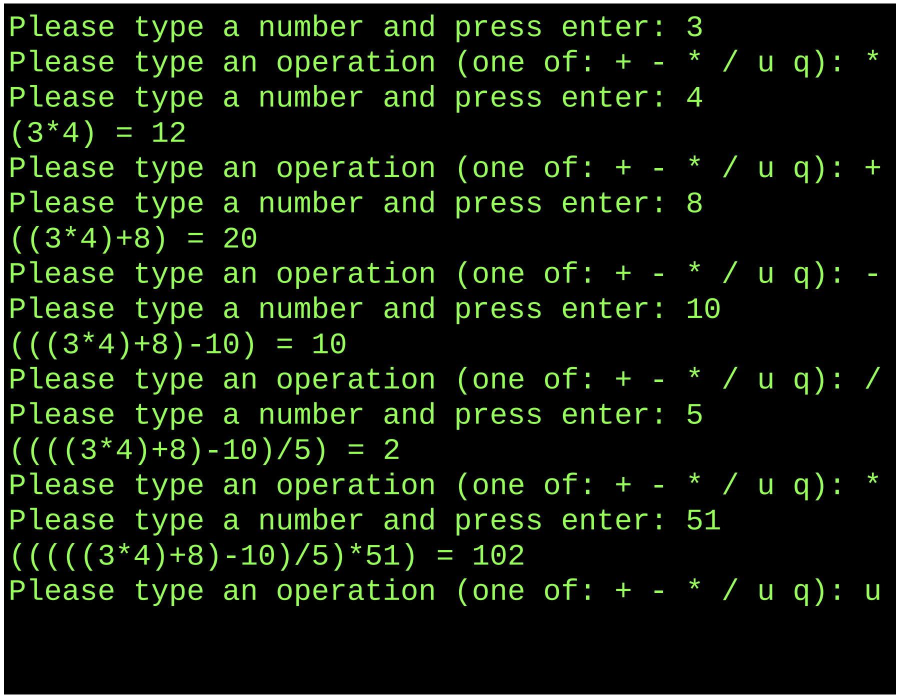
Get Number

Perform Operation

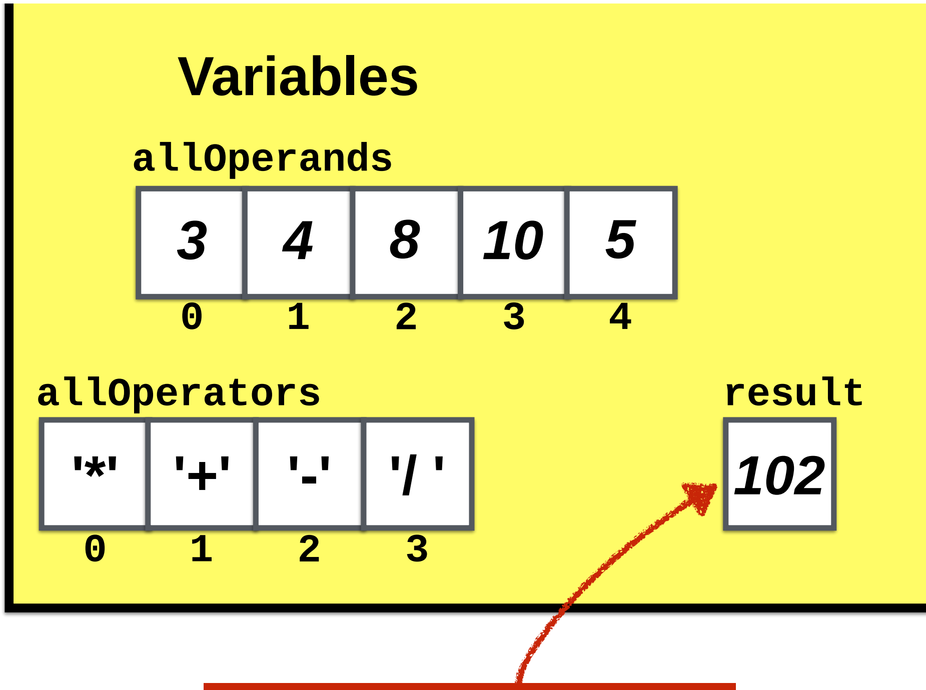
Output Result

End Program

Calculator64



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
```



Variables

allOperands

0	1	2	3	4
3	4	8	10	5

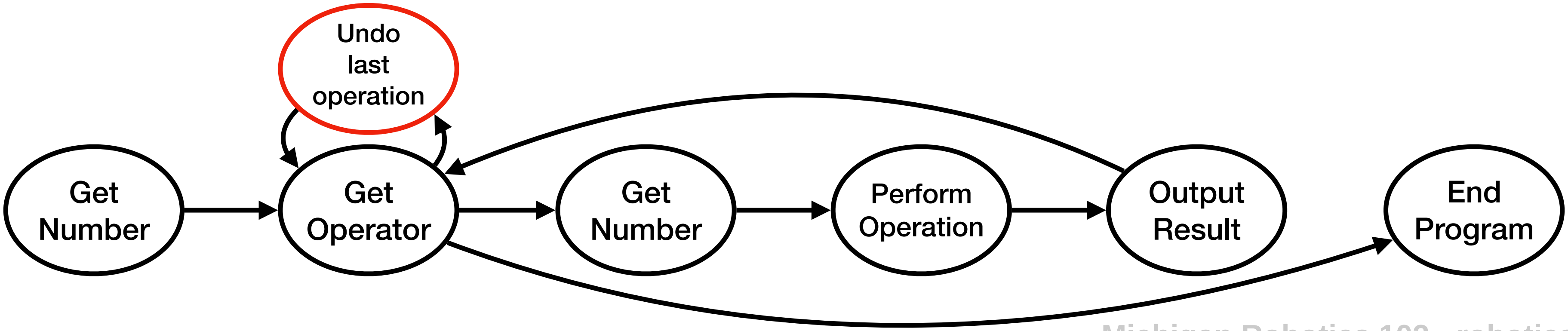
allOperators

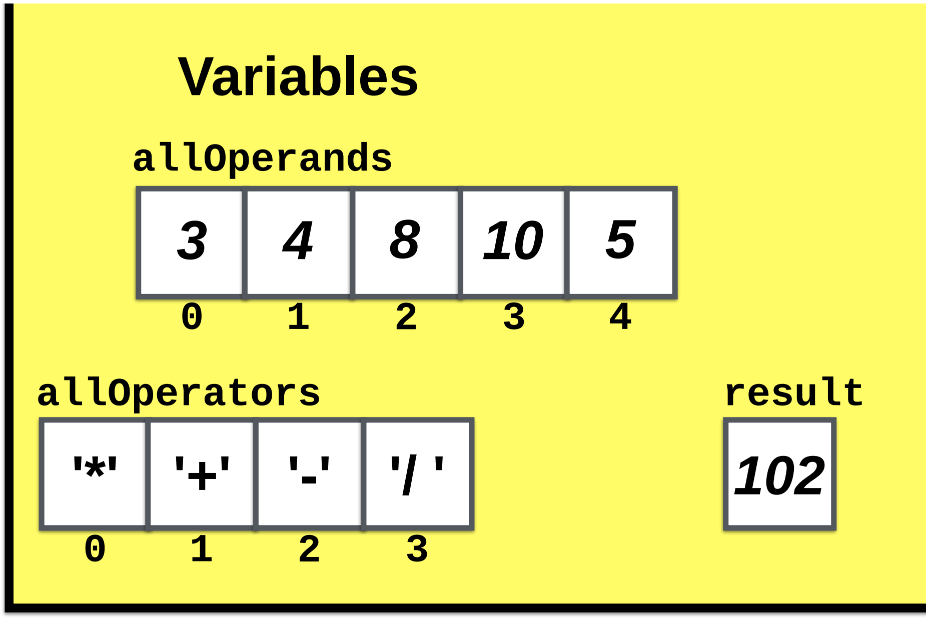
0	1	2	3
'*'	'+'	'-'	'/ '

result

102

Something is not right





Variables

allOperands

0	1	2	3	4
3	4	8	10	5

allOperators

0	1	2	3
'*'	'+'	'-'	'/ '

result

102

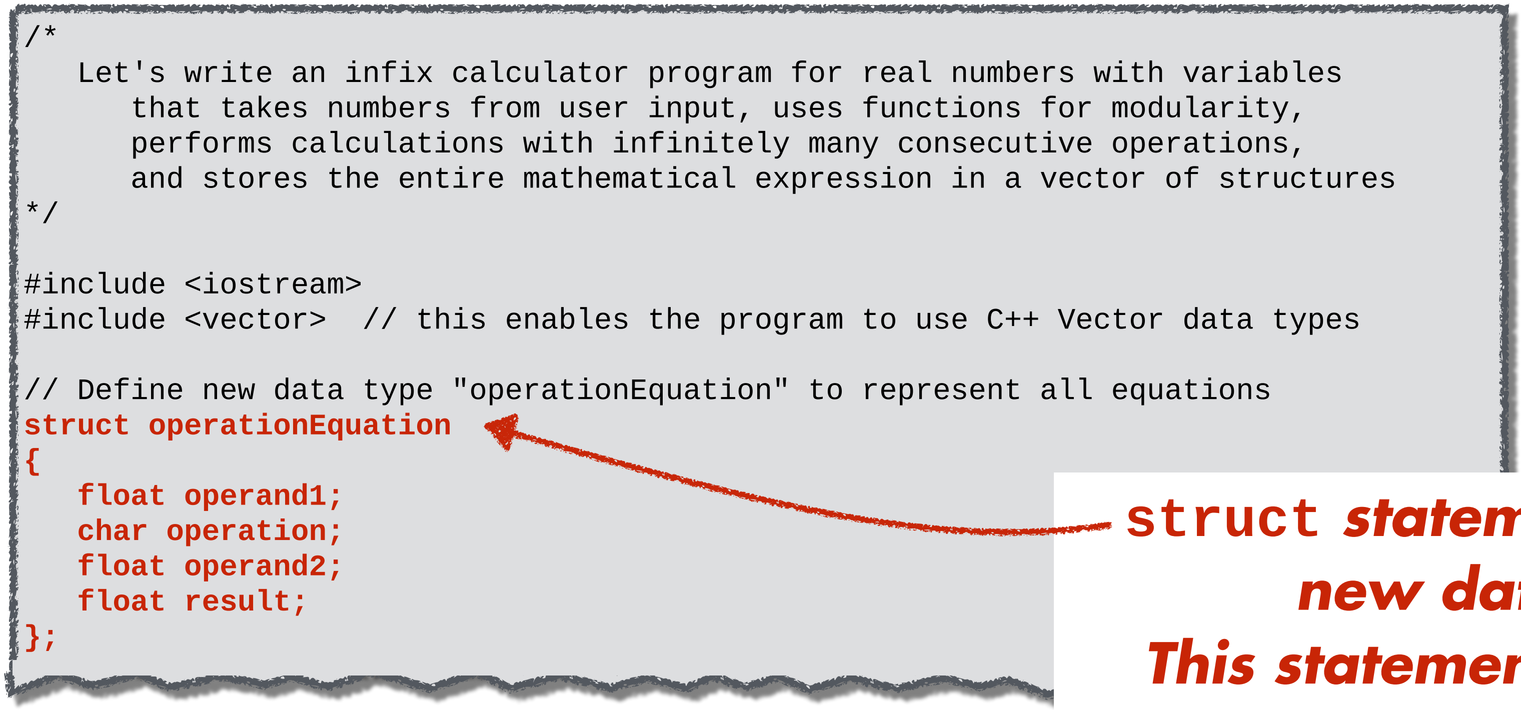
Store all variables in a common data structure

Store all variables in a common data structure

Vector of Structs

How to represent an operation with a struct ?

calculator.cpp (Version 64)



```
/*
   Let's write an infix calculator program for real numbers with variables
      that takes numbers from user input, uses functions for modularity,
      performs calculations with infinitely many consecutive operations,
      and stores the entire mathematical expression in a vector of structures
*/

#include <iostream>
#include <vector>  // this enables the program to use C++ Vector data types

// Define new data type "operationEquation" to represent all equations
struct operationEquation
{
   float operand1;
   char operation;
   float operand2;
   float result;
};
```

`struct` ***statement defines a new data type.***

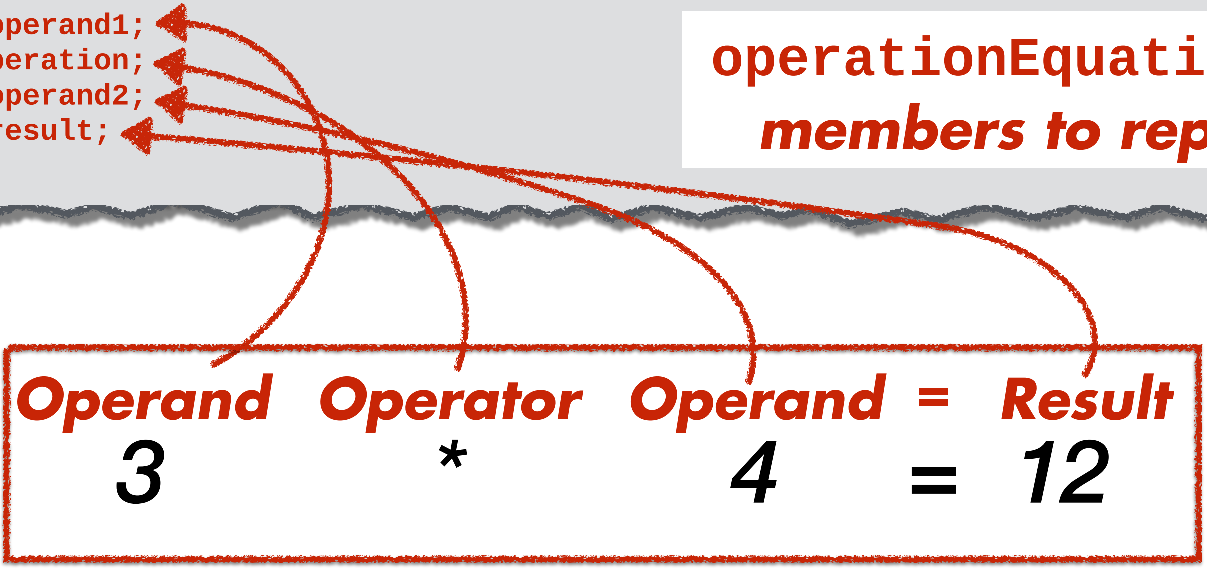
This statement creates the `operationEquation` ***data type***

How to represent an operation with a struct ?

calculator.cpp (Version 64)

```
/*
   Let's write an infix calculator program for real numbers with variables
      that takes numbers from user input, uses functions for modularity,
      performs calculations with infinitely many consecutive operations,
      and stores the entire mathematical expression in a vector of structures
*/

#include <iostream>
#include <vector>  // this enables the program to use C++ Vector data types

// Define new data type "operationEquation" to represent all equations
struct operationEquation
{
   float operand1;
   char operation;
   float operand2;
   float result;
};
```

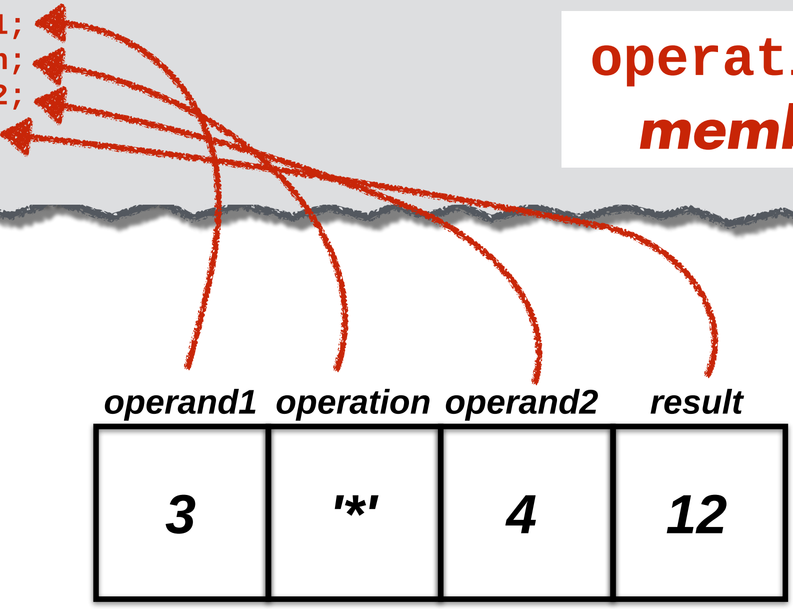
`operationEquation` ***data type has four members to represent an operation***



calculator.cpp (Version 64)

```
/*
   Let's write an infix calculator program for real numbers with variables
      that takes numbers from user input, uses functions for modularity,
      performs calculations with infinitely many consecutive operations,
      and stores the entire mathematical expression in a vector of structures
*/

#include <iostream>
#include <vector>  // this enables the program to use C++ Vector data types

// Define new data type "operationEquation" to represent all equations
struct operationEquation
{
   float operand1;
   char operation;
   float operand2;
   float result;
};
```

`operationEquation` data type has four members to represent an operation



operand1	operation	operand2	result
3	'*'	4	12

calculator.cpp (Version 64)

```
/*
   Let's write an infix calculator program for real numbers with variables
      that takes numbers from user input, uses functions for modularity,
      performs calculations with infinitely many consecutive operations,
      and stores the entire mathematical expression in a vector of structures
*/

#include <iostream>
#include <vector>  // this enables the program to use C++ Vector data types

// Define new data type "operationEquation" to represent all equations
struct operationEquation
{
   float operand1;
   char operation;
   float operand2;
   float result;
};
```

```
// Let's show an example of a struct variable
operationEquation myOperation;  // struct storing operands, operator, result

myOperation.operand1 = 3;
myOperation.operation = ‘*’;
myOperation.operand2 = 4;
myOperation.result = myOperation.operand1 * myOperation.operand2;
```

calculator.cpp (Version 64)

```
/*
   Let's write an infix calculator program for real numbers with variables
      that takes numbers from user input, uses functions for modularity,
      performs calculations with infinitely many consecutive operations,
      and stores the entire mathematical expression in a vector of structures
*/

#include <iostream>
#include <vector>  // this enables the program to use C++ Vector data types

// Define new data type "operationEquation" to represent all equations
struct operationEquation
{
   float operand1;
   char operation;
   float operand2;
   float result;
};
```

A variable can be declared with the data type we defined using `struct`

```
// Let's show an example of a struct variable
operationEquation myOperation;  // struct storing operands, operator, result

myOperation.operand1 = 3;
myOperation.operation = ‘*’;
myOperation.operand2 = 4;
myOperation.result = myOperation.operand1 * myOperation.operand2;
```

calculator.cpp (Version 64)

```
/*
   Let's write an infix calculator program for real numbers with variables
      that takes numbers from user input, uses functions for modularity,
      performs calculations with infinitely many consecutive operations,
      and stores the entire mathematical expression in a vector of structures
*/

#include <iostream>
#include <vector>  // this enables the program to use C++ Vector data types

// Define new data type "operationEquation" to represent all equations
struct operationEquation
{
   float operand1;
   char operation;
   float operand2;
   float result;
};
```

Properties of a structure are accessed using ‘.’ modifier in the form `struct.member`

```
// Let's show an example of a struct variable
operationEquation myOperation;  // struct storing operands, operator, result

myOperation.operand1 = 3;
myOperation.operation = ‘*’;
myOperation.operand2 = 4;
myOperation.result = myOperation.operand1 * myOperation.operand2;
```

calculator.cpp (Version 64)

```
/*
   Let's write an infix calculator program for real numbers with variables
      that takes numbers from user input, uses functions for modularity,
      performs calculations with infinitely many consecutive operations,
      and stores the entire mathematical expression in a vector of structures
*/

#include <iostream>
#include <vector>  // this enables the program to use C++ Vector data types

// Define new data type "operationEquation" to represent all equations
struct operationEquation
{
   float operand1;
   char operation;
   float operand2;
   float result;
};
```

A member of a struct can be passed as an argument to a function

```
main()
```

```
   // Ask the user for the first operand
   getNumber(myOperation.operand1);  // access element y of struct x as x.y
```

```
// Function to prompt the user to input a number that is returned in a variable
bool getNumber(float &number) {
```

calculator.cpp (Version 64)

```
/*
   Let's write an infix calculator program for real numbers with variables
      that takes numbers from user input, uses functions for modularity,
      performs calculations with infinitely many consecutive operations,
      and stores the entire mathematical expression in a vector of structures
*/

#include <iostream>
#include <vector>  // this enables the program to use C++ Vector data types

// Define new data type "operationEquation" to represent all equations
struct operationEquation
{
   float operand1;
   char operation;
   float operand2;
   float result;
};
```

A struct variable can be passed as an argument to a function

```
main()
```

```
      // Output operation result to screen
      outputResult(myOperation);
```

```
// Function defined to output operation results to screen
bool outputResult(struct operationEquation op) {  // only pass by value needed
```

calculator.cpp (Version 66)

```
/*
   Let's write an infix calculator program for real numbers with variables
      that takes numbers from user input, uses functions for modularity,
      performs calculations with infinitely many consecutive operations,
      and stores the entire mathematical expression in a vector of structures
*/

#include <iostream>
#include <vector>  // this enables the program to use C++ Vector data types

// Define new data type "operationEquation" to represent all equations
struct operationEquation
{
   float operand1;
   char operation;
   float operand2;
   float result;
};
```

```
main()
```

```
   // Vector of all operations using the operationEquation struct we created
   std::vector <operationEquation> allOperations;
```

A vector of structs

calculator.cpp (Version 66)

```
/*
   Let's write an infix calculator program for real numbers with variables
      that takes numbers from user input, uses functions for modularity,
      performs calculations with infinitely many consecutive operations,
      and stores the entire mathematical expression in a vector of structures
*/

#include <iostream>
#include <vector>  // this enables the program to use C++ Vector data types

// Define new data type "operationEquation" to represent all equations
struct operationEquation
{
   float operand1;
   char operation;
   float operand2;
   float result;
};
```

main()

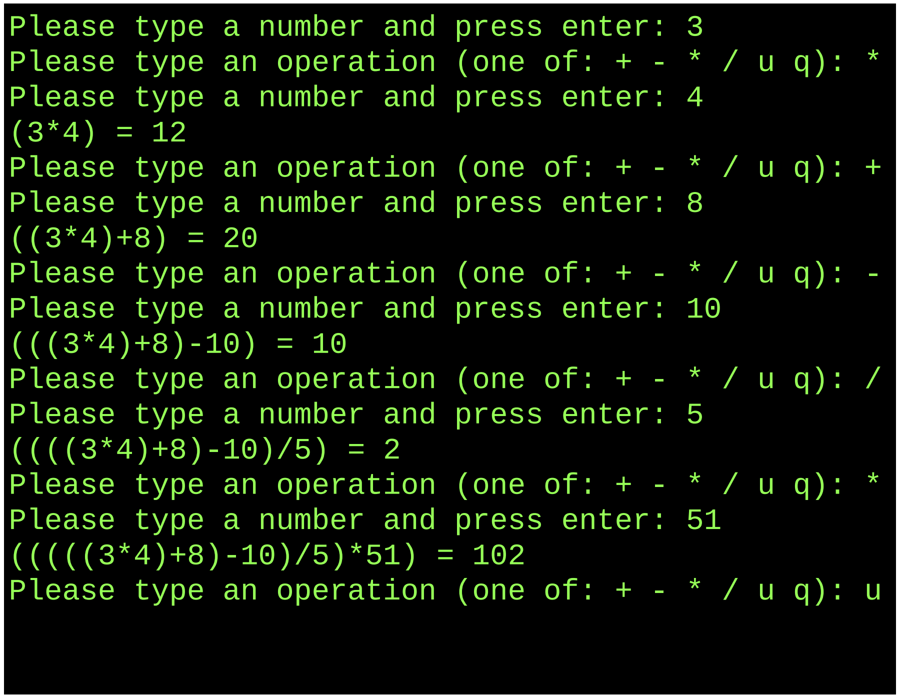
```
   // Vector of all operations using the operationEquation struct we created
   std::vector <operationEquation> allOperations;
```

```
// Output the entire current equation to the screen
bool outputCurrentEquation(std::vector<operationEquation> &ops) {
```

A vector of structs

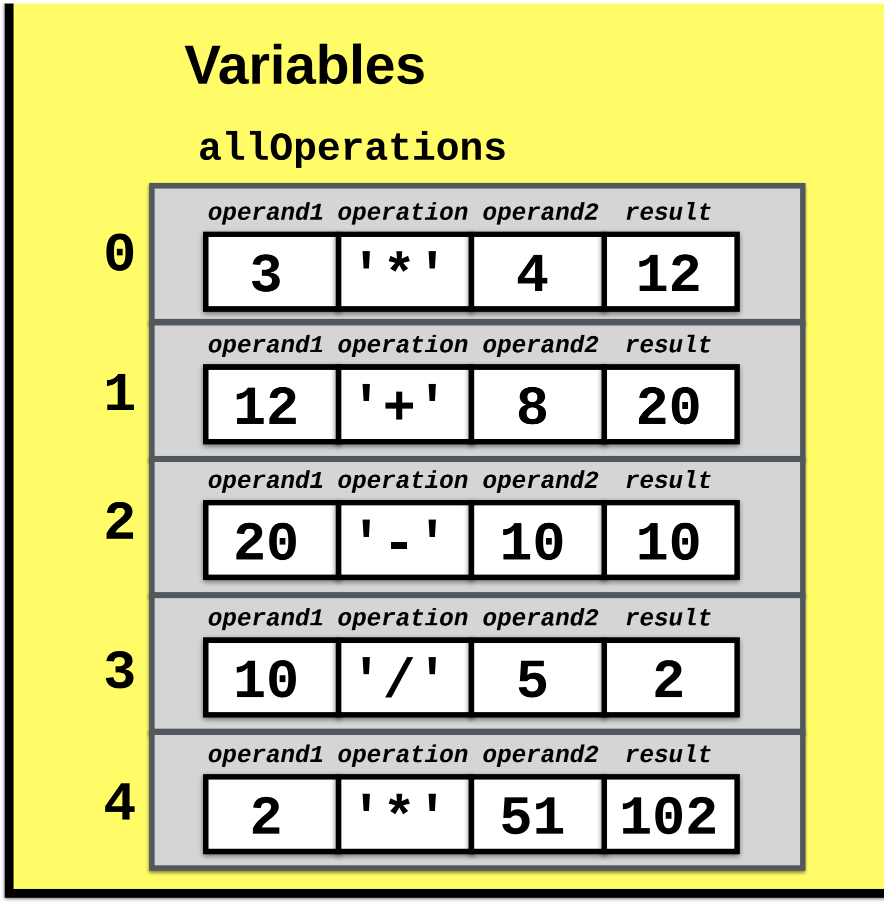
Can be passed as a function argument

Calculator66



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
```

Variables



allOperations

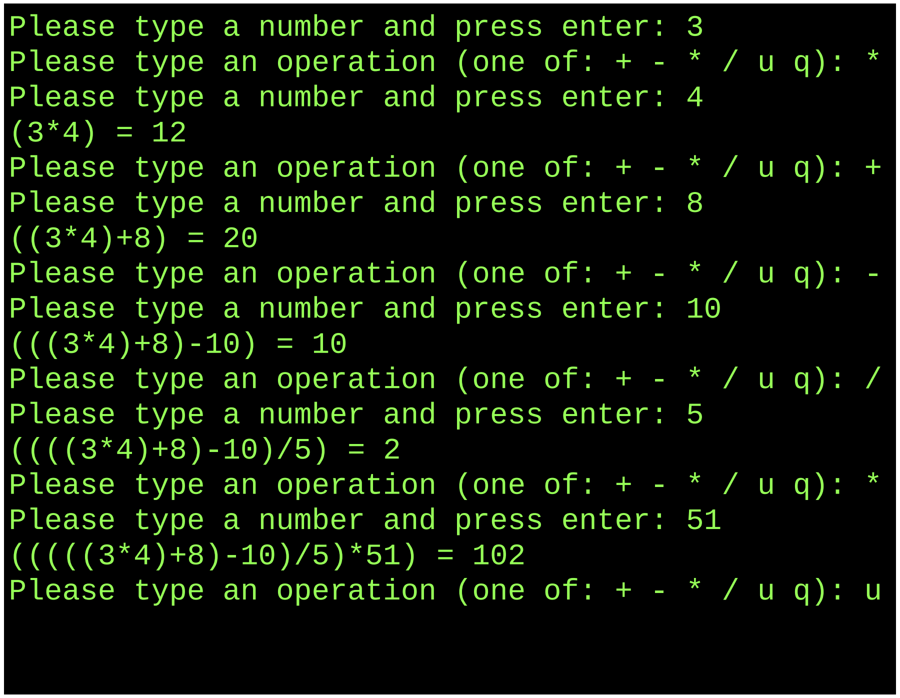
	operand1	operation	operand2	result
0	3	'*'	4	12
1	12	'+'	8	20
2	20	'-'	10	10
3	10	'/'	5	2
4	2	'*'	51	102

A vector of structs

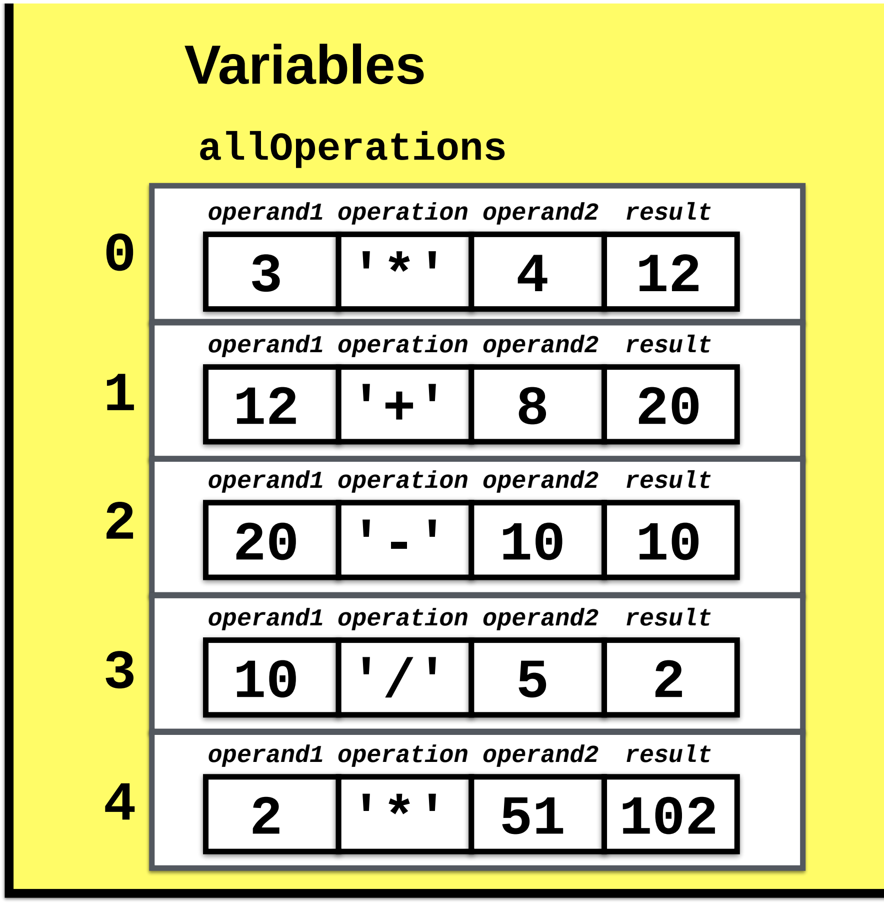
main()

```
  // Vector of all operations using the operationEquation struct we created
  std::vector <operationEquation> allOperations;
```

Calculator66



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
```



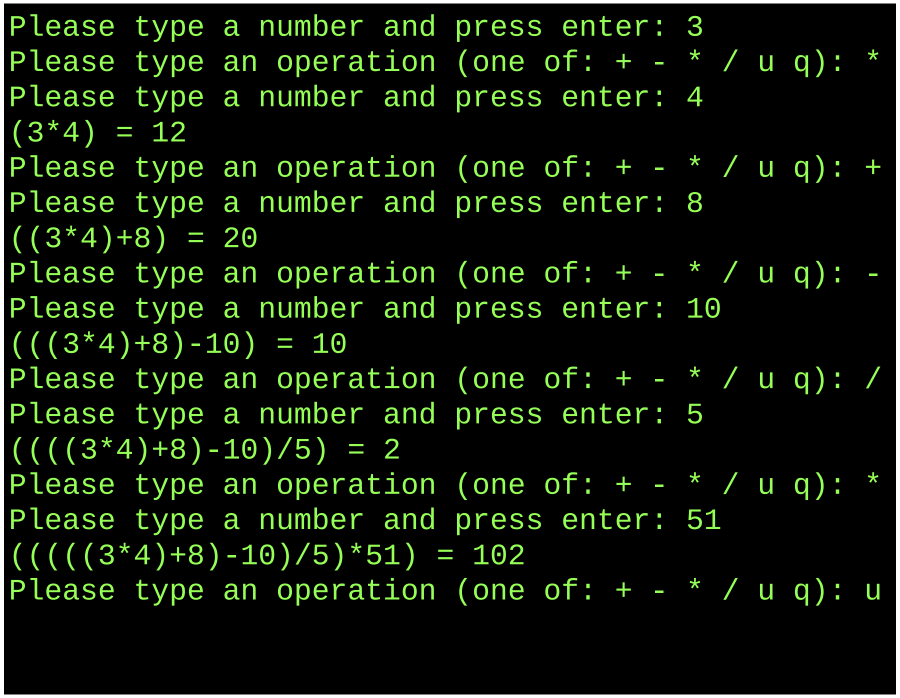
Variables

allOperations

	operand1	operation	operand2	result
0	3	'*'	4	12
1	12	'+'	8	20
2	20	'-'	10	10
3	10	'/'	5	2
4	2	'*'	51	102

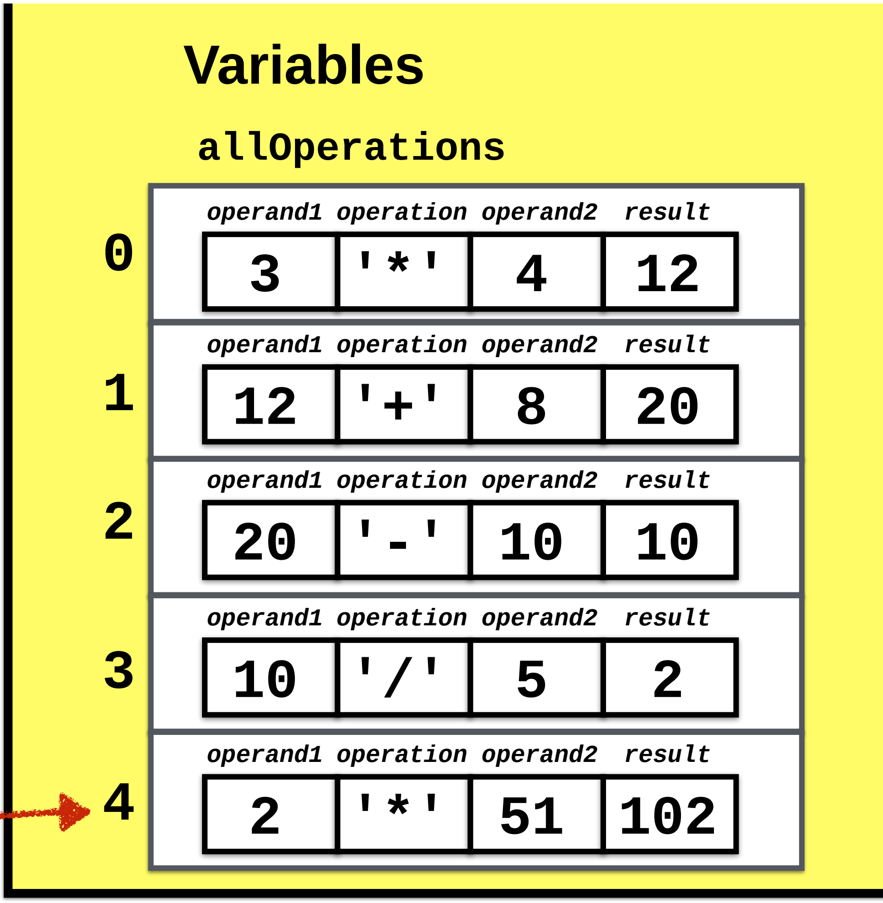
Now our undo should work

Calculator66



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
```

Variables



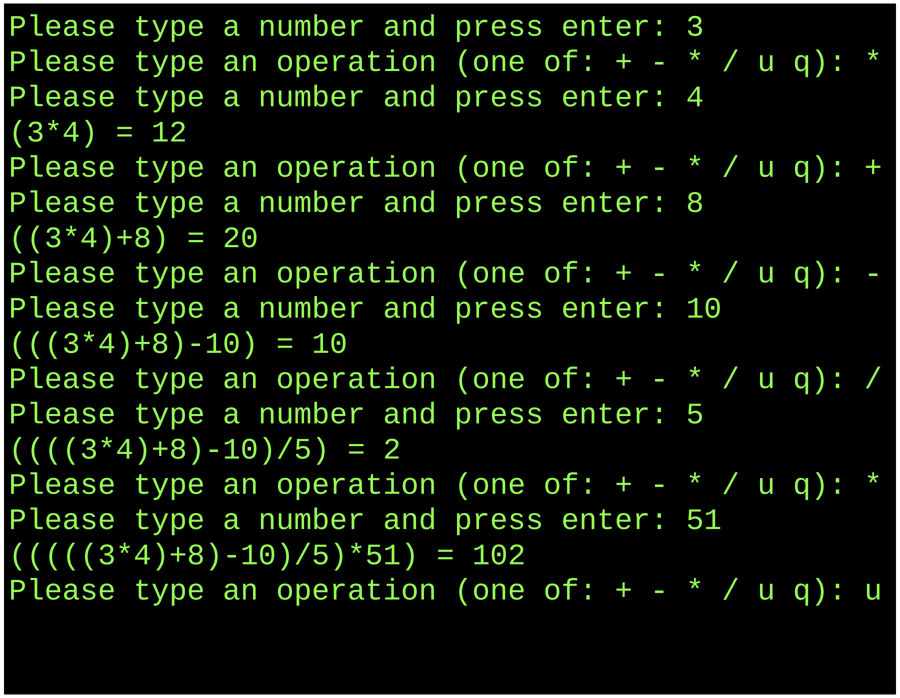
allOperations

	operand1	operation	operand2	result
0	3	'*'	4	12
1	12	'+'	8	20
2	20	'-'	10	10
3	10	'/'	5	2
4	2	'*'	51	102

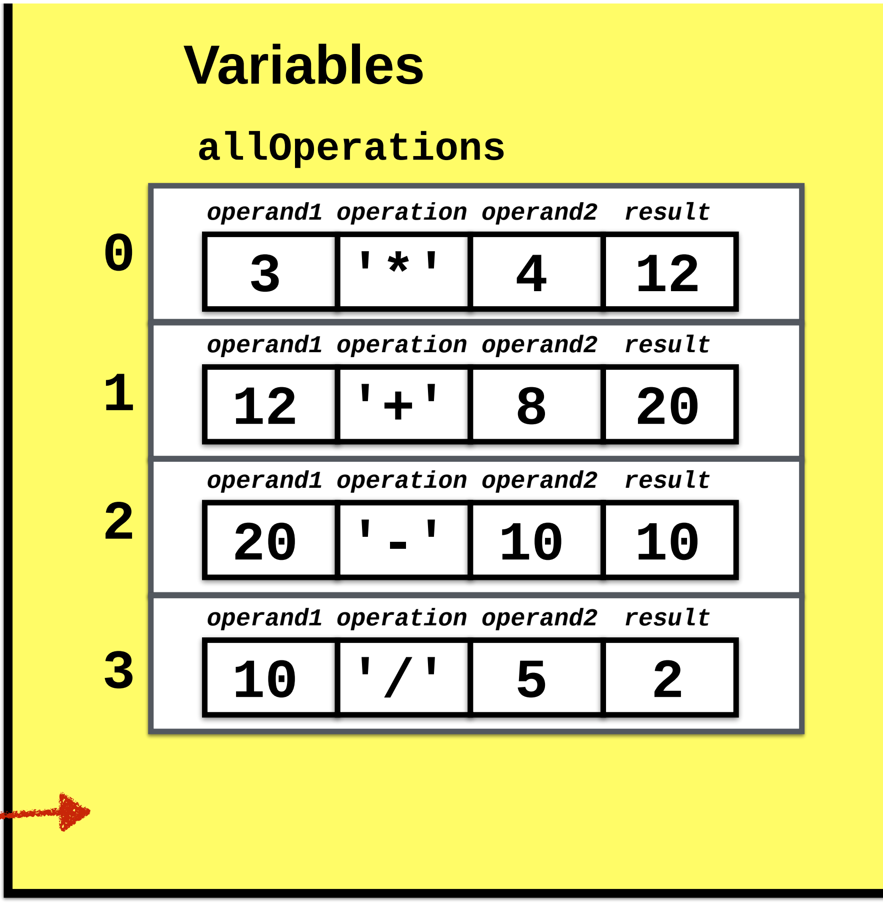
Remove last element of vector

```
allOperations.pop_back();
```

Calculator66



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
```



Variables

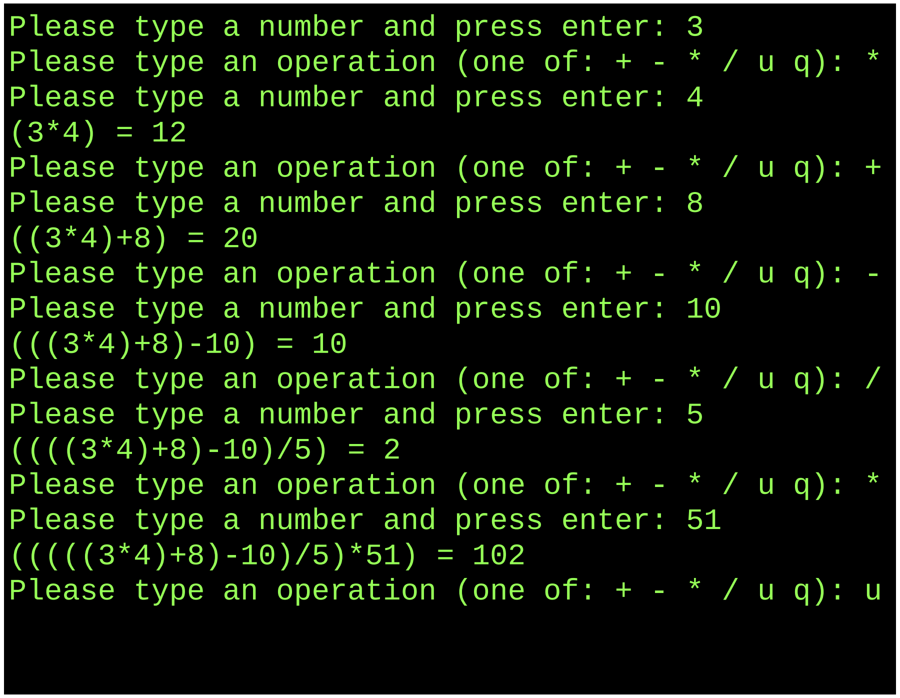
`allOperations`

	operand1	operation	operand2	result
0	3	'*'	4	12
1	12	'+'	8	20
2	20	'-'	10	10
3	10	'/'	5	2

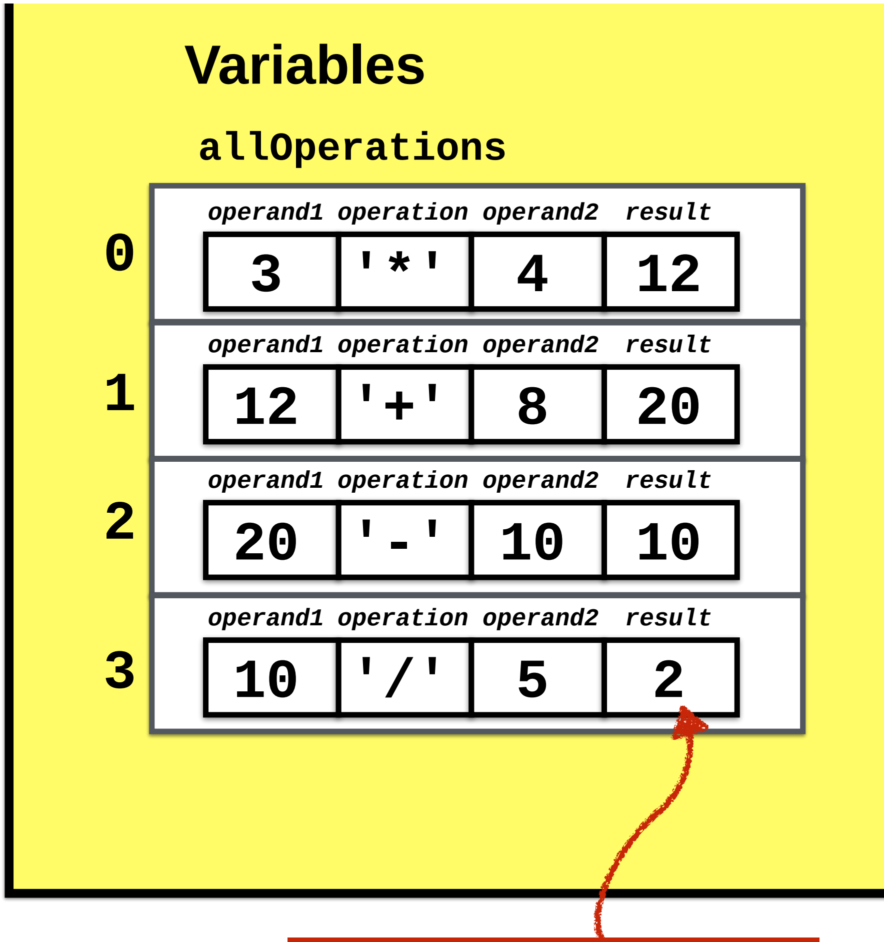
Remove last element of vector

```
allOperations.pop_back();
```

Calculator66



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
```



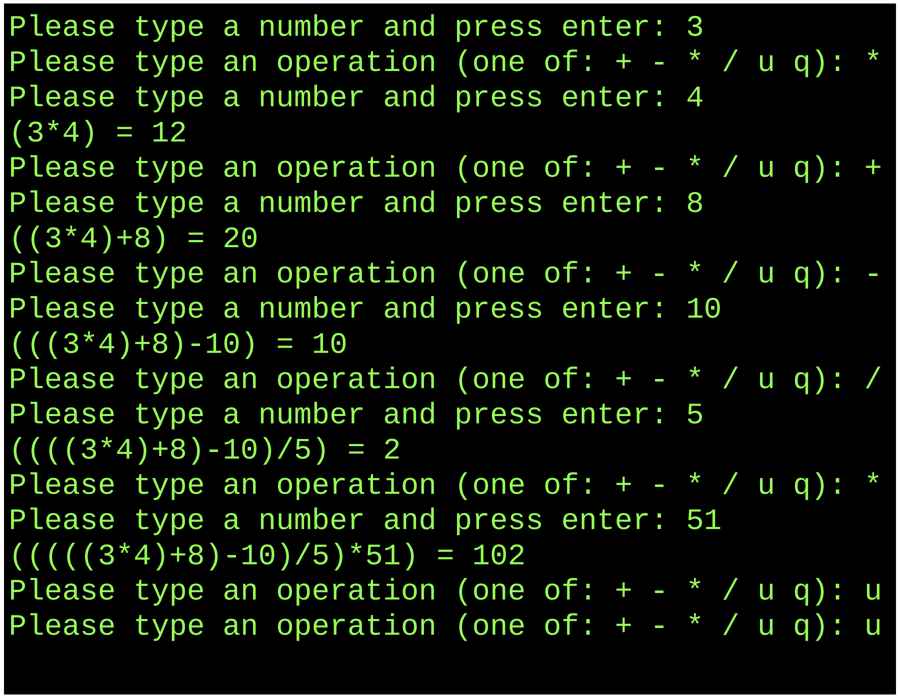
Variables

allOperations

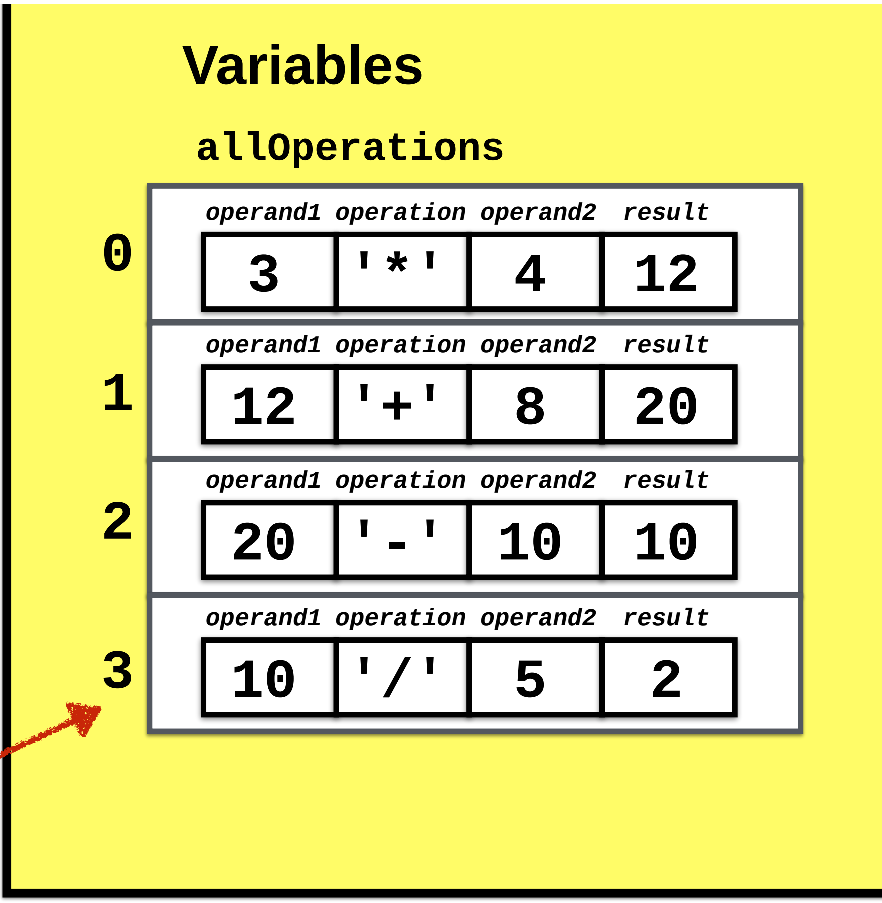
	operand1	operation	operand2	result
0	3	'*'	4	12
1	12	'+'	8	20
2	20	'-'	10	10
3	10	'/'	5	2

That is right!

Calculator66



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
```



Variables

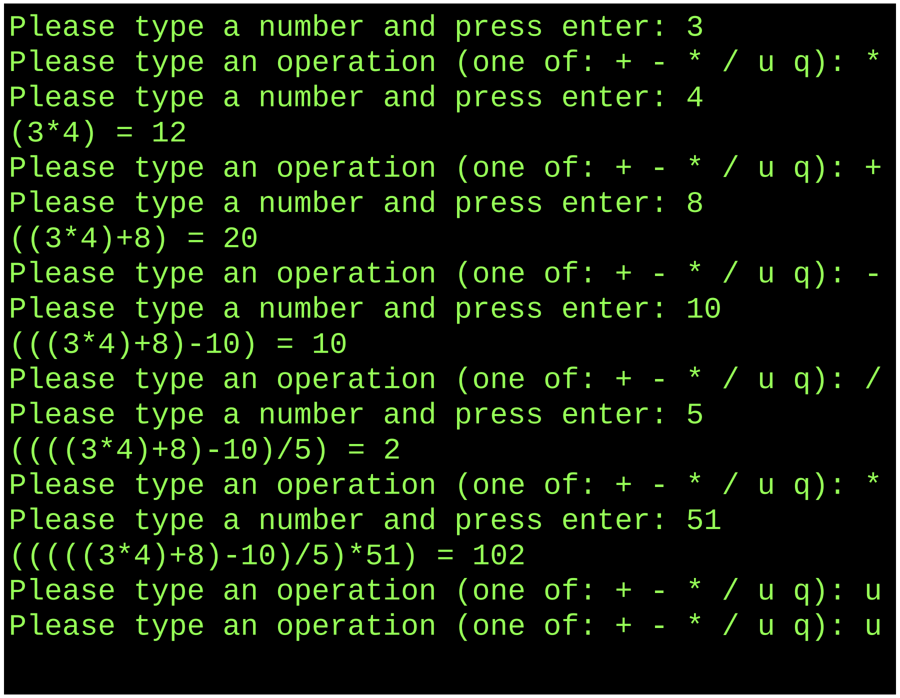
`allOperations`

	operand1	operation	operand2	result
0	3	'*'	4	12
1	12	'+'	8	20
2	20	'-'	10	10
3	10	'/'	5	2

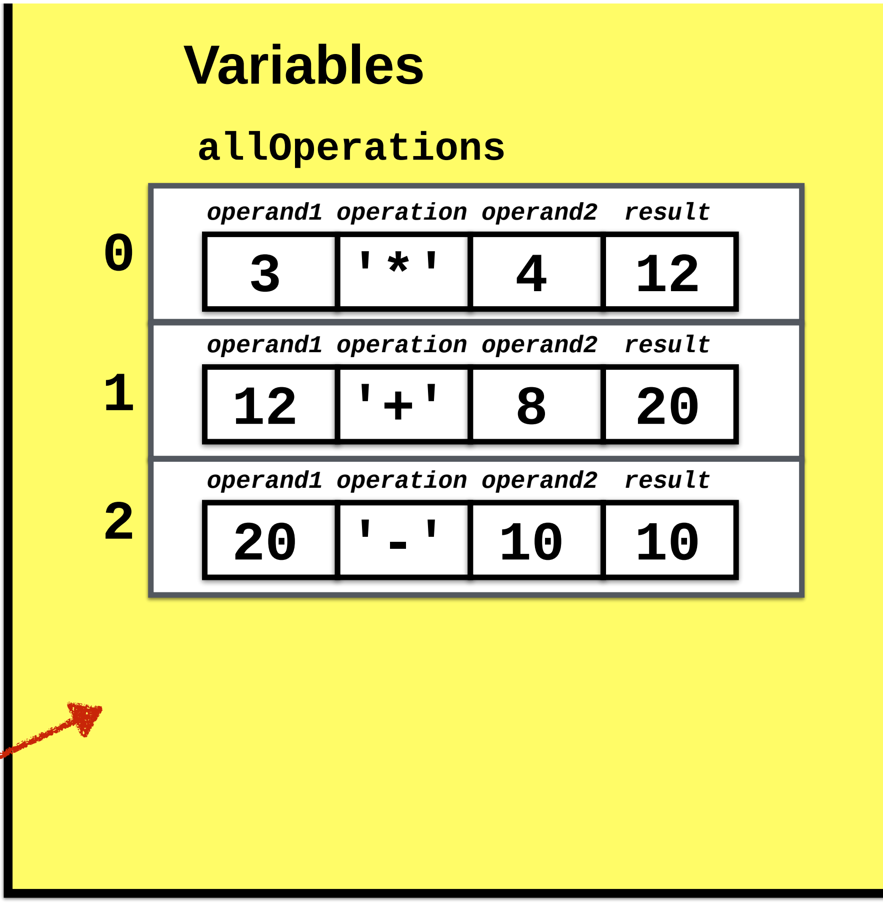
Remove last element of vector

```
allOperations.pop_back();
```

Calculator66



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
```



Variables

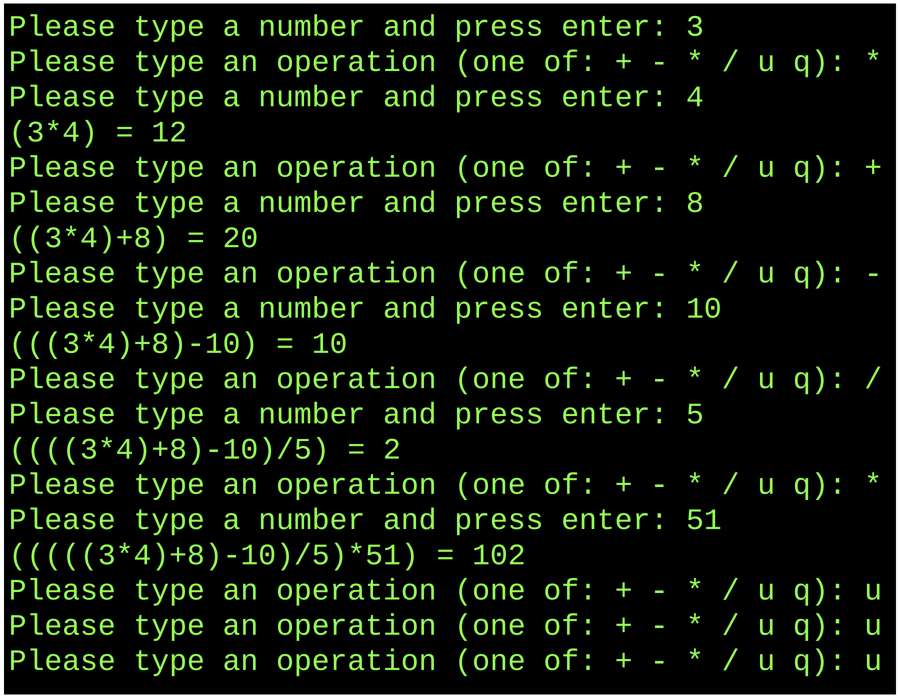
`allOperations`

	operand1	operation	operand2	result
0	3	'*'	4	12
1	12	'+'	8	20
2	20	'-'	10	10

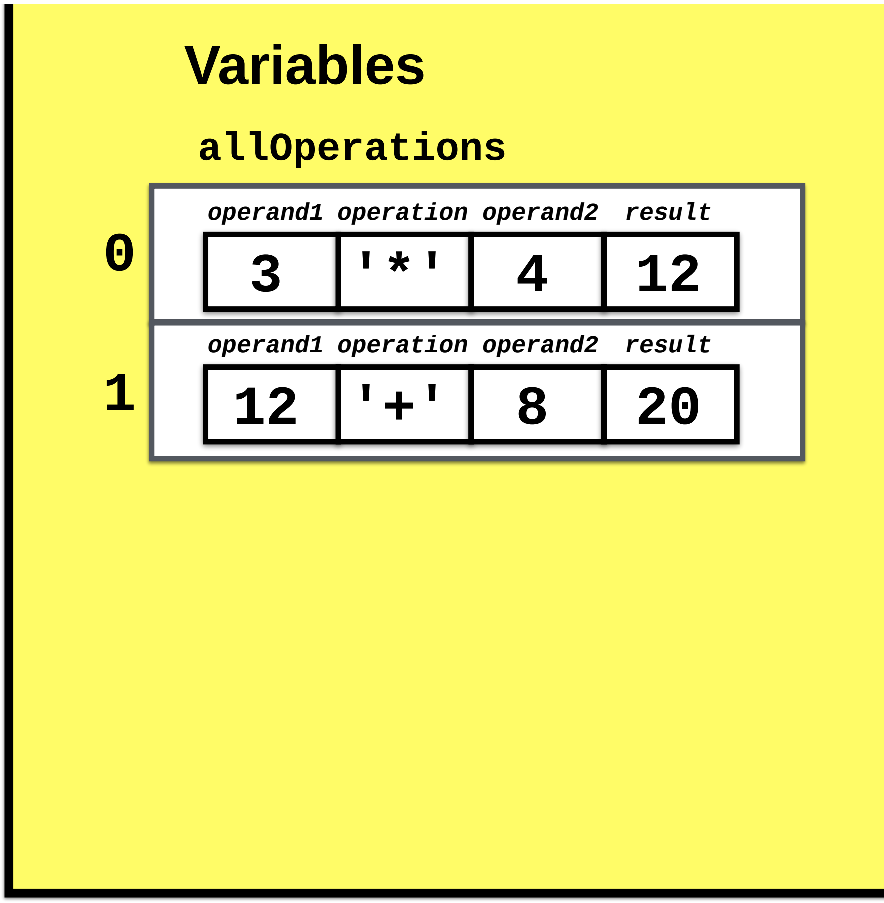
Remove last element of vector

`allOperations.pop_back();`

Calculator66



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
```

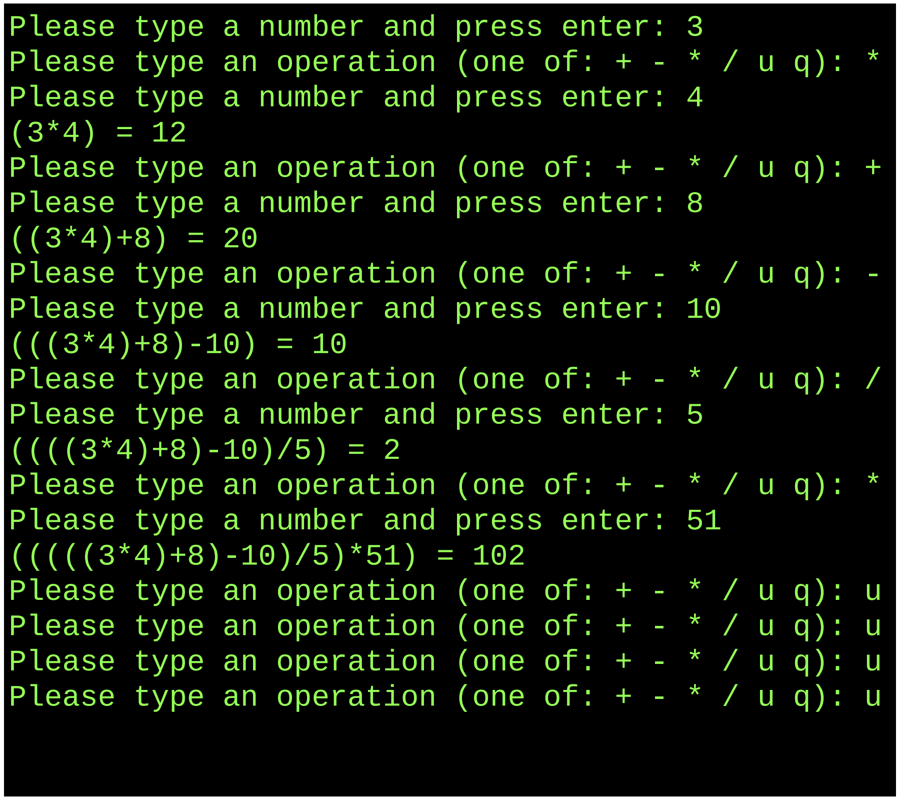


Variables

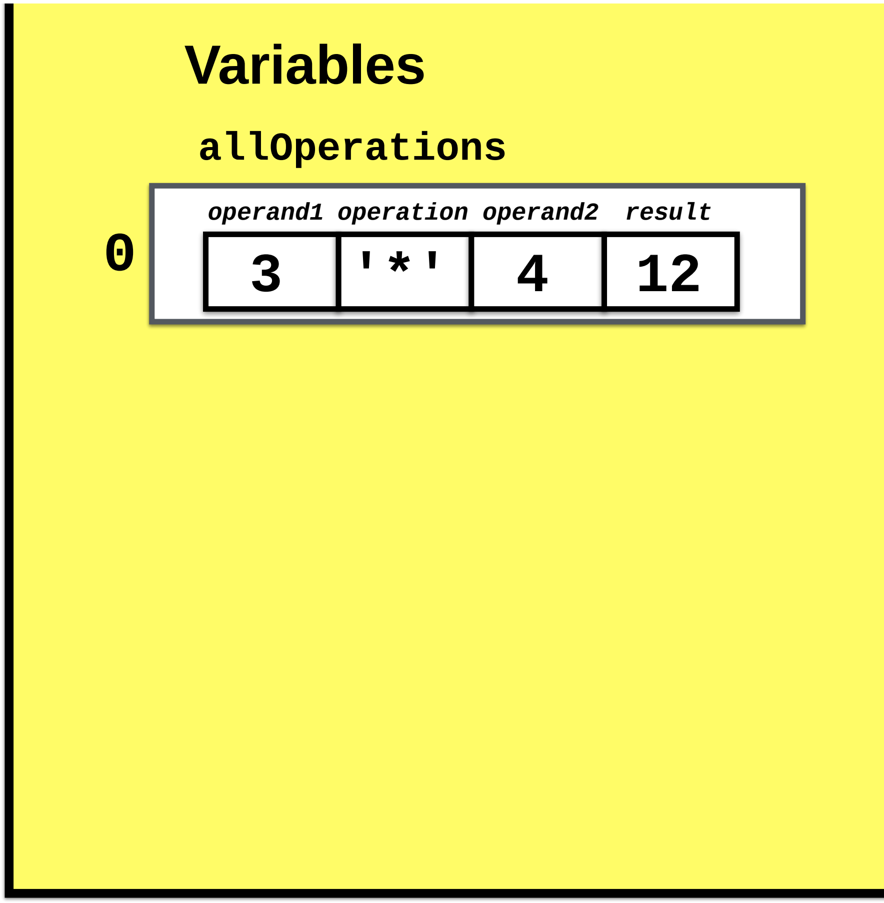
allOperations

	operand1	operation	operand2	result
0	3	'*'	4	12
1	12	'+'	8	20

Calculator66



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
```

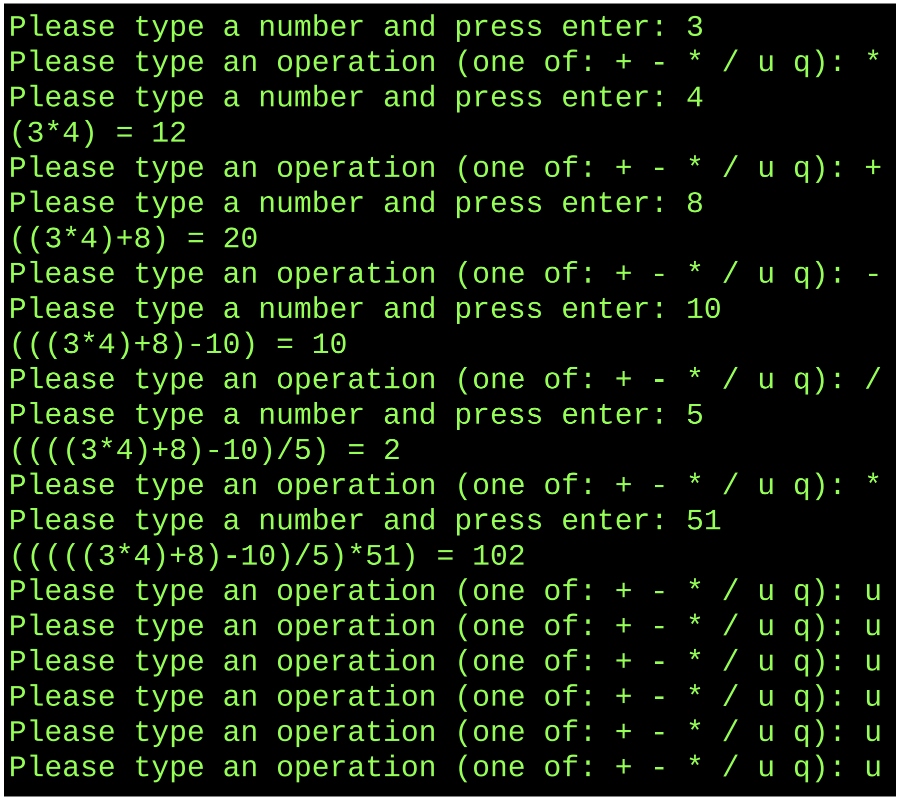


Variables

allOperations

	operand1	*operation*	*operand2*	*result*
0	3	'*'	4	12

calculator66

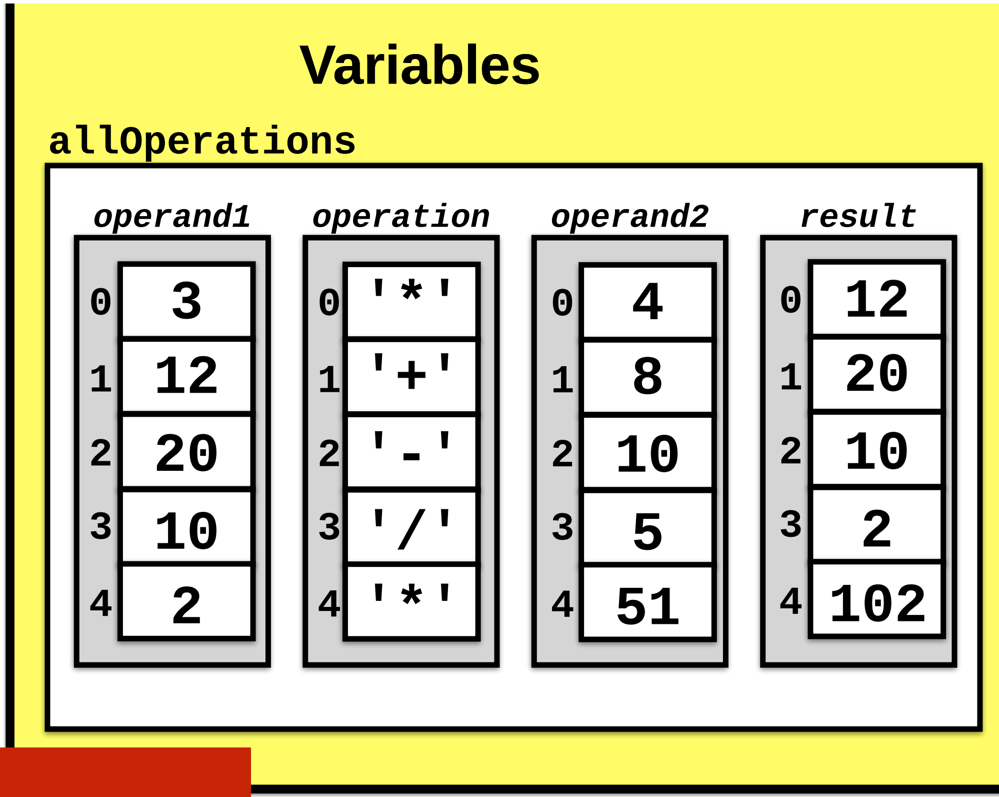


```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
```

Variables

`allOperations`

What should happen now ?



Variables

allOperations

	operand1	operation	operand2	result
0	3	'*'	4	12
1	12	'+'	8	20
2	20	'-'	10	10
3	10	'/'	5	2
4	2	'*'	51	102

Can we do this as a Struct of Vectors ?

calculator.cpp (Version 66 - Branch 01)

```
#include <iostream>
#include <vector>  // this enables the program to use C++ Vector data types

// Define new data type "operationEquation" to represent all equations
struct operationEquation
{
   std::vector <float> operand1;
   std::vector <char> operation;
   std::vector <float> operand2;
   std::vector <float> result;
};
```

A struct of vectors

calculator.cpp (Version 66 - Branch 01)

```cpp
#include <iostream>
#include <vector>  // this enables the program to use C++ Vector data types

// Define new data type "operationEquation" to represent all equations
struct operationEquation
{
   std::vector <float> operand1;
   std::vector <char> operation;
   std::vector <float> operand2;
   std::vector <float> result;
};
```

A struct of vectors

main()

```cpp
   // Let's declare our variables
   operationEquations allOperations;  // struct storing operands, operator, result
```

Declaration for struct-defined variable

calculator.cpp (Version 66 - Branch 01)

```
#include <iostream>
#include <vector>  // this enables the program to use C++ Vector data types

// Define new data type "operationEquation" to represent all equations
struct operationEquation
{
   std::vector <float> operand1;
   std::vector <char> operation;
   std::vector <float> operand2;
   std::vector <float> result;
};
```

A struct of vectors

```
// Function defined to perform specified operation on operands
bool performOperation(struct operationEquations &op) {
```

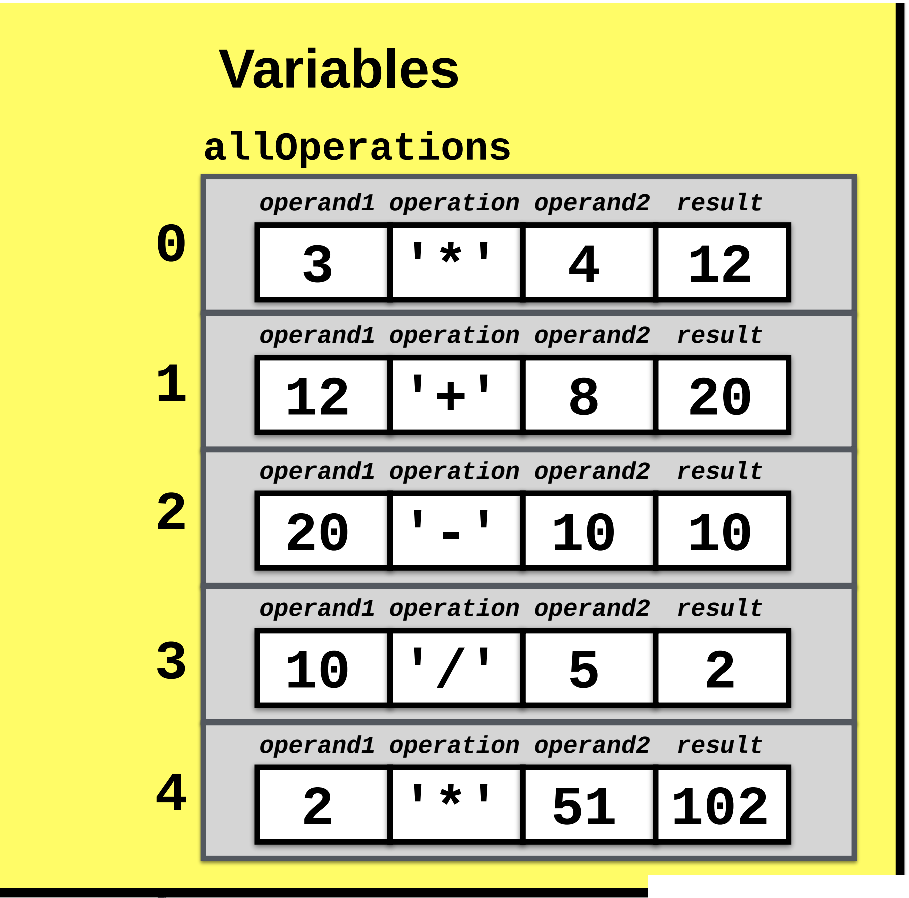
```
// Output the entire current equation to the screen
bool outputCurrentEquation(struct operationEquations &ops) {
```

Struct-defined variable as a function argument

main()

```
   // Let's declare our variables
   operationEquations allOperations;  // struct storing operands, operator, result
```

Declaration for struct-defined variable



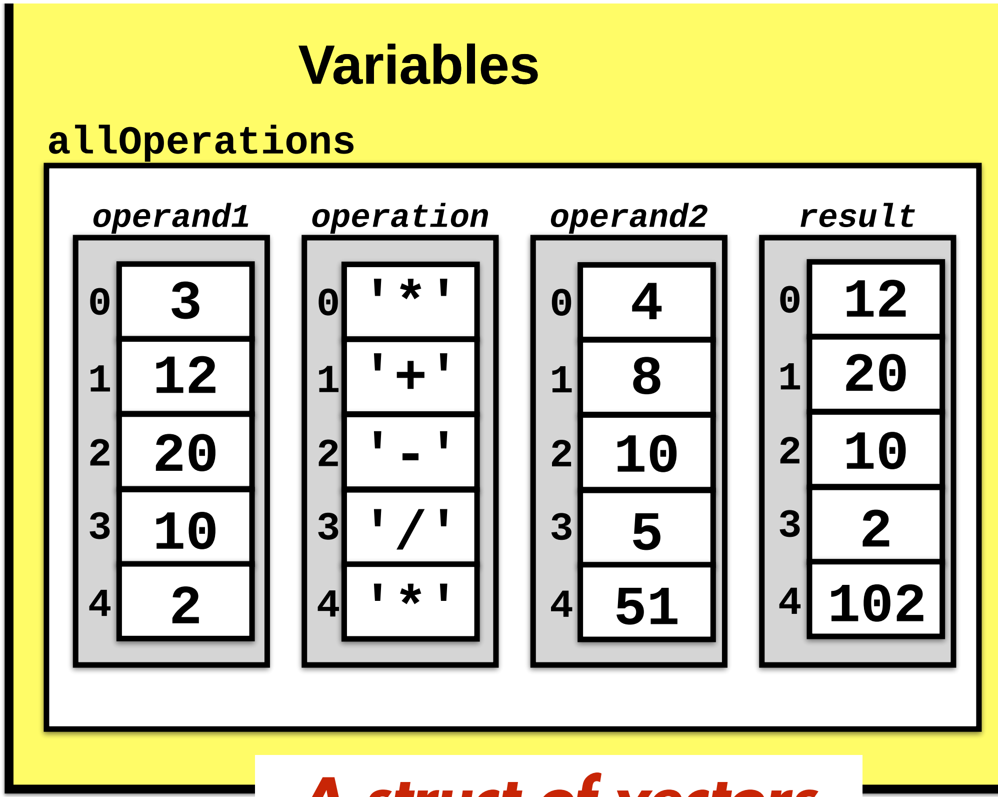
Variables

allOperations

	operand1	operation	operand2	result
0	3	'*'	4	12
1	12	'+'	8	20
2	20	'-'	10	10
3	10	'/'	5	2
4	2	'*'	51	102

```
struct operationEquation
{
   float operand1;
   char operation;
   float operand2;
   float result;
};
   std::vector <operationEquation> allOperations;
```

A vector of structs



Variables

allOperations

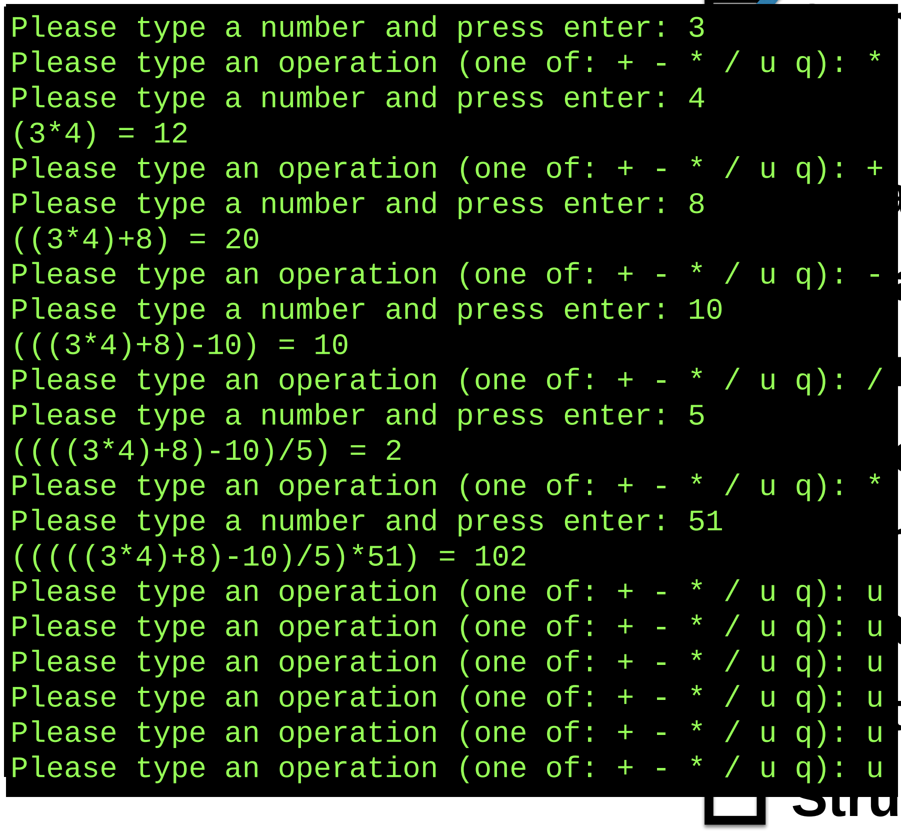
	operand1		operation		operand2		result
0	3	0	'*'	0	4	0	12
1	12	1	'+'	1	8	1	20
2	20	2	'-'	2	10	2	10
3	10	3	'/'	3	5	3	2
4	2	4	'*'	4	51	4	102

A struct of vectors

```
struct operationEquation
{
   std::vector <float> operand1;
   std::vector <char> operation;
   std::vector <float> operand2;
   std::vector <float> result;
};
```

Done

calculator66



```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
```

☑ Program Structure

☑ npile/Execute

rators

a Types

ables

r Input/Output

ctions

nching

ators

tors

Structs

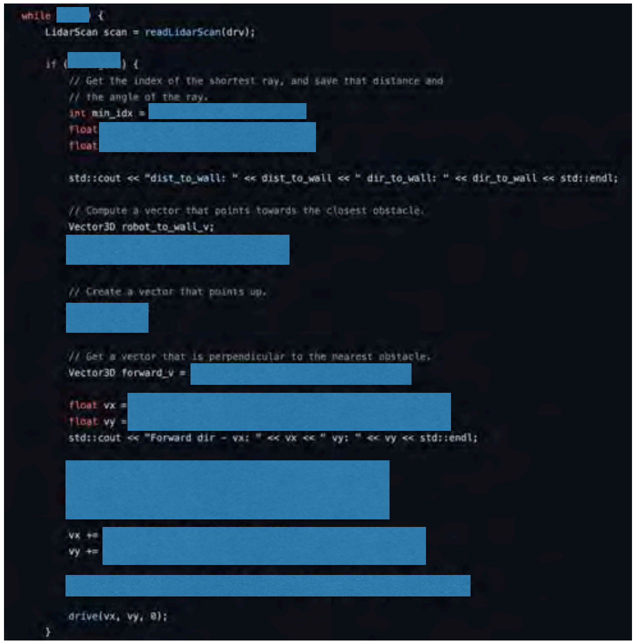
□ File Input/Output

calculator71

Save and load calculator results across executions

Now

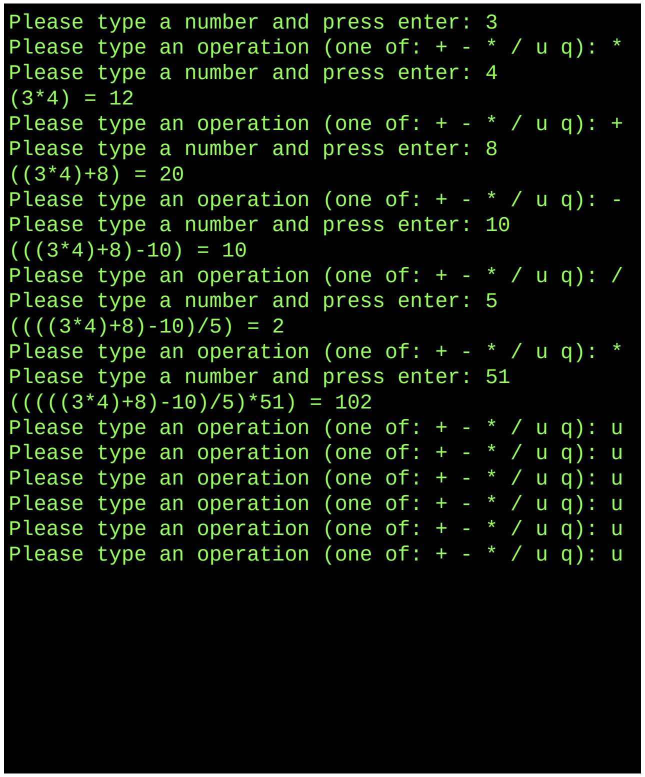
wall_follower.cpp - Project 1



```
while [redacted] {
    LidarScan scan = readLidarScan(drv);

    if ([redacted]) {
        // Get the index of the shortest ray, and save that distance and
        // the angle of the ray.
        int min_idx = [redacted]
        float [redacted]
        float

        std::cout << "dist_to_wall: " << dist_to_wall << " dir_to_wall: " << dir_to_wall << std::endl;

        // Compute a vector that points towards the closest obstacle.
        Vector3D robot_to_wall_v;
        [redacted]

        // Create a vector that points up.
        [redacted]

        // Get a vector that is perpendicular to the nearest obstacle.
        Vector3D forward_v = [redacted]

        float vx = [redacted]
        float vy =
        std::cout << "Forward dir - vx: " << vx << " vy: " << vy << std::endl;

        [redacted]

        vx += [redacted]
        vy +=

        [redacted]

        drive(vx, vy, 0);
    }
```

Done

calculator66

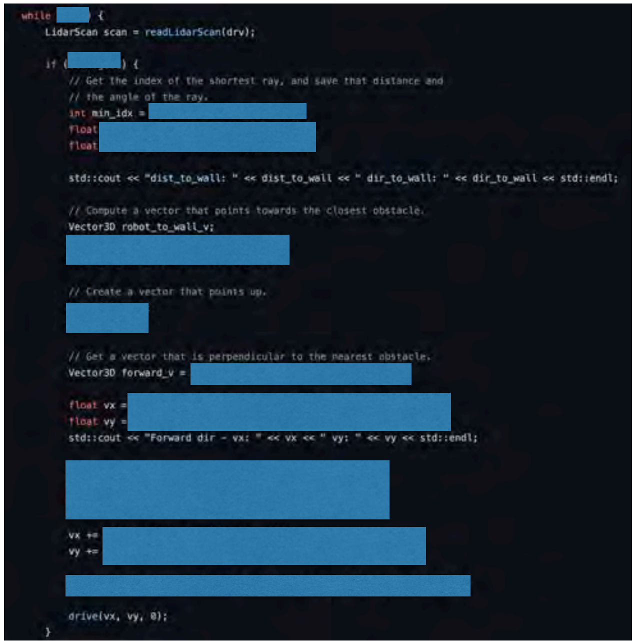


```
Please type a number and press enter: 3
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 4
(3*4) = 12
Please type an operation (one of: + - * / u q): +
Please type a number and press enter: 8
((3*4)+8) = 20
Please type an operation (one of: + - * / u q): -
Please type a number and press enter: 10
(((3*4)+8)-10) = 10
Please type an operation (one of: + - * / u q): /
Please type a number and press enter: 5
((((3*4)+8)-10)/5) = 2
Please type an operation (one of: + - * / u q): *
Please type a number and press enter: 51
(((((3*4)+8)-10)/5)*51) = 102
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
Please type an operation (one of: + - * / u q): u
```

- ☑ **Program Structure**
- ☑ **Compile/Execute**
- ☑ **Operators**
- ☑ **Data Types**
- ☑ **Variables**
- ☑ **User Input/Output**
- ☑ **Functions**
- ☑ **Branching**
- ☑ **Iterators**
- ☑ **Vectors**
- ☑ **Structs**
- □ **File Input/Output**

Now

wall_follower.cpp - Project 1



```
while [redacted] {
    LidarScan scan = readLidarScan(drv);

    if ([redacted]) {
        // Get the index of the shortest ray, and save that distance and
        // the angle of the ray.
        int min_idx = [redacted]
        float [redacted]
        float

        std::cout << "dist_to_wall: " << dist_to_wall << " dir_to_wall: " << dir_to_wall << std::endl;

        // Compute a vector that points towards the closest obstacle.
        Vector3D robot_to_wall_v;
        [redacted]

        // Create a vector that points up.
        [redacted]

        // Get a vector that is perpendicular to the nearest obstacle.
        Vector3D forward_v = [redacted]

        float vx = [redacted]
        float vy =
        std::cout << "Forward dir - vx: " << vx << " vy: " << vy << std::endl;

        [redacted]

        vx += [redacted]
        vy +=

        [redacted]

        drive(vx, vy, 0);
    }
```

calculator71

Save and load calculator results across executions

Michigan Robotics 102 - robotics102.org

A laser range scan is provided as a struct of vectors

C++ Vectors and Structs

Robotics 102

Introduction to AI and Programming
University of Michigan and Berea College
Fall 2021