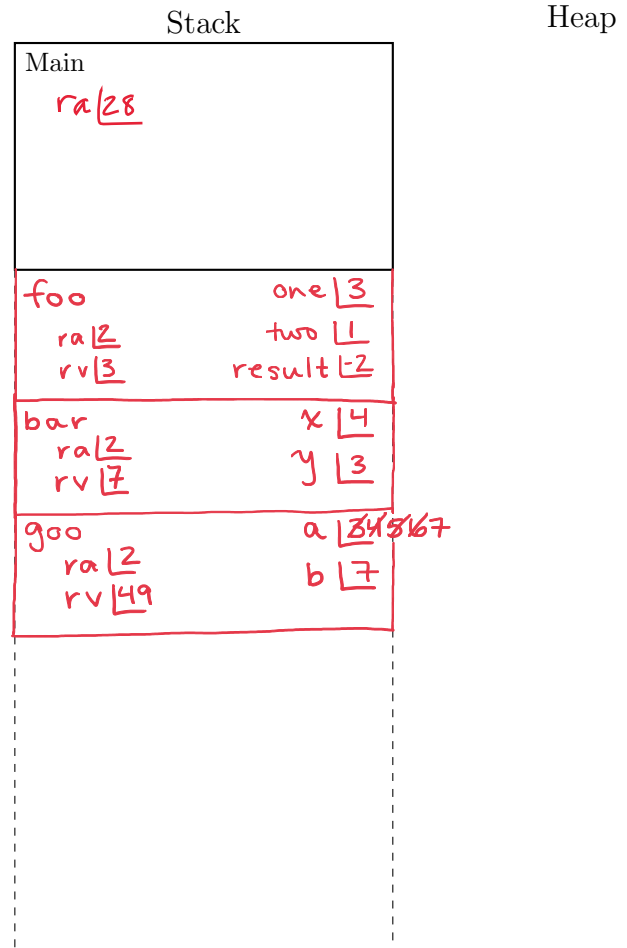


1. Given the code listing below, draw an environment diagram **paused at the moment line 3** is reached. In each frame on the stack, remember to include a space for the return address and return value, if any.

```

1 let main = async() => {
2   print(goo(foo(3, 1), bar(4, 3)));
3   // Pause here
4 };
5
6 let foo = (one: number, two: number): number => {
7   let result = two - one;
8   return one;
9 };
10
11 let bar = (x: number, y: number): number => {
12   if (x % y === 0) {
13     print("Go");
14   } else {
15     print("Heels");
16   }
17   return x + y;
18 };
19
20 let goo = (a: number, b: number): number => {
21   while (a < b) {
22     print("ayooo");
23     a++;
24   }
25   return a * b;
26 };
27
28 main();

```



The following questions are about the state of the program at the moment the evaluation is **paused at line 3**. If a question is asked about a name that has not been initialized, then respond with **undefined**.

- 1.1 From goo's frame, what is the value of a?
7
- 1.2 From goo's frame, how many times do you iterate through the while loop? 4
- 1.3 Write a function call to bar that results in "Go" being printed. bar(4,2)
- 1.4 From foo's frame, what is the value of result?
-2

1.5 What is the printed output of this program once it completes?

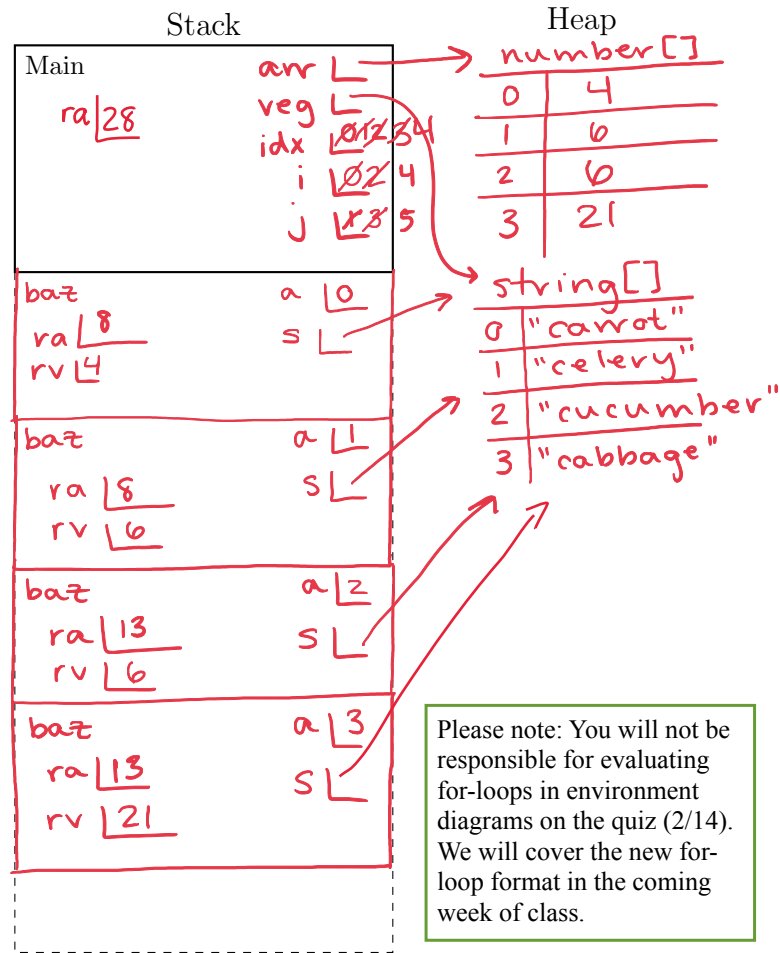
Heels ayooo ayooo ayooo ayooo 49

2. Given the code listing below, draw an environment diagram then answer the questions that follow. In each frame on the stack, remember to include a space for the return address and return value, if any.

```

1 let main = async () => {
2   let arr: number[] = [];
3   let veg = ["carrot", "celery", "cucumber", "
4     cabbage"];
5
6   let idx = 0;
7
8   for (let i = 0; i < veg.length; i = i + 2) {
9     arr[idx] = baz(idx, veg);
10    idx++;
11  }
12
13  for (let j = 1; j < veg.length; j = j + 2) {
14    arr[idx] = baz(idx, veg);
15    idx++;
16  }
17
18  let baz = (a: number, s: string[]): number => {
19    if (a % 2 === 0) {
20      print("Rah_rah");
21      return s.length + a;
22    } else {
23      print("Carolina");
24      return s[a].length * a;
25    }
26  };
27
28  main();

```



Please note: You will not be responsible for evaluating for-loops in environment diagrams on the quiz (2/14). We will cover the new for-loop format in the coming week of class.

The following questions are about the state of the program after it has finished running. If a question is asked about a name that has not been initialized, then respond with **undefined**.

2.1 How many times is the function `baz()` called?

4

2.2 What is the final value of `idx` (when the code is finished running)?

4

2.3 What is the printed output of the function after it has run completely?

Rah rah, Carolina, Rah rah, Carolina

2.4 In the spaces provided, write out the final values held in the array `arr`. Not all indices must be filled.

4	6	6	21			
0	1	2	3	4	5	6

3. Trace through the following code and answer the question on the right.

```
1 let animalNoise : string[] = [];  
2 for(let i = 0; i < 7; i++) {  
3   if (i % 3 === 0) {  
4     animalNoise[animalNoise.length] = "moo";  
5     i = animalNoise.length;  
6     animalNoise[i] = "bark";  
7   } else if (i < 3) {  
8     animalNoise[i] = "meow";  
9   } else {  
10    i++;  
11  }  
12 }  
13 let x = 5;  
14 for (let j = 0; j < animalNoise.length; j++) {  
15   if (x === 5) {  
16     animalNoise[x] = "baaaa";  
17     x = x + 2;  
18   } else {  
19     if ( x > 10) {  
20       animalNoise[j] = "woof";  
21       x -= 8;  
22     }  
23     x += 1;  
24   }  
25 }
```

3.1 What is the value of x after the following code runs? **4**

3.2 What is the length of the animalNoise array after the following code runs? **6**

3.3 Fill in the array outline with the elements that reside in each index. Unused indices are allowed.

"moo"	"bark"	"meow"	"moo"	"bark"	"woof"
0	1	2	3	4	5

3.4 Which strings are repeated twice in the animalNoise array after the following code runs? **"moo", "bark"**

3.5 At what indices in the animalNoise array do the previous strings appear? **0, 1, 3, 4**

4. Given a number array of positive integers, define and implement a function called `replicate` that replaces the number at index `i` with `arr[i]` copies of itself. For example, `replicate([3, 2, 1])` would return `[3, 3, 3, 2, 2, 2, 1]`. To receive credit for this question you must use `loop(s)`.

```
let replicate = (nums: number[]) => {
  let output = [];
  for (let i = 0; i < nums.length; i++) {
    let current = nums[i];
    for (let j = 0; j < current; j++) {
      output[output.length] = current;
    }
  }
  return output;
}
```

5. In the space below, write a function called `reverse` that takes in a number array and returns an array that is the same as the input array backwards. For example, the function call `reverse([1, 2, 3, 4])`; should return `[4, 3, 2, 1]`. To receive credit for this question you must use `loop(s)`.

```
let reverse = (arr: number[]): number[] => {
  for (let i = 0; i < arr.length / 2; i++) {
    let oppositeIndex = arr.length - i - 1;
    let temp = arr[i];
    arr[i] = arr[oppositeIndex];
    arr[oppositeIndex] = temp;
  }
  return arr;
};
```

Alternate solution:

```
-----
let reverse = (arr: number[]): number[] => {
  let rv: number [] = [];
  for (let i = arr.length - 1; i >= 0; i--) {
    rv[rv.length] = arr[i];
  }
  return rv;
};
```