

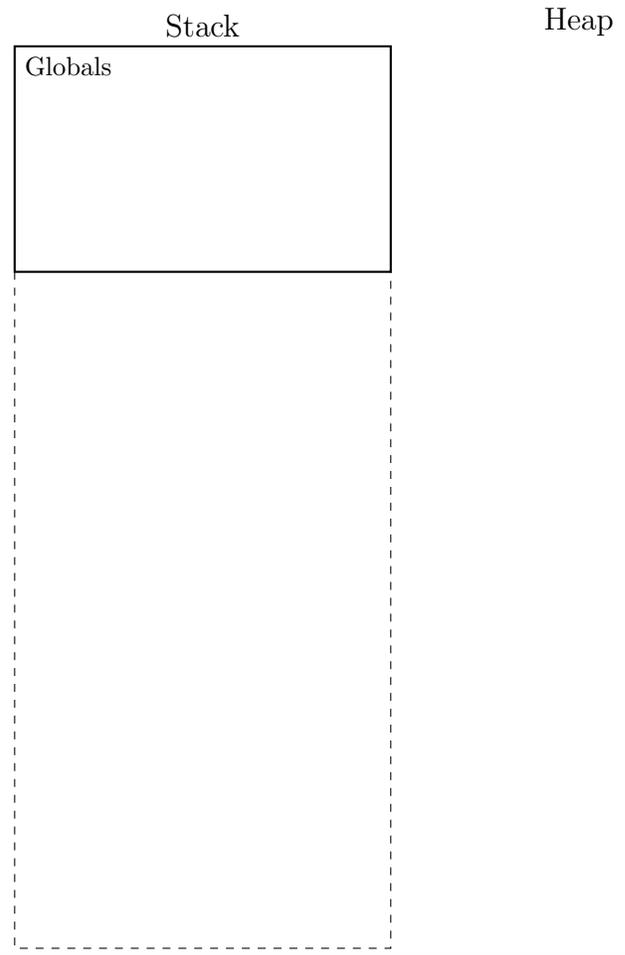
Please print out and fill in this worksheet by hand. Make sure that when submitting your assignment to Gradescope that you scan each page as a whole page, and that they are uploaded in the correct order and in the proper orientation. **Points will be deducted for not following these guidelines.**

1. 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 export let main = async () => {
2   let foo = 7;
3   foo = scopeInvestigator(foo);
4   foo -= 5;
5   print(foo);
6 };
7
8 let scopeInvestigator = (n: number): number => {
9   let foo = 10;
10  print(foo);
11  {
12    let bar;
13    let foo = 20 * n;
14    print(foo);
15    {
16      if (foo > 100) {
17        let foo = 12;
18        print(foo);
19      }
20      foo /= 4;
21      bar = 2;
22      print(foo);
23    }
24    print(foo);
25    print(bar);
26  }
27  print(foo);
28  return foo;
29 };
30
31 main();

```



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**.

1.1 What is the printed output of the above code? Put the values of each **print** statement in the box whose number corresponds to the line number that the **print** statement occurs on.

5	10	14	18	22	24	25	27

1.2 How many blocks are nested within the block for the `scopeInvestigator` function?

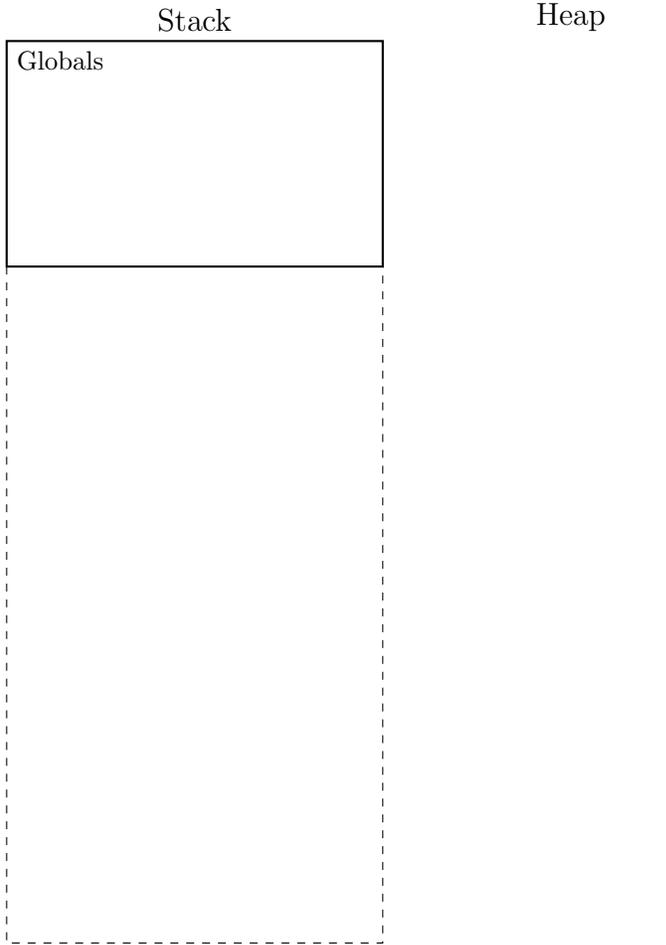
1.3 On line 11, a block opens up. On which line does this block close?

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 export let main = async () => {
2   let i: number = 18;
3   let arr = [0, 1, 1];
4   i = baz(i, arr);
5 };
6
7 let baz = (i: number, arr: number[]): number => {
8   for (let i = 0; i < 5; i++) {
9     arr[arr.length] = arr[arr.length - 1] +
10      arr[arr.length - 2];
11   }
12   i /= 9;
13   return i;
14 };
15
16 main();

```



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 different variables named `i` are defined in this program?

2.2 What is the final value of `i` in the block that begins on line 8 and ends on line 11?

2.3 What is the final value of `i` in the block that begins on line 7 and ends on line 14?

2.4 How many blocks are nested within the block for the `baz` function?

2.5 What is the length of `arr` after the code has finished running?

2.6 What is the final value of `i` right before the program finishes its execution on line 5?

2.7 Fill in the elements of the `arr` array in their proper index boxes below. Not all indices must be used.

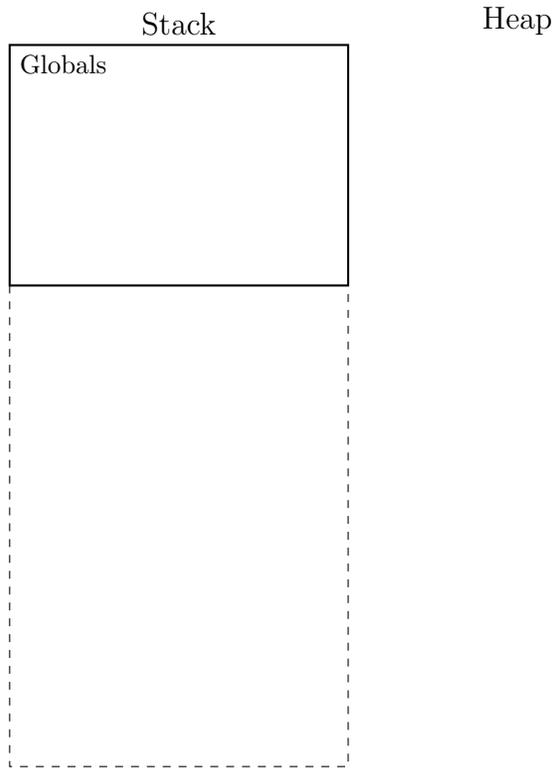
0	1	2	3	4	5	6	7	8	

3. Given the code listing below, draw an environment diagram **paused at the moment that line 14** is reached. 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 a: string[] = ["a"];
2 let b: string[] = ["b"];
3
4 export let main = async () => {
5   swap();
6   print(a[0] + "_-_" + b[0]);
7 };
8
9 let swap = (): void => {
10  let temp = a;
11  a = b;
12  b = temp;
13  print(a[0] + "_-_" + b[0]);
14  // Break Here!
15 };
16
17 main();

```



The following questions are about the state of the program at the moment the evaluation is **paused at line 14**. If a question is asked about a name that has not been initialized, then respond with **undefined**.

3.1 From `swap`'s frame, use name resolution to look up the name `temp`. What frame is it found in?

3.2 From `swap`'s frame, use name resolution to look up the name `a`. What frame is it found in?

3.3 From `swap`'s frame, use name resolution to look up the value of the expression `temp[0]`. What is the value?

3.4 From `swap`'s frame, use name resolution to look up the value of the expression `a[0]`. What is the value?

3.5 From `swap`'s frame, use name resolution to look up the value of the expression `b[0]`. What is the value?

3.6 From `main`'s frame, use name resolution to look up the value of the expression `temp[0]`. What is the value?

3.7 From `main`'s frame, use name resolution to look up the value of the expression `a[0]`. What is the value?

3.8 From `main`'s frame, use name resolution to look up the value of the expression `b[0]`. What is the value?

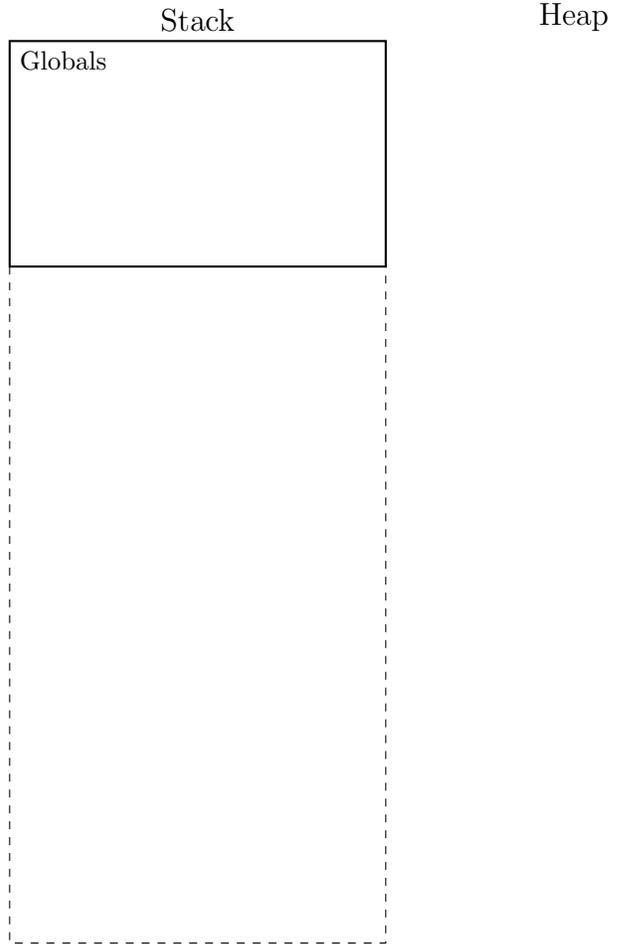
3.9 What is the printed output of this program once it completes?

4. 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 class Person {
2   name: string;
3   age: number;
4 }
5
6 export let main = async () => {
7   let a: Person = new Person();
8   a.name = "Jim";
9   a.age = 32;
10  let b = [10, 20, 30];
11  let c = 2;
12  hbd(a, b, c);
13  print("a:" + a.age + " b:" + b[0] + " c:" + c);
14 };
15
16 let hbd = (a: Person, b: number[], c: number):void
17   => {
18   a.age++;
19   b = [17];
20   c = c * 3;
21   print("a:" + a.age + " b:" + b[0] + " c:" + c);
22 };
23 main();

```



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**.

**4.1** From the **hbd** frame, what is the type of the variable **a**?

**4.2** How many properties does an object of type **Person** have?

**4.3** How many **Person** objects are on the heap?

**4.4** After the program has executed, is it possible to access the array that was initialized on line 10 from **main**? (Yes or No)

**4.5** From the **hbd** frame, is the variable **b** a value type or a reference type?

**4.6** What is the printed output of this program?