Summer Work
AP Computer Science A

Ramapo High School
Summer 2019

Directions

This packet is designed to help you review computer programming skills to prepare for AP Computer Science A. Everyone taking AP Computer Science A is expected to complete this packet. It will be collected during the first few days of school and graded for correctness.

There are two versions of this packet. **You only need to complete one of them, depending on which language you know.**

- If you have previously taken Computer Science Honors or Computer Science for Engineers, start the Python version on the next page.
- If you have previously taken AP Computer Science Principles, flip to the middle of the packet and start the Javascript version.
- If you have not taken any of those classes, but you are signed up for AP Computer Science A complete the version for the language you are most familiar with. If you are not sure which version to complete, please contact mcaulfield@rih.org for further directions. No matter what, you need to complete one version of the summer packet.

There are two parts, multiple-choice and free-response. For the multiple-choice part, there is an answer sheet. Fill out this answer sheet and be ready to hand it in on the first day of school. The free-response consists of writing code to solve programming problems. For each problem, take a screenshot, and send your screenshots to mcaulfield@rih.org by Friday September 13, 2019. There are more detailed directions in each version.

Have a great summer!

Name ____

Summer packet: AP Computer Science A

Python version

For students coming from CS for Engineers or CS Honors

Summer 2019

Directions

- There are two versions of this packet. You only have to do one version, depending on which language you know. This is the Python version. You should only do this version if you are familiar with Python. If you previously took Computer Science Honors or Computer Science for Engineers, this is the right version for you.
- There are two parts to this packet: multiple-choice and free-response coding.

Part 1: Multiple-choice

- Multiple choice: Please write down your answers (A) – (D) on the answer sheet.
- Be prepared to hand in your answer sheet on the first day of school.

Multiple choice answers

1. ________

2. ________

3. ________

4. ________

5. ________

6. ________

7. ________

8. ________

9. ________

10. ________

11. ________

12. ________

13. ________

14. ________

15. ________

16. ________

17. ________

18. ________

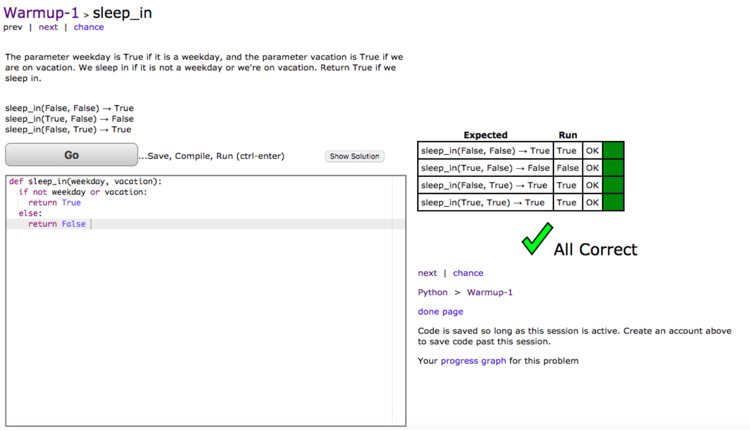
19. ________

20. ________

Part 2: Free-response coding

- Go to `https://codingbat.com/python`. For each problem, check your answer by clicking the "Go" button. If all the tests are "green" you got the problem right. Once you complete a function and verify that it's correct, please take a screenshot that includes your code the tests for correctness. Keep all your screenshots, and email them to mcaulfield@rih.org any time over the summer. Deadline is Friday, September 13, 2019 at the latest.

- Example screenshot: Include the directions, your code, and the tests that indicate you got the problem right.



The parameter weekday is True if it is a weekday, and the parameter vacation is True if we are on vacation. We sleep in if it is not a weekday or we're on vacation. Return True if we sleep in.

sleep_in(False, False) → True
sleep_in(True, False) → False
sleep_in(False, True) → True

```
def sleep_in(weekday, vacation):
  if not weekday or vacation:
    return True
  else:
    return False
```

Expected	Run		
sleep_in(False, False) → True	True	OK	🟩
sleep_in(True, False) → False	False	OK	🟩
sleep_in(False, True) → True	True	OK	🟩
sleep_in(True, True) → True	True	OK	🟩

✓ All Correct

Code is saved so long as this session is active. Create an account above to save code past this session.

Your progress graph for this problem

Problems

1. String-2: `double_char`
2. String-2: `count_hi`
3. Logic-2: `lone_sum`
4. Logic-2: `lucky_sum`
5. List-1: `first_last6`
6. List-1: `common_end`
7. List-1: `rotate_left3`
8. List-1: `middle_way`
9. List-2: `count_evens`
10. List-2: `centered_average`

Save all screenshots and send to mcaulfield@rih.org any time over the summer. Deadline is Friday, September 13, 2019 at the latest.

Multiple-choice questions

Choose the best answer for each question. Please write all answers on the answer sheet above.

1. Consider the code segment,

```
a = 2
b = a + 1
a += 2
print(b)
a = a * 2
print(a)
```

What is printed?

a)
```
3
8
```

b)
```
5
8
```

c)
```
3
4
```

d)
```
5
4
```

2. Consider the code segment,

```
s = "ab"
t = "c"
t += s
s = "d"
s += t
print(s)
```

What is printed?

a) `dcab`

b) `abcd`

c) `dcba`

d) `cabd`

3. What is printed when we do:

```
print(50 % 7 + 20 // 3 + 2 ** 3)
```

a) `10`
b) `12`
c) `15`
d) `17`

4. Consider the code segment:

   ```
   a = "5"
   b = "2"
   print(a + b, int(a) + float(b))
   ```

   What is printed?

   a) `7 7.0`

   b) `52 7`

   c) `52 7.0`

   d) `52.0 7`

5. Suppose `a` and `b` are some unknown numbers, such that `a % b` → 0. Which of the following are possible values of `a` and `b`?

   a) `a = 5, b = 7`
   b) `a = 7, b = 5`
   c) `a = 10, b = 2`
   d) `a = 2, b = 10`

6. If we type in the shell,

   ```
   >>> len("five" + "one")
   ```

   What is returned?

   a) `6`
   b) `7`
   c) `8`
   d) `51`

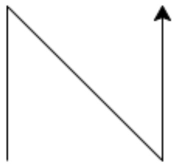
7. When we run the code,

   ```
   t = 1
   for x in range(3, 7, 2):
           t = t * x
   print(t)
   ```

   What is printed?

   a) `15`
   b) `105`
   c) `120`
   d) `360`

8. Consider the image, generated using Turtle Graphics:



Suppose that we have the lines:

```
import turtle
ada = turtle.Turtle()
```

which of the following is the code that will draw this image? You can assume that the turtle starts out facing to the right. And the `goto()` function moves the turtle but keeps it facing in the same direction.

a)
```
ada.forward(100)
ada.goto(100, 0)
ada.forward(100)
```

b)
```
ada.left(90)
ada.forward(100)
ada.goto(100, 0)
ada.forward(100)
```

c)
```
ada.goto(0, 100)
ada.forward(100)
ada.right(90)
ada.forward(100)
```

d)
```
ada.goto(100, 100)
ada.right(90)
ada.forward(100)
ada.right(90)
ada.forward(100)
```

9. A certain code segment prints,

```
10 6 2
```

which of the following matches this output?

a)
```
for i in range(10, 2):
          print(i, end = " ")
```

b)
```
for i in range(2, 10, 4):
          print(i, end = " ")
```

c)
```
for i in range(10, 1, 4):
          print(i, end = " ")
```

d)
```
for i in range(10, 1, -4):
          print(i, end = " ")
```

10. When we run the script,

```
import math
def f(L):
     sum = 0
     for x in L:
             sum += x ** 2
     return sum

def g(a, b):
     return math.sqrt(a - b)

def main():
     my_list = [2, 1, 5]
     print(f(my_list))

if __name__ == "__main__":
     main()
```

What is printed?

a) `8`

b) `30`

c) `64`

d) `70`

11. The function `fun()` takes a list as a parameter and returns a value. What is the best description of what `fun()` does?

```
def fun(L):
        s = 0
        for x in L:
                y = max(0, x)
                s += y
        return s
```

a) Returns the sum of all the numbers in `L`.

b) Returns the sum of the positive numbers in `L` only; negative numbers do not contribute to the total.

c) Returns how many positive numbers there are in `L`.

d) Returns `0`

12. What is printed when we do,

```
wordlist = ["math", "science", "music"]
for word in wordlist:
        print(len(word), end =  " ")
```

a) `math science english`

b)
```
math
science
english
```

c) `4 7 5`

d) `3 3 3`

13. What is printed as a result of the segment,

```
n = 1000
while n > 200:
        n //= 2
print(n)
```

a) `0`

b) `125`

c) `500`

d) `1000`

14. Which of the following expressions is `False` for any `float` value `x`?

a) `x < 1 or x > 3`

b) `x > 1 or x < 3`

c) `x < 1 and x > 3`

d) `x > 1 and x < 3`

15. Consider the function,

```
def f(x, y):
        if x % 2 == 0:
                if y > 0:
                        return "a"
                else:
                        return "b"
        else:
                if y < 0:
                        return "c"
                else:
                        return "d"
```

What is returned from the call `f(3, 5)`?

a) `"a"`

b) `"b"`

c) `"c"`

d) `"d"`

16. Suppose we do,

```
a = [3, 6, 7]
<missing code>
print(a)
```

And this prints,

```
[3, 4, 5, 6, 7]
```

Which of the following can replace `<missing code>` so that this works?

I.
```
a.insert(1, 5)
a.insert(1, 4)
```

II.
```
a.insert(1, 4)
a.insert(1, 5)
```

III.
```
a.append(5)
a.append(4)
a.sort()
```

a) III only

b) I and III

c) II and III

d) I, II, and III

17. What is printed when we run the script,

```
def fun(L):
        total = 0
        for i, x in enumerate(L):
                total += i + 2 * x
        return total

def main():
        a = [10, 20, 30]
        print(fun(a))

main()
```

a) `63`

b) `120`

c) `123`

d) `126`

18. What is printed when we do,

```
a = [10, 20, 30, 40, 50]
b = []
b.append(a.pop(0))
b.append(a.pop(1))
b.append(a.pop(2))
print(b)
```

a) `[30, 20, 10]`

b) `[10, 20, 30]`

c) `[10, 30, 50]`

d) `[50, 30, 10]`

19. What is printed when we do,

```
d = {}                  # Empty dictionary.
L = [2, 3, 4, 3, 2]
M = [17, 18, 19, 20, 21]

for x, y in zip(L, M):
        d[x] = y

print(d[2])
```

a) `2`

b) `17`

c) `21`

d) There is a KeyError.

20. Suppose we do,

```
inventory = {"apples" : 20, "bananas" : 50}
inventory["carrots"] = 80
inventory["apples"] = 40
del inventory["bananas"]
inventory["carrots"] += 10

for x in inventory:
        print(x, ":", inventory[x])
```

Which of the following is a *possible* output of this segment?

a)
```
apples : 20
apples : 40
bananas : 50
carrots : 90
```

b)
```
apples : 20
carrots : 80
bananas : 50
```

c)
```
carrots : 90
apples : 40
```

d)
```
apples : 20
carrots : 80
```

Name ____

Summer packet: AP Computer Science A

Javascript version

For students coming from AP Computer Science Principles

Summer 2019

Directions

- There are two versions of this packet. You only have to do one version, depending on which language you know. This is the Javascript version. You should only do this version if you are familiar with Javascript. If you previously took AP Computer Science Principles, this is the right version for you.
- There are two parts to this packet: multiple-choice and free-response coding.

Part 1: Multiple-choice

- Multiple choice: Please write down your answers (A) – (D) on the answer sheet.
- Be prepared to hand in your answer sheet on the first day of school.

Multiple choice answers

1. ________

2. ________

3. ________

4. ________

5. ________

6. ________

7. ________

8. ________

9. ________

10. ________

11. ________

12. ________

13. ________

14. ________

15. ________

16. ________

17. ________

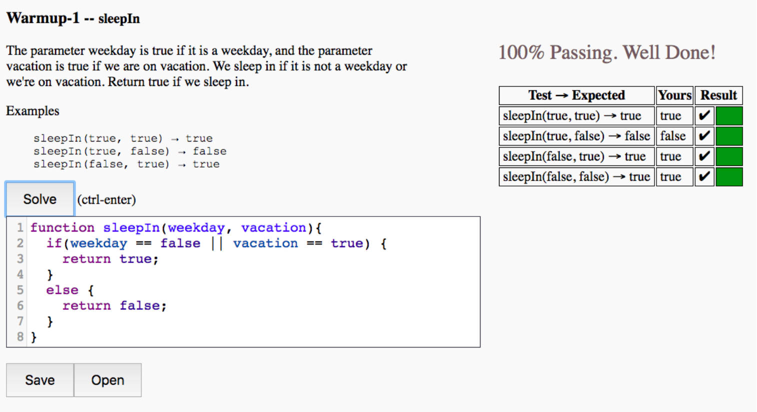
18. ________

19. ________

20. ________

Part 2: Free-response coding

- Go to `https://the-winter.github.io/codingjs/`. For each problem, check your answer by clicking the "Solve" button. If all the tests are "green" you got the problem right. Once you complete a function and verify that it's correct, please take a screenshot that includes your code the tests for correctness. Keep all your screenshots, and email them to mcaulfield@rih.org any time over the summer. Deadline is Friday, September 13, 2019 at the latest.
- Make sure to `return` the answer, not `console.log()` etc.
- For help with syntax, click "JS Syntax Help."
- Example screenshot: Include the directions, your code, and the tests that indicate you got the problem right.



Warmup-1 -- sleepIn

The parameter weekday is true if it is a weekday, and the parameter vacation is true if we are on vacation. We sleep in if it is not a weekday or we're on vacation. Return true if we sleep in.

Examples

```
sleepIn(true, true) → true
sleepIn(true, false) → false
sleepIn(false, true) → true
```

```
function sleepIn(weekday, vacation){
  if(weekday == false || vacation == true) {
    return true;
  }
  else {
    return false;
  }
}
```

100% Passing. Well Done!

Test → Expected	Yours	Result	
sleepIn(true, true) → true	true	✔	🟩
sleepIn(true, false) → false	false	✔	🟩
sleepIn(false, true) → true	true	✔	🟩
sleepIn(false, false) → true	true	✔	🟩

Problems

1. Warmup-1: `monkeyTrouble`
2. Warmup-1: `sumDouble`
3. Warmup-1: `makes10`
4. String-1: `helloName`
5. String-1: `makeTags`
6. Array-1: `firstLast6`
7. Array-1: `plusTwo`
8. Array-1: `swapEnds`
9. Array-1: `sum2`
10. Array-2: `countEvens`

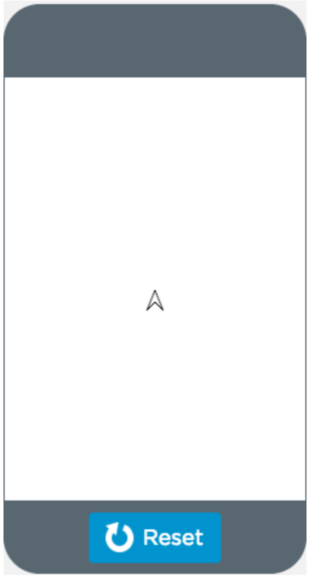
Save all screenshots and send to mcaulfield@rih.org any time over the summer. Deadline is Friday, September 13, 2019 at the latest.

Multiple-choice questions

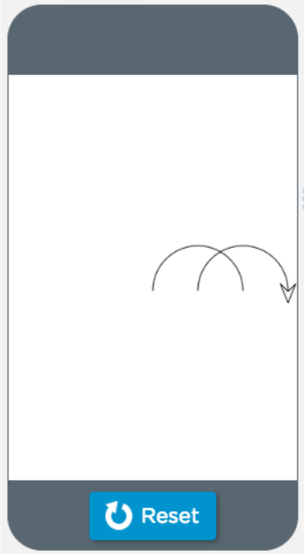
Choose the best answer for each question. Please write all answers on the answer sheet above.

1. Consider the Turtle Graphics drawing:

Start:



Finish:



Which segment will draw the picture on the right?

A)
```
arcRight(180, 50);
penUp();
turnLeft(90);
moveForward(50);
turnRight(90);
penDown();
arcRight(180, 50);
```

B)
```
arcRight(180, 50);
penUp();
turnRight(90);
moveForward(50);
turnRight(90);
penDown();
arcRight(180, 50);
```

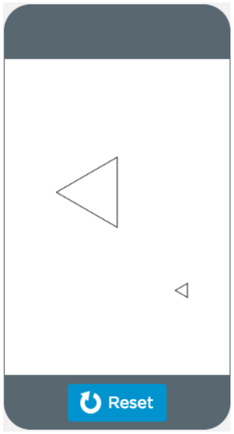
C)
```
arcRight(180, 50);
turnRight(90);
moveForward(50);
turnRight(90);
arcRight(180, 50);
```

D)
```
arcRight(180, 50);
penUp();
moveForward(50);
penDown();
arcRight(180, 50);
```

2. Consider the Turtle Graphics code and the image it draws:

```
drawTriangle(100);
penUp();
move(100, 100);
penDown();
drawTriangle(20);

function drawTriangle(side) {
        for(var i = 0; i < 3; i++) {
                moveForward(side);
                turnLeft(120);
        }
}
```



What is the total of the perimeters of the two triangles?

A) 60 pixels

B) 120 pixels pixels

C) 300 pixels

D) 360 pixels

3. Here are two scripts:

Script I

```
randomDot(randomNumber(0, 255));

function randomDot(num) {
        penRGB(num, num, num);
        dot(50);
}
```

Script II

```
randomDot();

function randomDot(num) {
        penRGB(randomNumber(0, 255),randomNumber(0, 255),randomNumber(0, 255));
        dot(50);
}
```

What is the *best* description of the two scripts?

A) Scripts I and II both draw a dot with a random color. The dot can be any color your screen can display.

B) Scripts I and II both draw a gray dot. The shade of gray can vary from black to white.

C) Script I draws a gray dot with some random shade, and Script II draws a dot with some random color.

D) Script I draws a dot with some random color, and Script II draws a gray dot with some random shade.

4. Consider the Turtle Graphics function:

```
function drawSomething(x, y) {
        penColor(x);
        dot(y);
        moveForward(5 * y);
        dot(y);
}
```

Which of the following calls will run correctly without errors?

A) `drawSomething("blue", 100);`

B) `drawSomething(100, "blue");`

C) `drawSomething(100, 200, 100);`

D) `drawSomething();`

5. Suppose that `X` is a number where `X MOD 5` is `0`, and `42 MOD X` is 2. What is X?

A) 8

B) 15

C) 20

D) 25

6. Which of the following is a true statement about Binary Search?

A) Binary search is an algorithm where you start at the beginning of a series of items and look through them one at a time until you find the item you are looking for.

B) Binary search always works, but it is slower than Linear Search.

C) Binary search does not work if the list is sorted.

D) Binary search works by first going to the middle of a sorted list, and moving left or right depending on if the target value is less than or greater than the value in the middle.

7. In Javascript, the mod operator is `%`. For example, `17 % 5` is 2.

Which of the following scripts will print `"EVEN"` if the user inputs an even number, and `"ODD"` if the user inputs an odd number? Should work for any number.

A)

```
var x = promptNum("Enter a number");
if(x == 2) {
        console.log("EVEN");
}
else {
        console.log("ODD");
}
```

B)

```
var x = promptNum("Enter a number");
if(x % 2 == 0) {
        console.log("EVEN");
}
else {
        console.log("ODD");
}
```

C)

```
var x = promptNum("Enter a number");
if(2 % x == 0) {
        console.log("EVEN");
}
else {
        console.log("ODD");
}
```

D)

```
var x = promptNum("Enter a number");
if(x / 2 == 0) {
        console.log("EVEN");
}
else {
        console.log("ODD");
}
```

8. The largest whole number that can be represented in 4 bits is 1111. What is this number in decimal?

   A) 7

   B) 8

   C) 15

   D) 16

9. What is printed when we run the script:

```
var list = [4, 5, 1, 8, 2];
var ans = fun(list);
console.log("ANS: " + ans);

function fun(a) {
        var m = a[0];
        for(var i = 0; i < a.length; i++) {
                if(a[i] > m) {
                        m = a[i];
                }
                console.log(m);
        }
        return m;
}
```

   A)
```
4
5
1
8
2
ANS: 8
```

   B)
```
4
5
5
8
8
ANS: 8
```

   C)
```
4
5
1
8
2
ANS: 2
```

   D)
```
4
5
1
8
2
ANS: 4
```

10. What is printed when we do,

```
var n = 0;
var x = 1;
var y = 100;
while(x < y) {
        x *= 2;
        y -= 10;
        n++;
}
console.log(n);
```

   A) 5

   B) 6

   C) 40

   D) 64

11. Which of the following prints,

```
5 6
```

A) `console.log(2 + 3 + 2 * 3);`

B) `console.log((2 + 3) + 2 * 3);`

C) `console.log((2 + 3) + " " + 2 * 3);`

D) `console.log(2 + " " + 3 + 2 * 3);`

12. Consider the script,

```
var x = promptNum("Enter a number. ");

if(x < 10 || x > 20) {
        console.log("FIRST");
}
else {
        console.log("SECOND");
}
```

Which of the following code segments is equivalent to this? In other words, which segment will have exactly the same outputs for the same inputs?

A)

```
var x = promptNum("Enter a number");

if(x < 10) {
        console.log("FIRST");
}
else if(x > 20) {
        console.log("FIRST");
}
else {
        console.log("SECOND");
}
```

B)

```
var x = promptNum("Enter a number");

if(x >= 10 || x <= 20) {
        console.log("SECOND");
}
else {
        console.log("FIRST");
}
```

C)

```
var x = promptNum("Enter a number");

if(x >= 20) {
        console.log("FIRST");
}
else if(x <= 10) {
        console.log("FIRST");
}
else {
        console.log("SECOND");
}
```

D)

```
var x = promptNum("Enter a number");

if(x < 10) {
        if(x > 20) {
           console.log("FIRST");
        }
        else {
           console.log("SECOND");
        }
}
else {
        console.log("SECOND");
}
```

13. What is printed when we do,

```
var x  = 0;

f(4);
g(5);
f(4);

console.log(x);

function f(a) {
        x += a;
}

function g(a) {
        x *= a;
}
```

A) `0`

B) `13`

C) `24`

D) `80`

14. Consider the pseudocode algorithm.

```
list = [0, 17, 42, 0, 17]
c = 0
FOR EACH item IN list {
        IF(item == 17) {
                c = c + 2
        }
        IF(item == 42) {
                c = c - 1
        }
}
DISPLAY(c)
```

What is displayed?

(A) `0`

(B) `1`

(C) `2`

(D) `3`

15. What is displayed when we do, in JavaScript,

```
var x = 5;
var y = 3;
var z = x + y;
y = z;
z = x;
console.log(x + " " + y);
```

(A) `5 8`

(B) `8 5`

(C) `8 3`

(D) `5 3`

16. A city uses a software system to calculate tolls for cars entering the city. The rules are:

- If it's rush hour, all cars pay $20 no matter what their weight.
- If it's not rush hour, cars that weigh 1000 or more kilograms pay $20, and cars that weigh less pay $10.

Which of the following pseudocode algorithms correctly represents the algorithm used to calculate tolls?

(A)
```
IF(rushHour OR weight < 1000) {
        DISPLAY(20)
}
ELSE {
        DISPLAY(10)
}
```

(B)
```
IF(rushHour AND weight < 1000) {
        DISPLAY(20)
}
ELSE {
        DISPLAY(10)
}
```

(C)
```
IF(rushHour OR weight >= 1000) {
        DISPLAY(20)
}
ELSE {
        DISPLAY(10)
}
```

(D)
```
IF(rushHour AND weight >= 1000) {
        DISPLAY(20)
}
ELSE {
        DISPLAY(10)
}
```

17. Suppose we have an App Lab program with a single button, `pushButton`. The code for the app is,

```
var x = 0;
onEvent("pushButton", "click") {
        var y = x * 2;
        x = x + 1;
        console.log(y);
}
```

If we click the button twice, what is printed?

(A)
```
0
2
```

(B)
```
1
2
```

(C)
```
0
0
```

(D)
```
0
1
```

18. What is printed when we run the script,

```
var x = doMath(10);
var y = x * 2;
console.log(y);

function doMath(num) {
        var ans = num * 2 + 5;
        return ans;
}
```

A) `40`

B) `45`

C) `50`

D) `55`

19. Suppose we do,

```
var nums = [10, 11, 12];
appendItem(nums, 13);
insertItem(nums, 1, 14);
removeItem(nums, 0);
console.log(nums);
```

What is printed?

A) `[14, 11, 12, 13]`

B) `[10, 14, 12, 13]`

C) `[13, 10, 14, 11, 12]`

D) `[10, 11, 12, 13]`

20. What is printed when we do,

```
nums = [9, 5, 9, 5, 9];

for(var i = 0; i < nums.length; i++) {
        if(nums[i] == 5) {
                console.log(i);
        }
}
```

A)
```
1
3
```

B)
```
5
5
```

C)
```
i
i
```

D)
```
0
1
2
3
4
```