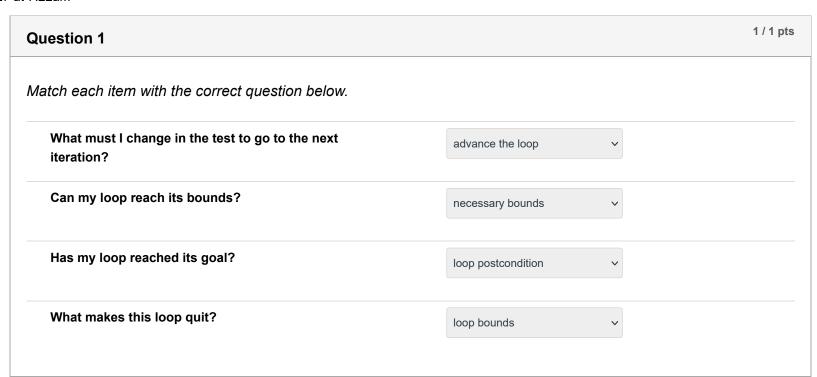
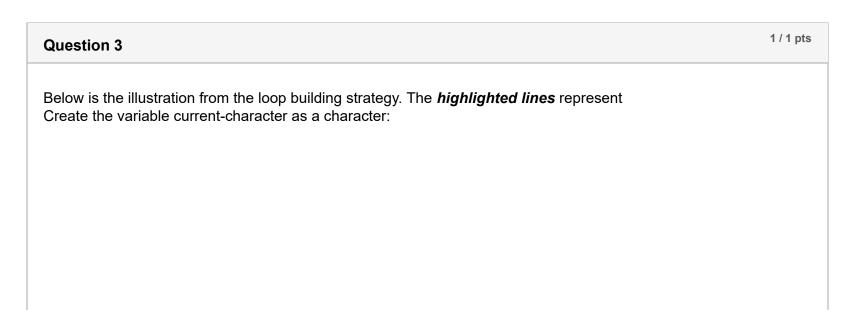
(!) Correct answers are hidden.

Submitted Jun 27 at 7:22am



0 / 1 pts **Question 2** The highlighted section below illustrates: Given: the variable str is a string (may be empty) Create the counter variable, initialized to ${ ext{-}}1$ If the variable str has any characters then Set counter to 0 Create the variable current-character as a character Place the first character in str into current-character While more-characters and current-character not a period Add one to (or increment) the counter variable Store the next character from str in current-character If current-character is a period then Add one to the counter to account for the period. Else Set counter to -2 If counter is -1 the string was empty Else if counter is -2 there was no period None of these a necessary condition a postcondition a boundary condition a loop guard an intentional condition



Incorrect

```
Given: the variable str is a string (may be empty)
Create the counter variable, initialized to -1
If the variable str has any characters then
    Set counter to 0
    Create the variable current-character as a character
    Place the first character in str into current-character
    While more-characters and current-character not a period
       Add one to (or increment) the counter variable
       Store the next character from str in current-character
    If current-character is a period then
       Add one to the counter to account for the period.
      Set counter to -2
If counter is -1 the string was empty
Else if counter is -2 there was no period

    loop postcondition

bounds precondition

    goal operation

advancing the loop

    goal precondition

loop bounds
```

```
0 / 1 pts
Question 4
The highlighted section below illustrates:
    Given: the variable str is a string (may be empty)
    Create the counter variable, initialized to -1
   If the variable str has any characters then {
        Set counter to 0
        Create the variable current-character as a character
        Place the first character in str into current-character
        While more-characters and current-character not a period
           Add one to (or increment) the counter variable
           Store the next character from str in current-character
        If current-character is a period then
          Add one to the counter to account for the period.
          Set counter to -2
    If counter is -1 the string was empty
    Else if counter is -2 there was no period
   a necessary condition
   a postcondition

    an intentional condition

   None of these
   O a loop guard
   a boundary condition
```

Incorrect

6. 7.	<pre>} cout << s << endl;</pre>
0	4
0	None of these
•	5
0	2

ncorrect	Question 6	0 /
	Look at the problem statement below. The of the loop is to count the number of characters in a	
	sentence.	
	How many characters are in a sentence? Count the characters in a string until a period is encountered. If the string contains any characters, then it will contain a period. Count the period as well.	
	plan	
	O None of these	
	O bounds	
	○ goal	

Incorrect

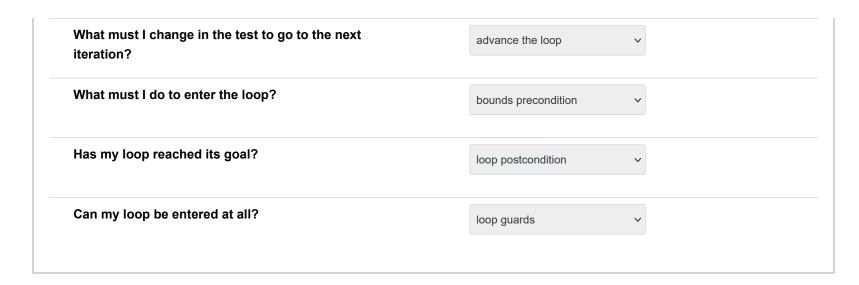
```
0 / 1 pts
Question 7
Below is the illustration from the loop building strategy. The highlighted lines represent:
    Given: the variable str is a string (may be empty)
    Create the counter variable, initialized to {	ext{-}}1
    If the variable str has any characters then
        Set counter to 0
       Create the variable current-character as a character
       Place the first character in str into current-character
       While more-characters and current-character not a period
           Add one to (or increment) the counter variable
           Store the next character from str in current-character
        If current-character is a period then
          Add one to the counter to account for the period.
          Set counter to -2
   }
If counter is -1 the string was empty
    Else if counter is -2 there was no period
   advancing the loop

    goal precondition

   bounds precondition
   goal operation
   O loop bounds
   O loop postcondition
```

```
Question 8

Match each item with the correct question below.
```



```
1 / 1 pts
Question 9
Below is the illustration from the loop building strategy. The highlighted lines represent.
While more-characters and current-character not a period:
    Given: the variable str is a string (may be empty)
   Create the counter variable, initialized to {	ext{-}}1
   If the variable str has any characters then
        Set counter to 0
       Create the variable current-character as a character
       Place the first character in str into current-character
       While more-characters and current-character not a period
           Add one to (or increment) the counter variable
           Store the next character from str in current-character
        If current-character is a period then
          Add one to the counter to account for the period.
        Else
         Set counter to -2
   If counter is -1 the string was empty
    Else if counter is -2 there was no period
   bounds precondition
   O goal precondition
   advancing the loop
   goal operation
   loop bounds

    loop postcondition
```

```
1 / 1 pts
Question 10
Below is the illustration from the loop building strategy. The highlighted lines represent.
Store the next character from str in current-character:
    Given: the variable str is a string (may be empty)
   Create the counter variable, initialized to -1
   If the variable str has any characters then
       Set counter to 0
       Create the variable current-character as a character
       Place the first character in str into current-character
       While more-characters and current-character not a period
           Add one to (or increment) the counter variable
          Store the next character from str in current-character
       If current-character is a period then
          Add one to the counter to account for the period.
         Set counter to -2
    If counter is -1 the string was empty
   Else if counter is -2 there was no period
   advancing the loop
   goal operation
```



O goal precondition			
O bounds precondition			
O loop postcondition			
O loop bounds			

```
1 / 1 pts
Question 11
In H05, here is the pseudocode for the loop body. What line of code needs to appear immediately after the loop body to make the
algorithm complete?
     sum <- 0
     number <- 0
     for each character in str
        Set current character -> ch
        If ch is a digit then
           digit <- ascii-to-decimal(ch)</pre>
           number <- number * 10
           number <- number + digit
           sum <- sum + number
           number <- 0
   sum += number;
   o number = number + digit;

    None of these answers is correct

   number = number + sum;
   o sum = number + digit;
```

```
1 / 1 pts
Question 12
In H05, here is the pseudocode for the loop body. What code would you write to "grab the current character"?
     sum <- 0
     number <- 0
     for each character in str
        Set current character -> ch
        If ch is a digit then
           digit <- ascii-to-decimal(ch)</pre>
           number <- number * 10
number <- number + digit
           sum <- sum + number
           number <- 0
        char ch = str.at(i)
     string ch = str.substr(i, 1);
   O char ch = char[i];

    None of these answers is correct

   char ch; str.at(ch);
```

```
Question 13

In H05, here is code for the loop that is used. What is the underlined portion?

for (size_t i{0}, len{str.size()}; i < len; ++i)
{
}
```



 advancing the loop the bounds precondition the loop postcondition the goal precondition the loop operation 	
the loop postconditionthe goal precondition	
O the goal precondition	
O the loop operation	

```
In H05, here is code for the loop that is used. What is the underlined portion?

for (size_t i{0}, len{str.size()}; i < len; ++i)
{
}

the bounds precondition

the loop operation

the loop bounds

the loop bounds

the loop postcondition

the loop postcondition

the loop postcondition

the loop precondition

the loop postcondition
```

```
Here is an implementation of countCode() from H06. What is its problem?

int countCode(const std::string& str) {
    int result = 0;
    for (size_t i = 0, len = str.size() - 3; i < len; ++i) {
        string subs = str.substr(i, 4);
        if (subs.substr(0, 2) == "co" && subs.back() == 'e') {
            result++;
        }
    }
    return result;
}

It produces the correct output for all input values

It produces incorrect output for strings with a length less than 3

It compiles, but the loop should use len = str.size() - 4

It works correctly, but you should use int for your indexes, not size_t

It does not compile
```

Incorrect Question 16

In H06, your first task is to write **stubs** for each of the functions in the library. Below is an attempt to do that. What is the problem?

int count std::stri	ng zipZap(const std:: Code(const std::string ng everyNth(const std ixAgain(const std::st	ng& str) { } I d::string& str, int		
O you have	not removed the std:: from the	prototypes		
O you are i	nissing the return statement			
you have	removed the semicolons from the	e prototypes		
O There is	nothing wrong with the stubs. The	y are fine		
O the brace	es are formatted incorrectly			

In H06, your first task is to write stubs for each of the functions in the library. Below is an attempt to do that. What is the problem? std::string xipZap(const std::string& str); { return ""; } int countCode(const std::string& str); { return 0; } std::string everyNth(const std::string& str, int n); { return ""; } bool prefixAgain(const std::string& str, int n); { return false; } There is nothing wrong with the stubs. They are fine you have not removed the std:: from the prototypes the braces are formatted incorrectly you have not removed the semicolons from the prototypes the wrong values are being returned

Question 18	1 / 1 pts
In H06, your first task is to write stubs for each of the functions in the library. Below is an attempt to do that. \text{the problem?}	What is
<pre>std::string zipZap(const std::string& str) { return "";} int countCode(const std::string& str) { return 0; } std::string everyNth(const std::string& str, int n) { return ""; } bool prefixAgain(const std::string& str, int n) { return false; }</pre>	
the wrong values are being returned	
o you have removed the semicolons from the prototypes	
There is nothing wrong with the stubs. They are fine	
the braces are formatted incorrectly	
O you have not removed the std:: from the prototypes	