Students:

This content is controlled by your instructor, and is not zyBooks content. Direct questions or concerns about this content to your instructor. If you have any technical issues with the zyLab submission system, use the **Trouble with lab** button at the bottom of the lab.

21.4 Question 4 (5 marks)

In this exercise you will create a class called `LogSanitiser` that processes user log entries by replacing sensitive information. Details below

You are provided with the following files:

- `main.cpp` - starter code to use as you develop the class (non used to evaluate your code)
- `task.h` - read only header file
- `task.cpp` - you will write your C++ code here

---

class String Transformer

Private data member:

```
char sanitizedLog[100];
```

a built-in array to store the user log (up to 100 characters)

Constructor to initialise the `sentence`:

```
LogSanitizer(const char* inputLog);
```

which stores the provided `inputLog` in the `sanitizedLog` array

Public member functions:

```
const char* sanitizeLog();
```

This function does the following:

- replace all numeric characters `0-9` with '!'
- replace vowels (a, e, i, o, u, both uppercase and lowercase) with '*'
- returns the `sanitizedLog`

```
const char* getSanitizedLog() const;
```

- this is a `getter` to return the `sanitizedLog`

Example For the following:

```
LogSanitizer sanitizer("User ID: ABC123, Code: AEIOU");
sanitizer.sanitizeLog();
std::cout << sanitizer.getSanitizedLog() << std::endl;
```

the output should be:

```
"*s*r *D: *BC###, C*d*: *****"
```

NOTE: all separating spaces are unchanged from the input string

671842.4329690.qx3zqy7

LAB ACTIVITY 21.4.1: Question 4 (5 marks) 0 / 2

Current file: main.cpp Load default template...

```
1  #include <iostream>
2  #include "task.h"
3
4  int main() {
5     LogSanitizer sanitizer("User ID: ABC123, Code: AEIOU");
6     sanitizer.sanitizeLog();
7     std::cout << sanitizer.getSanitizedLog() << std::endl;
8
9     return EXIT_SUCCESS;
10 }
11
```

Develop mode | Submit mode

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above) → **main.cpp** (Your program) →

Program output displayed here

Coding trail of your work What is this?

```
History of your effort will appear here once you begin
working on this zyLab.
```

Trouble with lab?