

CISS350: Data Structures and Advanced Algorithms
Quiz q02

Name: YOUR EMAILScore:

Q1. Write a function `replace()` that on calling `replace(x, y, z)`, replaces every occurrence of string `y` in string `x` with string `z`. For instance the output of

```
char x[1024] = "hello world hello missouri hello columbia";
char y[1024] = "hello";
char z[1024] = "goodbye";
replace(x, y, z);
std::cout << x << '\n';
```

is

```
goodbye world goodbye missouri goodbye columbia
```

Note that if `x` is "aaa", `y` is "aa", and `z` is "A", then after called `replace()`, `x` is "Aa". You must *not* use any string function/method provided by C/C++.

ANSWER:

```
#include <iostream>

void replace(char * x, const char * y, const char * z)
{
}

int main()
{
    char x[1024], y[1024], z[1024];
    std::cin >> x >> y >> z;
    replace(x, y, z);
    std::cout << '[' << x << "]\n";
    return 0;
}
```

(Hint: For each `i`, you want to check if starting with `x[i]` you can see the characters of `y`, i.e., `x[i]` is `y[0]`, `x[i+1]` is `y[1]`, etc. This part finds the `y` inside your `x`. If so, then you do a replacement of `y` inside `x` with `z`, starting at `i`. You want to consider two cases: when length of `y` is \geq the length of `z` and when length of `y` is $<$ the length of `z`. It's probably a good idea to write several helper functions.)

INSTRUCTIONS

In the file `thispreamble.tex` look for

```
\renewcommand\AUTHOR{}
```

and enter your email address:

```
\renewcommand\AUTHOR{jdoe5@cougars.ccis.edu}
```

(This is not really necessary since alex will change that for you when you execute `make`.) In your bash shell, execute “`make`” to recompile `main.pdf`. Execute “`make v`” to view `main.pdf`.

Enter your answers in `main.tex`. In the bash shell, execute “`make`” to recompile `main.pdf`. Execute “`make v`” to view `main.pdf`.

For each question, you’ll see boxes for you to fill. For small boxes, if you see

```
1 + 1 = \answerbox{}
```

you do this:

```
1 + 1 = \answerbox{2}
```

`answerbox` will also appear in “true/false” and “multiple-choice” questions.

For longer answers that need typewriter font, if you see

```
Write a C++ statement that declares an integer variable name x.
\begin{answercode}
\end{answercode}
```

you do this:

```
Write a C++ statement that declares an integer variable name x.
\begin{answercode}
int x;
\end{answercode}
```

`answercode` will appear in questions asking for code, algorithm, and program output. In this case, indentation and spacing is significant. For program output, I do look at spaces and newlines.

For long answers (not in typewriter font) if you see

```
What is the color of the sky?
\begin{answerlong}
\end{answerlong}
```

you can write

```
What is the color of the sky?  
\begin{answerlong}  
The color of the sky is blue.  
\end{answerlong}
```

A question that begins with “T or F or M” requires you to identify whether it is true or false, or meaningless. “Meaningless” means something’s wrong with the question and it is not well-defined. Something like “ $1 + 2 = 4$ ” is either true or false (of course it’s false). Something like “ $1+2 = 4?$ ” does not make sense.

When writing results of computations, make sure it’s simplified. For instance write 2 instead of $1 + 1$.

HIGHER LEVEL CLASSES.

For students beyond 245: You can put L^AT_EX commands in `answerlong`.

More examples of meaningless statements: Questions such as “Is $42 = 1+2$ true or false?” or “Is $42 = \{2\}^{\{3\}}$ true or false?” does not make sense. “Is $P(42) = \{42\}$ true or false?” is meaningless because $P(X)$ is only defined if X is a set. For “Is $1 + 2 + 3$ true or false?”, “ $1 + 2 + 3$ ” is well-defined but as a “numerical expression”, not as a “proposition”, i.e., it cannot be true or false. Therefore “Is $1 + 2 + 3$ true or false?” is also not a well-defined question.

More examples of simplification: When you write down sets, if the answer is $\{1\}$, do not write $\{1, 1\}$. And when the values can be ordered, write the elements of the set in ascending order. When writing polynomials, begin with the highest degree term.

When writing a counterexample, always write the simplest.