

CISS350: Data Structures and Advanced Algorithms
Quiz q10603

Name: YOUR EMAILScore:

For each of the following questions, evaluate the expression written in either prefix/Polish notation or postfix/Reverse Polish notation, or write ERROR if the expression is invalid. You only need to write down the answer. You need not explain.

Q1. Polish notation: $* + 3 5 - 7 2$

ANSWER:

Q2. Polish notation: $+ / 8 2 - 4 * 7 2$

ANSWER:

Q3. Polish notation: $* 8 * 2 + 3 - 2 / 7 2$

ANSWER:

Q4. Polish notation: $/ + - * 2 3 7 4 2$

ANSWER:

Q5. Reverse Polish notation: $2 3 4 * 5 * +$

ANSWER:

Q6. Reverse Polish notation: $5 7 2 + 6 -$

ANSWER:

Q7. Complete the following function that returns **true** if the integer parameter **n** is a palindrome. An integer is a palindrome if it's the same if the read it left-to-right or right-to-left. For instance 0, 12321, and 123321 are palindromes. Reading 12321 left-to-right, you see 1, 2, 3, 2, 1, which is the same if you read it right-to-left.

But -1 and 1232 are not palindromes. Note that reading -1 left-to-right gives you -, 1 and reading it right-to-left gives you 1, -.

ANSWER:

```
bool is_palindrome(int n)
{
}
```

(HINT: You can use a stack. But you don't have to. There are basically two ways to solve this problem, you either look at the digits of n , or you form the reverse of n with the digits. There are variations of the above ideas.)

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.