CISS350: Data Structures and Advanced Algorithms
Quiz q10604

Name: YOUR EMAIL Score:

Q1. Infix to Polish notation:
Convert the following in infix notation `1 + (2 + 3) * 4 + 5` to Polish notation. If what is given has an error, write ERROR. Note that you are converting the expression, you are not evaluating the expression.
Answer:

Q2. Polish to infix notation:
Convert the following in Polish notation `+ / 8 2 - 4 * 7 2` to infix notation. If what is given has an error, write ERROR. Note that you are converting the expression, you are not evaluating the expression.
Answer:

Q3. Infix to Reverse Polish notation:
Convert the following in infix notation `1 / (2 + 3 * 4) * 5` to Reverse Polish notation. If what is given has an error, write ERROR. Note that you are converting the expression, you are not evaluating the expression.
Answer:

Q4. Reverse Polish to infix notation:
Convert the following in Reverse Polish notation `2 3 4 * 5 * +` to infix notation. If what is given has an error, write ERROR. Note that you are converting the expression, you are not evaluating the expression.
Answer:

Q5. Using five `1`s and as many `+`s as you like, write a mathematical expression in Reverse Polish notation that will have the largest possible stack during the evaluation of the expression.
Answer:

Q6. Using five `1`s and as many `+` as you like, write a mathematical expression in Reverse Polish notation that will have the smallest possible stack during the evaluation of the expression.

Answer:

Q7. Complete the following function that returns `true` if the vector of numeric and operator strings is a valid RPN expression. For instance `{"1", "2", "+"}` is valid and `{"1", "+", "2"}` is not valid. Note that you are not evaluating the expression. The following functions are provided:

```
bool is_numeric(const std::string & s);
bool is_operator(const std::string & s);
```

Answer:

```
#include <iostream>
#include <string>
#include <vector>
#include <stack>

// Returns true if string is numeric.
// Assume s.size() > 0 and no sign ('+' or '-')
bool is_numeric(const std::string & s)
{
    if (s.size() == 0) return false;
    for (unsigned int i = 0; i < s.size(); ++i)
    {
        if (s[i] < '0' || '9' < s[i])
        {
            return false;
        }
    }
    return true;
}

// Returns true if string is an operator.
// Assume operator is +, -, *, /, %
bool is_operator(const std::string & s)
{
    if (s.size() != 1) return false;
    return (s[0] == '+' || s[0] == '-' || s[0] == '*' || s[0] == '/' ||
            s[0] == '%');
}
```

```
bool is_rpn(const std::vector< std::string > & v)
{
    for (unsigned int i = 0; i < v.size(); ++i)
    {
        std::string t = v[i];
    }
    return true;
}

int main()
{
    std::vector< std::string > v {"1", "2", "+"};
    std::cout << is_rpn(v) << '\n';
    return 0;
}
```

(Hint: Do you really need a stack?)

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 LaTeX 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.