CISS350: Data Structures and Advanced Algorithms
Quiz q10602

Name: YOUR EMAIL Score: □

Q1. Write a function `f` that sorts `v` (a `std::vector< int >` object) by the lowest order digit in the following way. (The lower order digit of `42` is `2`.) Create an array `x` of 10 `std::list` objects. `x[0]` will be a list of integers with lower order digit `0`. `x[1]` will be a list of integers with lower order digit `1`. Etc. If the integer you read from `v` is `42`, you perform insert tail on `x[2]` with `42`. If the integer you read from `v` is `705`, you perform insert tail on `x[5]` with `705`. For instance if `42, 62, 32` are the only integers in `v` with lowest order digit of `2`, then `x[2]` will be the linked list `42, 62, 32` where `42` is the head. After all values of `v` are placed in `x`, you copy the values from `x` back into `v` from index 0 to index 9, and for each `x[i]`, you process the values from head to tail.

(This is essentially a radix sort on the lowest order digit. If you repeat the above process for the 10s digit, for the 100s digit, etc., you'll get the radix sort using linked lists.)

A skeleton code is given. Here's a test case:

```
52 31 75 73 51 -1
v: [52, 31, 75, 73, 51]
[]
[31, 51]
[52]
[73]
[]
[75]
[]
[]
[]
[]
v: [31, 51, 52, 73, 75]
```

Answer:

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

```
std::ostream & operator<<(std::ostream & cout, const std::vector< int > & v)
{
    std::string delim = "";
    cout << "[";
    for (auto x: v)
    {
        cout << delim << x;
        delim = ", ";
    }
    cout << "]";
    return cout;
}

std::ostream & operator<<(std::ostream & cout, const std::list< int > & v)
{
    std::string delim = "";
    cout << "[";
    for (auto x: v)
    {
        cout << delim << x;
        delim = ", ";
    }
    cout << "]";
    return cout;
}

void f(std::vector< int > & v)
{
    std::vector< std::list< int > > lists(10);

    // TODO: Add values in v to lists

    for (auto list: lists)
    {
        std::cout << list << '\n';
    }

    // TODO: Copy values in lists to v
}

int main()
{
    std::vector< int > v;
    while (1)
    {
        int t;
        std::cin >> t;
        if (t == -1) break;
        v.push_back(t);
    }
    std::cout << "v: " << v << '\n';
    f(v);
    std::cout << "v: " << v << '\n';

    return 0;
}
```

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.