CISS240: Introduction to Programming
Quiz q0305

Name: sithurman1@cougars.ccis.edu Score:

This is a closed-book, no compiler, 5 minute quiz.

Q1. The following code fragment has a repeating chunk of code that repeats 3 times:

```
int i = 1, j = 1;

j = j * i;
i = i + 1;

j = j * i;
i = i + 1;

j = j * i;
i = i + 1;
```

If the goal is to compute $1 \times 2 \times 3 \times \cdots \times 8$, how many times does the repeating chunk of code appear?
Answer:

```
8
```

Q2. What is the final value of `i` at the end of this code fragment?

```
int i = 0, j = 1;

i = i * 10 + j;
j = j + 1;

i = i * 10 + j;
j = j + 1;

i = i * 10 + j;
j = j + 1;

i = i * 10 + j;
j = j + 1;

i = i * 10 + j;
j = j + 1;
```

```
i = i * 10 + j;
j = j + 1;
```

Answer:

```
123456
```

Q3. What is the final value of `k` at the end of this code fragment?

```
int i = 135692468, j = 1000000, k;

k = i / j % 10;
j = j / 10;

k = i / j % 10;
j = j / 10;

k = i / j % 10;
j = j / 10;

k = i / j % 10;
j = j / 10;

k = i / j % 10;
j = j / 10;

k = i / j % 10;
j = j / 10;
```

Answer:

```
6
```

Q4. What is the final value of `k` at the end of this code fragment?

```
int i = 135792468, j = 1, k = 0;

k = k * 10 + i / j % 10;
j = j * 10;

k = k * 10 + i / j % 10;
j = j * 10;

k = k * 10 + i / j % 10;
j = j * 10;

k = k * 10 + i / j % 10;
j = j * 10;
```

```
k = k * 10 + i / j % 10;
j = j * 10;

k = k * 10 + i / j % 10;
j = j * 10;
```

Answer:

```
864297
```

Q5. What is the final value of `k` at the end of this code fragment?

```
int i = 135792468, j = 1, k = 0;

k = k + i / j % 10;
j = j * 10;

k = k + i / j % 10;
j = j * 10;

k = k + i / j % 10;
j = j * 10;

k = k + i / j % 10;
j = j * 10;

k = k + i / j % 10;
j = j * 10;

k = k + i / j % 10;
j = j * 10;
```

Answer:

```
36
```

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.