CSC 209 Review 6 Solution

August 24, 2020

1 Exercises

1. I need to write which of the supplied function calls don't work and explain why.

   - `b)` String format in `printf` expects character constant, but string literal is used
   - `c)` String format in `printf` expects string but character constrant is used
   - `e)` The first argument in `printf` expects pointer but character constrant (an integer) is used isntead
   - `h)` The first argument in `putchar` expects a character, but string literal (a pointer to character) is used
   - `i)` The first argument in `puts` expects a pointer to character, but character constant (an integer) is used

   Notes

   - **putchar**
     - **Syntax:** `int putchar(int char)`
     - Writes a character (an unsigned char) specified by the argument char to stdout.
     - Does not append a new line to the output
     - Is similar to printf but for character
   - **puts**
     - **Syntax:** `int puts(const char *str)`
     - Writes a string to stdout up to but not including the null character
     - Appends a newline character to the output.
     - Is similar to printf but for string
   - **Character Constant**
     - **Syntax:** `' ... '`

  - Is represented by an <u>integer</u>
- **String Literal**
  - **Syntax:** `" ... "`
  - Has a sequence of characters inside
  - Ends with `\0`
  - Is represented by a <u>pointer</u>

    <u>Example</u>

    "When you come to a fork in the road, take it"
- **Escape Squences in String Literal**
  - A common example is '`\n`'
    - causes the cursor to advance to the next line

2. First, I need to write which of the provided function calls are legal, and write the output produced

   The solution to the first part is:

   - `b)` [output: a]
   - `c)` [output: abc]

   Second, I need to write which of the following function calls are illegal, and explain why.

   The solution to the second part is:

   - `a) purchar` expects a character constant (an integer) but a value of type pointer to `char` is used
   - `d) puts` expects a variable of type pointer to `char`, but a variable of type pointer to `char` is used

3. I need to write the values of `i`, `j`, `k` in the function

   ```
   scanf("%d%s%d", &i, s, &j)
   ```

   if the user enters `12abc34 56def78`.

   The solution to this problem is:

   - `i`
   - `j`
   - `k`