Lab 13 – Bitwise Ops

Bitwise ops, strtok, double pointers, review

strtok (ex from cplusplus.com)

Example

```c
/* strtok example */
#include <stdio.h>
#include <string.h>

int main ()
{
  char str[] ="- This, a sample string.";
  char * pch;
  printf ("Splitting string \"%s\" into tokens:\n",str);
  pch = strtok (str," ,.-");
  while (pch != NULL)
  {
    printf ("%s\n",pch);
    pch = strtok (NULL, " ,.-");
  }
  return 0;
}
```

Output:

```
Splitting string "- This, a sample string." into tokens:
This
a
sample
string
```

How strtok works (not thread safe)

- It accepts a string (Param 1) and a delimiter (Param 2). Both are `char *`
- Once the string is passed in the first round, we pass in NULL for future calls.
- When we pass in null, it will use the last string it used
  - Param 1 is a static variable within strtok.
- Param 2 should consist of the delims we want to look for. Note that if any one of the argued symbols are found, it will split.
- strtok will return the string upto the symbol or the end of the string. If we reached the end previously, then NULL is returned and we should stop.
- For example, strtok(“hi no dash”, “-”); will return “hi no dash” since “-” was not found.

Using Double Pointers

```
159  void foo(int ** headPtr, int* newHead) {
160      // No different that setting a new value for int *.
161
162      *headPtr = newHead;
163  }
164
164
```

Bitwise ops

- | (BITWISE OR)
  - `0 | 0 = 0` ||| `1 | 0 = 1`
  - `0 | 1 = 1` ||| `1 | 1 = 1`
- & (BITWISE AND)
  - `0 & 0 = 0` ||| `1 & 0 = 0`
  - `0 & 1 = 0` ||| `1 & 1 = 1`
- ^ (BITWISE XOR)
  - `0 ^ 0 = 0` ||| `1 ^ 0 = 1`
  - `0 ^ 1 = 1` ||| `1 ^ 1 = 0`
- << (Arithmetic shift left)
  - `0000 1010 << 2 = 0010 1000` === `10 << 2 = 40`
- >> (Arithmetic shift right)
  - `0011 0100 >> 2 = 0000 1101` === `52 >> 2 = 13`

```
(0101 1010 1110 ^ 0101 1110 1010)&
(1011 0000 1111 | 0100 0101 0000)

      (0101 1010 1110
     ^ 0101 1110 1010)
    & (1011 0000 1111
     | 0100 0101 0000)
       ---------------
       ???? ???? ????
```

Questions?

Sample Questions:
https://forms.office.com/r/GKWbRri5K4

Exam Review

Understanding Levels of Indirection

3

Consider the following code.
What are the values of x and y after the call to foo?
(1 Point)

```
29 void f2(void) {
30     int x = 0; // Addr = 0x0020
31     int y = 124; // Addr = 0x0024
32
33     foo(&x, &y);
34 }
35
36 void foo(int * this, int * that) {
37     if (this > that) {
38         *this = that;
39     } else {
40         *that = this;
41     }
```

- ○ x = 0; y = 0
- ○ x = 0x0024; y = 124
- ○ x = 0x0020; y = 0x0024
- ○ x = 0x0020; y = 0x0020
- ○ x = 0; y = 0x0020
- ○ x = 0; y = 124
- ○ x = 124; y = 0
- ○ x = 0; y = 124
- ○ (We cannot tell from this info)
- ○ x = 0; y = 0x0024

16

In C, a variable name MAY start with the following:
(1 Point)

- [ ] An ampersand (&)
- [ ] *Alowercasecharacter*
- [ ] *Astar*(∗)
- [ ] *Anon* − *ASCIIcharacter*
- [ ] *Aspecialcharacter*(. *etc.*)
- [ ] *Adigit*
- [ ] *Anuppercasecharacter*

27

Select the most correct function call when passing an array arr.
(1 Point)

- [x] foo(arr)
- [ ] foo(&arr[0])
- [ ] foo(&arr)
- [ ] foo(arr[0])

32

Which bitwise operators can we use to clear individual bits?
(1 Point)

We only have one variable (we cannot use another tmp). Clearing means setting the bit to 0.

- [ ] <<
- [ ] >>
- [ ] |
- [ ] &
- [ ] ^

36

What is a data structure?
(1 Point)

Answer assuming we are using C, C++, Swift, Java, Haskell, or **any** other language. Hint: There is only one answer globally accepted.

- ○ A class
- ○ A way of organizing memory in a computer
- ○ A structure
- ○ An object of any kind
- ○ A union

On the exam, this will be one question. **Write DNE if no such solution exists.**

```
12 void testFunc() {
13      int x = 100; // ADDR = 0x00
14      int y = 240; // ADDR = 0x04
15      int * xPtr = &x; // ADDR = 0x08
16      int * yPtr = &y; // ADDR = 0xF0
17 }
```

54

What is the value of xPtr?
(1 Point)

Enter your answer

55

What is the value of yPtr
(1 Point)

Enter your answer

Be sure to look over the 3rd section of the quiz. It will help with understanding some of the harder problems. Note: The questions related to the given code may differ, but the overall idea is the same.

Lab Final

- Review Lab 11
- 3 hours (normal time)
  - If the previous lab is still in here, it will not cut your time
  - If majority of people are working, you will get a *few* extra minutes
- No notes, labs, or internet; **ONLY the book! Digital or paper version ok!**
  - The book can be time consuming! Do not rely on it!
- No phones (I will allow music, but you cannot change it once we start)!
- Project submission will be via USB Drive (USB A and C). Hand back the paper.
  - **PUT YOUR NAME IN YOUR PROJECT AND YOUR PAPER! NO NAME = NO GRADE!**
- Failure of any steps = 0. The rules will be laid out on the handout

0 on Lab final = automatic fail, possibly with a report

Due Dates

- PA 8 due Friday December 8th
- Final grades are due around the 18th (I cannot recall the exact date)
- All submissions or disagreements with grades are not accepted after the 8th.
- You may reach out about PA 8 grades after they are posted, but do not delay!