CS 354: Intro to Computer Systems (Spring 2019 @ Epic)

In-class Worksheet 2

1. The code snippet below tries to create a duplicate of a given string. Can you identify the bug in this code and correct it? You may assume that the functions `strlen()`, `strcpy()` work as expected and the calling function (e.g., `main()`) frees the allocated memory. Also assume that the pointer `s` points to a valid string and is NOT NULL. `[3 points]`

```
char* strdup(char *s) {
    char *d = malloc(strlen(s));
    if (d == NULL)
        return NULL;
    strcpy(d,s);
    return d;
}
```

BUG:

FIX:

2. What is the output of the following code snippet? Assume **malloc** is successful and the contents in the malloc'ed memory are all initialized to **0 (zero)**.
`[5 points]`

```
#include <stdio.h>
#include <stdlib.h>

#define ARRAY_LENGTH(A) (sizeof(A) / sizeof(A[0]))

int main(int argc, char *argv[]) {
    int *p = malloc(sizeof(int) * 5);
    for (i = 0; i < ARRAY_LENGTH(p); ++i) {
        printf("p[%d] = %d\n", i, p[i]);
    }
    int a[] = {1, 2, 3, 4, 5};
    int i;
    for (i = 0; i < ARRAY_LENGTH(a); ++i) {
        printf("a[%d] = %d\n", i, a[i]);
    }
    return 0;
}
```

OUTPUT:

"There are only 10 types of people in the world: those who understand binary, and those who don't."

3. Integer Representations

   Assume that the size of an **integer** in a machine is **1 byte (8 bits)**. Answer the following questions with respect to this machine.

   (a) What is the maximum value of an unsigned integer (in decimal)?

   (b) What is the maximum value of a signed integer (in decimal)?

   (c) What is the minimum value of a signed integer (in decimal)?

   (d) If 8 (eight) bits are used to represent an address on this machine, how many unique addresses are there?

   (e) What is the value of the bit pattern 1010 1010 when it is interpreted as:

      i. a hexadecimal integer:

      ii. an unsigned decimal integer:

      iii. a signed decimal integer:

   (f) What is the binary representation of the following integer values (given in decimal) on this machine?

      i. -1 (minus one)

      ii. 42

      iii. -128

"Low-level programming is good for the programmer's soul."

4. Bitwise Operations

Consider the following code snippet. Answer the following questions based on this code snippet. You should also assume that an **integer (signed and unsigned)** takes **1 byte** of memory on this machine.

```
int snum = -1;
unsigned int unum = 1;
```

(a) What are the values of the following expressions? In other words what is the value that will be printed if these expressions are used in a print statement. e.g., `printf("0x%x", snum & unum);` You may assume that the right shift on a **signed** integer will be an **arithmethic** right shift and that on an **unsigned** integer will be a **logical** right shift. Write all values in hexadecimal.

Expression	Value
`snum & unum`	0x
`snum ^ unum`	0x
`snum && unum`	0x
`(snum « 7) | unum`	0x
`(snum « 7) » 7`	0x
`(unum « 7) » 7`	0x

(b) What is the output of the following print statement?

```
printf("%d\t%d\t%d\t", nameMe(1), nameMe(0), nameMe(-1));
```

Assuming that the valid inputs to this function will only be in the range `[-127, 127]`, can you give this function a suitable name based on what it is doing?

```
int nameMe(int x) {
    int y = ~x + 1;
    return y;
}
```

OUTPUT:

Suitable **name** for this function:

5. More low-level programming! How low are we going to go?!

Assume a **little-endian** machine in which the size of a **char** and **short int** are **1 byte** and **2 bytes** respectively. An **int** and a **pointer** take **4 bytes** of storage.

Consider the following code snippet. Assume that the variable **num** is allocated starting from memory address **0x3000**.

```
int num = 0x0FF1CE;
int *pnum = &num;
char *pchr = (char *)pnum;
```

(a) What are the values of the following expressions? In other words what is the value that will be printed if these expressions are used in a print statement. e.g., `printf("0x%x", *pnum);` Write all values in hexadecimal. `[7 points]`

Expression	Value
`*pnum`	0x
`(short)num`	0x
`pchr[1]`	0x
`(char)num`	0x
`++(*pchr)`	0x
`++pchr`	0x
`++pnum`	0x

(b) The function `is_str_longer()` compares the length of two strings `s` and `t` and should return:

i. 1 (one), if strlen(s) > strlen(t)

ii. 0 (zero), otherwise

The function signature of strlen is: `unsigned int strlen(char *s);` Will the function `is_str_longer()` work as expected? If yes, just write "WORKS". If not, mention the issue and provide a fix for the same. You may assume that the pointers `s` and `t` point to valid strings and are NOT NULL.

```
int is_str_longer(char *s, char *t) {
    if (strlen(s) - strlen(t) > 0)
        return 1;
    else
        return 0;
}
```

ANSWER: