CS61, Fall 2011
Assignment 1: Representation of information, and C self assessment

Assigned: Tuesday, September 6

Due: **Tuesday, September 13, 11:59pm**

Question One Due: **Thursday, September 8, 5:00pm**

Instructions

- Do Question 1 as soon as possible. It will take you less than five minutes to fill in the form.
- Hand in your answers to Questions 2–4 by 11:59pm on Tuesday, September 13. You may either hand in a hard copy at the lecture or email an electronic version to `cs61-staff@seas.harvard.edu`. Make sure that your name is clearly written on every page.
- You do not need to submit your answers for Question 5, but you are welcome to discuss your answers with course staff.

Questions

1. **Student information** (5 points)

   Please go to `http://tinyurl.com/cs61-fall-form` and fill in the requested information. (This form is linked to from the CS 61 home page `http://cs61.seas.harvard.edu`.)

   We will use this information to create an account for you on the CS 61 infrastructure. **You must complete this form by 5pm on Thursday, September 8.**

2. **Hexadecimal notation** (25 points)

   Fill in the following table by performing the appropriate number conversions.

Decimal	Binary	Hexadecimal
16		
32		
64		
128		
4,096		
0		
135		
	`0001 1101`	
	`1100 0101`	
		`0x42`
		`0x8A`
6,796,017		`0x67B2F1`
23,035	`0101 1001 1111 1011`	

3. **Two's complement encoding** (15 points)

   Assume that we are using 4 bits to represent integers using two's complement encoding. Fill in the following table.

Decimal	Binary	Hexadecimal
-1		
6		
-6		
		`0x7`
	`1000`	
		`0xC`
		`0xD`
		`0xE`

4. **Base -2 encoding** (25 points)

   In class we saw several ways of encoding negative integers into bits. Another possible encoding is to express the numbers in base -2.

   Representing numbers using a negative base works exactly the same as for a positive base: the kth column from the right represents multiples of b^{k-1}, where b is the base. For example, `1011` is the base -2 representation the number -9, since

$$\begin{aligned} & 1 \times (-2)^3 \quad + \quad 0 \times (-2)^2 \quad + \quad 1 \times (-2)^1 \quad + \quad 1 \times (-2)^0 \\ = \quad & -8 + 0 + (-2) + 1 \\ = \quad & -9. \end{aligned}$$

(a) Write the base -2 representation for all integers from -8 to 8 inclusive.

(b) Suppose we use $2n$ bits to represent integers using base -2 encoding, for $n > 0$. What is the largest integer we can represent? What is the smallest?

(c) Do you think that base -2 encoding is a good representation for computers to use? Explain your answer.

5. **C self assessment** (0 points)

This course requires programming in C. This question is intended to help you understand whether you have sufficient knowledge of the C programming language to be comfortable with the programming assignments that occur later in the course.

You do not need to submit your answers for this question. However, you are welcome to meet with course staff to discuss any difficulties you have with any of the questions.

If you can answer all of the questions confidently, then you are likely to be fine with the C programming aspects of this course. If you are unsure of the answers to some of the questions, don't panic: you may be alright in the course provided you spend time in the first few weeks learning C concepts, and practicing C programming. Feel free to contact course staff if you are unsure whether you are sufficiently prepared.

(a) What is the output of the following program?[†]

```
#include <stdlib.h>
#include <stdio.h>
#define print_int(str) printf("%s : %d\n",#str,(str))
int main() {
    int y = 100;
    int *p = malloc(sizeof(int));
    *p = 10;
    y = y / (*p);
    print_int(y);
    return 0;
}
```

(b) The following program is intended to read and print a number, but it doesn't. What is/are the mistakes?[†]

```
#include <stdio.h>
int main() {
    int n;
    printf("Enter a number:\n");
    scanf("%d\n",n);

    printf("You entered %d \n",n);
    return 0;
}
```

[†]Question adapted from *C Puzzles* (`http://www.gowrikumar.com/c/`) by Gowri Kumar.

(c) Implement the function `int strncopy(char *trg, char *src, int n)`, which copies at most `n` characters from the string `src` to `trg`. That is, if `src` is a string of length m, then the fuction copies $\min(m+1, \mathtt{n})$ characters. The function returns the number of characters copied.

Make sure you compile your code, and try running it on some test cases.

(d) What is the output of the following program?

```
#include<stdio.h>
struct foo { int x; struct foo *p; };

struct foo makeFoo(int x, struct foo *p) {
    struct foo temp;
    temp.x = x;
    temp.p = p;
    return temp;
}

int main() {
    struct foo a = makeFoo(3, NULL);
    struct foo b = makeFoo(5, &a);
    struct foo c = makeFoo(7, &b);
    struct foo *d = c.p->p;

    printf("%d %d %d \n", b.x, (*(c.p)).x, d->x);
    return 0;
}
```

(e) Suppose we have a doubly linked list data structure, where nodes in the list are represented with the following structure.

```
struct node {
    int data;
    struct node *next;
    struct node *prev;
}
```

The `prev` field of the first element of the list contains the value `NULL`, as does the `next` field of the last element.

Implement the function `struct node *reverse(struct node *head)`. The argument `head` points to the first element of the list. The function should reverse the list, and return a pointer to the first element of the reversed list. Your implementation should not allocate any new structures.