CSE31 HW 1

This assignment checks your understanding of C using pointers and structs with review of number representation. You can fill in this document directly for your submission.

Problem 1

a. Given the 8-bit binary integers below, fill in the corresponding base 10 values according to the listed representations:

Binary	Unsigned	Signed	1’s Complement	2’sComplement	Biased
1100 1010					
0011 1001					
0110 1010					
1001 0000					

b. Fill T/F in the following table:

Property	Unsigned	Signed	1’s Comp	2’s Comp	Biased
Can represent positive numbers					
Can represent negative numbers					
Has more than one representation for 0					
Use the same addition process as unsigned					

c. What is the value in decimal of the most negative 16-bit 2’s complement integer?

d. What is the value in decimal of the most positive 16-bit signed integer?

Problem 2

Write a C function named **`swapArray`** that, given two integer arrays of size "*n*", swap the content of these arrays. For example, the program segment

```
int main (int argc, char **argv) {
  int *arr1, *arr2;
... // Assume some code here to fill-in both arrays
  swapArray(arr1, arr2, n);
... // Assume some code here to print both arrays
}
```

would print the following output if arr1 contains [10 20 30 40 50 60 70 80 90 100] and arr2 contains [0 9 8 7 6 5 4 3 2 1]:

```
arr1 after swapping: 0 9 8 7 6 5 4 3 2 1
arr2 after swapping:  10 20 30 40 50 60 70 80 90 100
```

Note: you only need to implement the **`swapArray`** function, no need to worry about how the main program does the input and output.

```
void swapArray( int* a1, int* a2, int size){

}
```

Problem 3

a. The following function should allocate space for a new string, copy the string from the passed argument into the new string, and convert every upper-case character in the **new** string into a lower-case character (do not modify the original string). Fill-in the blanks and the body of the ***for() loop***:

```
char* changeCase(char* str) {
     char* p;
     char* result;
     result = (char*) malloc(_____________________________);

     strcpy(______________, _______________);

     for( p=result; *p!='\0'; p++ ) {
     /* Fill-in 'A' = 65, 'a' = 97, 'Z' = 90, 'z' = 122 */






     }
          return result;
}
```

b. Consider the code below. The **`changeCase_name()`** function should convert the ith name to lower case by calling **`changeCase_by_ref`**, which should in turn call **`changeCase()`**. Complete the implementation of **`changeCase_by_ref`**. You may not change any part of **`changeCase_name`**.

```
          void changeCase_by_ref( char** n ) {  /* Fill-in */



          }

          void changeCase_name(char* names[], int i) { /* No not
          touch */
               changeCase_by_ref( &(names[i]) );
          }
```

Problem 4

a. Complete the following **setName**, **getStudentID**, and **setStudentID** functions:

```
#define MAX_NAME_LEN 128

typedef struct {

 char name[MAX_NAME_LEN];

 unsigned long sid;

} Student;


/* return the name of student s */

const char* getName(const Student* s) {

 return s->name;

}


/* set the name of student s */

void setName(Student* s, const char* name) {

 /* fill me in */




}


/* return the SID of student s */

unsigned long getStudentID(const Student* s) {

 /* fill me in */




}
```

```
 /* set the SID of student s */
void **setStudentID**(Student* s, unsigned long sid) {
 /* fill me in */



}
```

b. What is the logical error in the following function?

```
Student* makeDefault(void) {
 Student s;
 setName(&s, "John");
 setStudentID(&s, 12345678);
 return &s;
}
```