CIS 3362 Final Exam - Part B (Modern Private Key Cryptography) - 25 pts

Date: 12/9/2020
Start Time: 10:40 am EST
End Time: 11:15 am EST

You may use your class notes, reference sheets and calculator. Please still show each step but just put answers of calculations you made in your calculator.

Note: Please put your name in the document you turn in.

1) (8 pts) Let the input to the S-boxes in DES be 8EF943C62DA1, represented in hexadecimal. What is the output from the S-boxes, represented in hex? (Your grade will be 1 point per hex character.)

2) (7 pts) What is the result of the multiplication 03 x B6, in the AES field? In order to get credit, you must show all of your work by hand.

3) (10 pts) The code included below has produced the following output:

```
Gpqef3Yx8/+0LTrq4nmlgrj
```

What was the string entered by the user that produced this output?

To answer the question**, you may run and edit this code as you see fit.** Alternatively, you may solve the problem by hand. There are reliable ways to do either but the former is almost definitely faster. Regardless of which method you choose, explain what you did and explain why it works.

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

char* makeCode(char* word);
char intToChar(int n);
int charToInt(char c);

int main(void) {
    char* ptr = malloc(25);
    scanf("%s", ptr);
    char* res = makeCode(ptr);
    printf("%s\n", res);
    free(ptr);
    free(res);
    return 0;
}
```

```
char* makeCode(char* word) {
    int n = strlen(word);
    char* res = malloc(sizeof(char)*(n+1));
    for (int i=0; i<n; i++)
        res[i] = intToChar(charToInt(word[i])^(i&63));
    res[n] = '\0';
    return res;
}

char intToChar(int n) {
    if (n < 26) return 'A'+n;
    if (n < 52) return 'a'+n-26;
    if (n < 62) return '0'+n-52;
    if (n == 62) return '+';
    return '/';
}

int charToInt(char c) {
    if (c >= 'A' && c <= 'Z') return c - 'A';
    if (c >= 'a' && c <= 'z') return c - 'a' + 26;
    if (c >= '0' && c <= '9') return c - '0' + 52;
    if (c == '+') return 62;
    return 63;
}
```