# ICS 103, Term 093

## Computer Programming in C

**HW# 3 Solution**
**Due date: Monday, August 2, 2010**

**Q.1.** Numbers represented in hexadecimal representation have the base 16 and the digits (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F). Write a **recursive function** that receives an integer and displays its hexadecimal representation displaying only non-zero digits. Write a C program that asks the user to enter a decimal number and displays its hexadecimal representation using the developed function.

*Sample executions of the program are shown below:*

```
Enter a decimal number: 15
Hexadecimal number is F
Press any key to continue . . .
```

```
Enter a decimal number: 35
Hexadecimal number is 23
Press any key to continue . . .
```

```
Enter a decimal number: 1000
Hexadecimal number is 3E8
Press any key to continue . . .
```

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

void ToHex(int n);
char Hex(int d);

int main(void){

    int n;

    printf("Enter a decimal number: ");
    scanf("%d",&n);
    printf("Hexadecimal number is ");
    ToHex(n);
    printf("\n");

    system("pause");
    return 0;
}

void ToHex(int n){
     int r, q;
```

```
    r = n%16;
    q = n/16;
    if (q !=0) ToHex(q);
    printf("%c", Hex(r));
}

char Hex(int d){
    switch (d){
          case 0: return '0';
          case 1: return '1';
          case 2: return '2';
          case 3: return '3';
          case 4: return '4';
          case 5: return '5';
          case 6: return '6';
          case 7: return '7';
          case 8: return '8';
          case 9: return '9';
          case 10: return 'A';
          case 11: return 'B';
          case 12: return 'C';
          case 13: return 'D';
          case 14: return 'E';
          case 15: return 'F';
          default: printf("Error...\n");
    }
}
```

**Q.2.** You are required to write a program to read a text file and encrypt it and store the encrypted text into another file. The same program can be used to decrypt the encrypted file. Encryption/decryption is performed according to a password entered by the user. To implement your program, do the following:

**(i)** Write a function that receives an integer value and returns a value between 0 and 15. The digit is computed by summing all the digits of the entered number and if the sum is greater than 15, then the process is repeated until the sum becomes within the range [0,15]. For example, if the user enters 987, then the program computes 9+8+7=24>15. Then, the process is repeated and 2+4=6. Thus, the function will return the value 6.

**(ii)** Write a function that receives a character C and returns an encrypted character computed as C XOR N (XOR operation is ^ in C), where N is the number computed by the function in (i) with its input being a randomly generated number. You can generate a random number using the function rand().

**(iii)** Ask the user to enter the file name to encrypt/decrypt and read it. Then, ask the user to enter the output file name and read it. You can define the input and output file names as char infname[40], outfname[40]. Open the input file for reading and the output file for writing handling file not found error.

**(iv)** Ask the user to enter a password integer number, pwd, and read it. Initialize the random number generator in the beginning of your program by the entered password using the function srand(pwd).

**(v)** Read each character in the input file and encrypt/decrypt it if it is not a new line character. Save the encrypted/decrypted character in the output file. A new line character should be written as is. Encryption/decryption of each character is performed by the function developed in (ii).

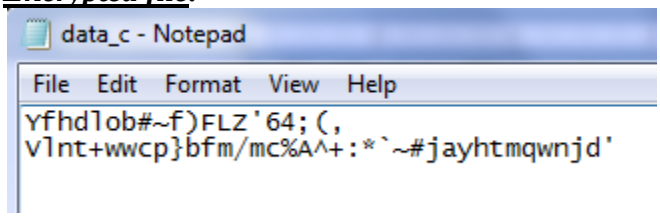
**(vi)** Test the correctness of your program by encrypting and then decrypting the message given below using the password 2010:
"Welcome to ICS 103!!
This question of HW#2 is interesting."

*Sample executions of the program are shown below:*

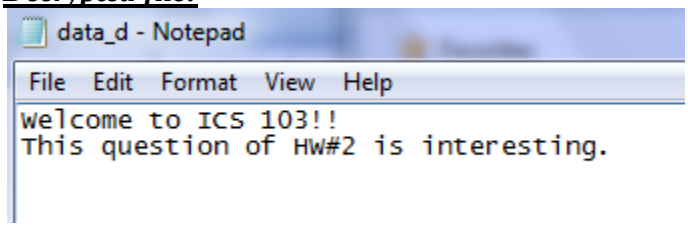
```
Enter the input file name to encrypt/decrypt: data.txt
Enter the output file name: data_c.txt
Enter your password number: 2010
Encryption/Decryption process completed ...
Press any key to continue . . .
```

```
Enter the input file name to encrypt/decrypt: data_c.txt
Enter the output file name: data_d.txt
Enter your password number: 2010
Encryption/Decryption process completed ...
Press any key to continue . . .
```

***Encrypted file:***




***Decrypted file:***




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

int convert(int n);
char encrypt(char c);

int main(void){

int pwd;
FILE *inf, *outf;
char c;
char infname[40], outfname[40];
```

```
  printf("Enter the input file name to encrypt/decrypt: ");
  scanf("%s",infname);
  inf = fopen( infname, "r");
  if (inf == NULL){
     printf("Cannot open %s for reading \n", infname);
     exit(1);
  }

  printf("Enter the output file name: ");
  scanf("%s",outfname);
  outf = fopen( outfname, "w");
  printf("Enter your password number: ");
  scanf("%d",&pwd);
  srand(pwd);

  while (  fscanf(inf,"%c",&c) != EOF ) {
      if ( c  !=  '\n'  )
          c = encrypt(c);
      fprintf(outf,"%c",c);
  }

  printf("Encryption/Decryption process completed ...\n");
  system("pause");
  return 0;
}

int convert(int n){

    int r, sum;

    while (n>15){
          sum = 0;
          do {
                r = n%10;
                n = n/10;
                sum += r;
              } while (n !=0);
          n = sum;
    }
    return n;
}

char encrypt(char c){
     int k, j;

     k = rand();
     j = convert(k);
     c ^= j;
     return c;
}
```