

**ICS 103, Term 083**  
**Computer Programming in C**

**HW# 2 Solution**  
**Due date: Saturday, August 8, 2009**

- Q.1.** You are required to write a C program to do the following:
- (i) Ask the user to enter the length and width of a rectangle and read them.
  - (ii) Ask the user to enter a displacement number indicating the position of the rectangle from the left side of the screen and read it.
  - (iii) Ask the user to enter a fill character that will be used to fill the rectangle and read it.
  - (iv) Display the rectangle according to the entered specification. Use functions and modular programming in your solution.

*A sample execution of the program is shown below:*

```
Enter the length and width of rectangle: 4 8
Enter the rectangle displacement: 10
Enter the fill character: *
*****
*****
*****
*****
Press any key to continue . . .
```

```
Enter the length and width of rectangle: 5 5
Enter the rectangle displacement: 20
Enter the fill character: 0
00000
00000
00000
00000
00000
Press any key to continue . . .
```

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

```
void disps(int size);
void displ(int size, char fill);
void dispr (int length, int width, int dis, char fill);
```

```
int main(void)
{
    int length, width, dis;
    char fill;
    int i;
    printf("Enter the length and width of rectangle: ");
    scanf("%d %d",&length, &width);
```

```

printf("Enter the rectangle displacement: ");
scanf("%d",&dis);
printf("Enter the fill character: ");
scanf(" %c", &fill);
dispr (length, width, dis, fill);

system("pause");
return 0;
}

void disps(int size){
    int i;
    for (i=0; i<size; i++)
        printf(" ");
}

void displ(int size, char fill){
    int i;
    for (i=0; i<size; i++)
        printf("%c",fill);
    printf("\n");
}

void dispr (int length, int width, int dis, char fill)
{
    int i;

    for (i=0; i<length; i++){
        disps(dis);
        displ(width, fill);
    }
}

```

- 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) 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]`. You can read the input file name using the function `gets(infname)`, which is available in `stdlib`.
  - (ii) Open the input file for reading and the output file for writing handling file not found error.
  - (iii) Ask the user to enter a password number, `pwd`, and read it.
  - (iv) 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 XORING the character with a randomly generated value in the range [0-15]. Initialize the random number generator in the beginning of your program by the entered password using the function `srand(pwd)`. You can generate a random number in the range [0-15] using the function `rand()%16`.

- (v) Test the correctness of your program by encrypting and then decrypting the message given below using the password 1967:  
“Welcome to ICS 103!!  
This question of HW#2 is interesting.”

*A sample execution of the program is shown below:*

```
Enter the input file name to encrypt/decrypt: data.txt
Enter the output file name: data_c.txt
Enter your password number: 1967
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: 1967
Encryption/Decryption process completed ...
Press any key to continue . . .
```

**Encrypted file:**

```
Yoc\hgi x\*MD[]81?${+
^hep+x~byraja jn$JT,3,o[]jftjyetynjg)
```

**Decrypted file:**

```
Welcome to ICS 103!!
This question of HW#2 is interesting.
```

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

```
int main(void)
{
```

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

```
    printf("Enter the input file name to encrypt/decrypt: ");
    gets(infname);
    inf = fopen( infname, "r");
    if (inf == NULL){
```

```

printf("Cannot open %s for reading \n", infname);
exit(1);
}
printf("Enter the output file name: ");
gets(outfname);
outf = fopen( outfname, "w");

printf("Enter your password number: ");
scanf("%d",&pwd);
srand(pwd);

while ( fscanf(inf,"%c",&c) != EOF ) {
    if ( c != '\n' ){
        j = rand()%16;
        c ^= j;
        fprintf(outf,"%c",c);
    }
    else
        fprintf(outf,"%c",c);
}

printf("Encryption/Decryption process completed ...\n");

system("pause");
return 0;
}

```

- Q.3.** You are required to write a program that reads students ID's along with their exam scores from a file and computes the average, maximum and minimum. In addition, the program should assign a letter grade to each student according to the criteria given below:

Score	Grade
score >= 90	A+
85 <= score < 90	A
80 <= score < 85	B+
75 <= score < 80	B
70 <= score < 75	C+
65 <= score < 70	C
60 <= score < 65	D+
55 <= score < 60	D
score < 55	F

Then, your program should compute for each letter grade the number and percentage of students who got that grade. Finally, the program should compute the GRADE POINT AVERAGE (GPA) according to the following formula:

$$GPA = \frac{4A^+ + 3.75A + 3.50B^+ + 3B + 2.50C^+ + 2C + 1.50D^+ + 1D}{A^+ + A + B^+ + B + C^+ + C + D^+ + D + F}$$

All the results should be written in a file specified by the user.

A sample execution of the program is shown below:

```
Enter the input file name: scores.txt
Enter the output file name: grades.txt
Grades generated successfully...
Press any key to continue . . . _
```

scores.txt file:

200835160	93.24
200789790	90.29
200640760	87.94
200614800	81.47
200773150	82.47
200758650	80.00
200793130	88.53
200717090	77.06
200745290	75.59
200781030	74.12
200719570	77.18
200767570	69.71
200743690	68.24
200721890	69.82
200791670	67.35
200662320	57.06
200771010	55.12
200621800	50.24
200746710	60.00
200781890	70.00

grades.txt file:

Student ID	Score	Grade
200835160	93.24	A+
200789790	90.29	A+
200640760	87.94	A
200614800	81.47	B+
200773150	82.47	B+
200758650	80.00	B+
200793130	88.53	A
200717090	77.06	B
200745290	75.59	B
200781030	74.12	C+
200719570	77.18	B
200767570	69.71	C
200743690	68.24	C
200721890	69.82	C
200791670	67.35	C
200662320	57.06	D
200771010	55.12	D
200621800	50.24	F
200746710	60.00	D+
200781890	70.00	C+

Number of students = 20

Average = 73.77

Maximum = 93.24

Minimum = 50.24

	A+	A	B+	B	C+	C	D+	D	F
Number	2	2	3	3	2	4	1	2	1
Percentage	10.0	10.0	15.0	15.0	10.0	20.0	5.0	10.0	5.0
GPA =	2.58								

```

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

int main(void)
{

    FILE *inf, *outf;
    char infname[40], outfname[40];
    int status, id;
    int apc=0, ac=0, bpc=0, bc=0, cpc=0, cc=0, dpc=0, dc=0, fc=0, tc=0;
    double score, sum, max=0, min=100, gpa;

    printf("Enter the input file name: ");
    gets(infname);
    inf = fopen( infname, "r");
    if (inf == NULL){
        printf("Cannot open %s for reading \n", infname);
        system("pause");
        exit(1);
    }
    printf("Enter the output file name: ");
    gets(outfname);
    outf = fopen( outfname, "w");

    fprintf(outf,"Student ID \t Score \t Grade\n");
    fprintf(outf,"----- \t ---- \t ----\n");
    status =fscanf(inf,"%d %lf",&id, &score);
    while ( status != EOF ) {
        fprintf(outf,"%d \t %.2f \t ",id, score);
        if (score>= 90) {fprintf(outf,"A+\n"); apc++;}
        else if (score>= 85) {fprintf(outf,"A\n"); ac++;}
        else if (score>= 80) {fprintf(outf,"B+\n"); bpc++;}
        else if (score>= 75) {fprintf(outf,"B\n"); bc++;}
        else if (score>= 70) {fprintf(outf,"C+\n"); cpc++;}
        else if (score>= 65) {fprintf(outf,"C\n"); cc++;}
        else if (score>= 60) {fprintf(outf,"D+\n"); dpc++;}
        else if (score>= 55) {fprintf(outf,"D\n"); dc++;}
        else {fprintf(outf,"F\n"); fc++;}
        tc++;
        sum += score;
        if (score>max) max = score;
        if (score < min) min = score;
        status =fscanf(inf,"%d %lf",&id, &score);
    }

    fprintf(outf,"\nNumber of students = %d\n",tc);
    fprintf(outf,"Average = %.2f\n",sum/tc);

```

```

fprintf(outf,"Maximum = %.2f\n",max);
fprintf(outf,"Minimum = %.2f\n\n",min);

fprintf(outf,"          A+  A  B+  B  C+  C  D+  D  F\n");
fprintf(outf,"-----\n");
fprintf(outf,"Number   %4d %4d %4d %4d %4d %4d %4d %4d %4d\n",\
        apc,ac,bpc,bc,cpc,cc,dpc,dc,fc);
fprintf(outf,"Percentage %4.1f %4.1f %4.1f %4.1f %4.1f %4.1f %4.1f %4.1f %4.1f\n",\
        apc*100.0/tc,ac*100.0/tc,bpc*100.0/tc,bc*100.0/tc,\
        cpc*100.0/tc,cc*100.0/tc,dpc*100.0/tc,dc*100.0/tc,fc*100.0/tc);

gpa = (4*apc+3.75*ac+3.5*bpc+3*bc+2.5*cpc+2*cc+1.5*dpc+1*dc)/tc;
fprintf(outf,"GPA = %.2f\n",gpa);

printf("Grades generated successfully...\n");

system("pause");
return 0;
}

```