

**Lab. Exercises #9 Solution (Lab Date: Week 11 of Classes)**

Objectives are to:

1. Continue to practise on the use of arrays, repetition, decisions and functions in solving bigger problems.
2. Also, continue to practise on use of flowcharts.

Que. 1. For the following problem, give the structure chart, programs and flowcharts.

The class quiz consists of 20 True/False questions. There are 50 students who wrote the test. You are required to write a program to calculate the students' grades based on the following. If a student gets 90% or more, he gets an A, but a total score of 80% to 89% receives a B. A score of 70% to 79% receives a C grade, while a score of 60% to 69% receives a D. Anything lower than 60% gets a grade of an F.

Each student's ID and test answers will be entered. The output will be each student's ID, number correct, and grade, along with the single highest score for the class. Hint: You can use four one-dimensional arrays and any other needed structures.

**Solution to Que 1**

- i. In understanding the problem, the following shows the sample input data and output data from the problem description.

ID[4][10]	Answer [4][20]	Numcrrt[4]	Grade[4]
20031	0 1 ..... 0 0	19	A
20032	1 1 ..... 0 1	14	C
20033	0 1 ..... 0 1	10	F
20034	1 0 ..... 0 0	12	D

Anskey[20]	highestscore
0 1 ..... 0 0	19

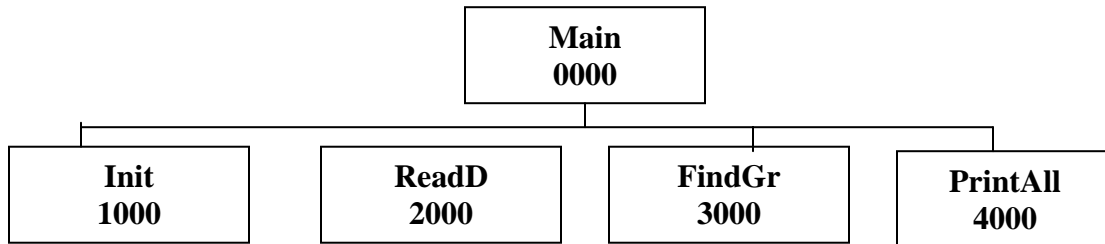
- ii. To be able to test the program, use smaller sized data. Use only 2 students and 5 test questions and test your program with the following sample data:

```

20031
1 0 1 1 1
20032
0 0 0 1 1
1 0 1 1 0

```

iii. You may use the following structure chart.



- Use the Init function to initialize Number\_correct array to 0.
- Use the ReadD function to read input data (ID, Answer, Anskey).
- Use the FindGr function to compute the grade for each student and the highest score.
- Use the function PrintAll to print all output.

Type in the following part of the solution, which only reads in the input data with the function ReadD and prints only the ID, Answers and Anskey read with the function PrintAll. Your job is to first type and run this program to ensure that you are reading your data well, then, complete the program by writing the function prototypes, calls and definitions for the functions Init, FindGr and PrintAll that will solve the given problem.

When you run the above sample program, you need to type in input data at the keyboard in the order you asked for them to be read in the function ReadD. You should type each ID on a separate line, followed by the 5 answers for this ID on the next line. At the end, you should type the 5 answers in the answer key. The input at the keyboard should look like the following:

```

20031
1 0 1 1 1
20032
0 0 0 1 1
1 0 1 1 0

```

Hand in your completed work to lab instructor and GA in a script file.

```

sol:~/bk2010/programs>cat lab9slnq1.c
#include <stdio.h>
#define STU 2
#define QUE 5

void Init (int []);
void ReadD (char id[][10], int answers[][QUE], int anskey[]);
int FindGr (int answers[][QUE], int anskey[], int numcrprt[], char grade[]);
void PrintAll (char id[][10], int numcrprt[], char grade[], int highsc);

int main (void) {
    int csize, numcrprt[STU], highestsc;
    char id[STU][10], grade[STU];
    int answers[STU][QUE], anskey[QUE];

    Init (numcrprt);
    ReadD (id, answers, anskey);
    highestsc = FindGr (answers, anskey, numcrprt, grade);
    PrintAll (id, numcrprt, grade, highestsc);
    return 0;
}

void Init (int numcrprt[]){
    int r;
    for (r=0; r<STU; r++){
        numcrprt[r] = 0;
    }
}

void ReadD (char id[][10], int answers[][QUE], int anskey[]){
    int r, c;
    for (r=0; r<STU; r++) {
        printf("Type student ID:\n");
        scanf ("%s", id[r]);
        printf("Type the student's answers:\n");
        for (c=0; c<QUE; c++){
            scanf ("%d", &answers[r][c]);}
    }
    printf("Type the answer key:\n");
    for (c=0; c<QUE; c++)
        scanf ("%d", &anskey[c]);
}

int FindGr (int answers[][QUE], int anskey[], int numcrprt[], char grade[]){
    int r, c;
    int highsc=0;
    for (r=0; r<STU; r=r+1)
        for (c=0; c<QUE; c=c+1)
            if (answers[r][c] == anskey[c])
                numcrprt[r] +=1;
    for (r=0; r<STU; r=r+1){
        if (numcrprt[r]>highsc)
            highsc=numcrprt[r];
        if (numcrprt[r] >= QUE*.9)
            grade[r] = 'A';
        else if (numcrprt[r] >= QUE*.8)
            grade[r] = 'B';
        else if (numcrprt[r] >= QUE*.7)
            grade[r] = 'C';
        else if (numcrprt[r] >= QUE*.6)
            grade[r] = 'D';
    }
}

```

```

        else
            grade[r] = 'F';
    }
    return highsc;
}
void PrintAll (char id[][10], int numcrtr[], char grade[], int highsc){
    int r;
    printf ("ID      Correct Grade \n");
    for (r=0; r<STU; r++)
        printf ("%s %5d %6c\n", id[r], numcrtr[r], grade[r]);
    printf("Highest Score: %d\n",highsc);
}

```

```
sol:~/bk2010/programs>cc lab9slnq1.c
```

```
sol:~/bk2010/programs>a.out
```

```
Type student ID:
```

```
20031
```

```
Type the student's answers:
```

```
1 0 1 1 1
```

```
Type student ID:
```

```
20032
```

```
Type the student's answers:
```

```
0 0 0 1 1
```

```
Type the answer key:
```

```
1 0 1 1 0
```

```
ID      Correct Grade
```

```
20031      4      B
```

```
20032      2      F
```

```
Highest Score: 4
```

```
sol:~/bk2010/programs>exit
```

```
exit
```

```
script done on Sat Sep 04 21:43:47 2010
```

Que. 2. Solve any of the string processing questions in section 8.9 of book, questions 8, 9, 10, 11.

---

```

/* This implements question 8 chapter (section 8.9) for
   string processing.  It is a program that reads whole
   lines of student records organized as:
       John Adams Computer Science
       Mary Cooks Mathematics
       Todd Black English
   The program prints the output as first name, last name
   and department.
*/

```

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

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

```
/*declare function prototypes*/
```

```
main () {
```

```

char line[100], firstname[12], lastname[12], dept1[10], dept2[10];
int r;

printf("    First Name        Last Name        Department\n");
/* Read the entire line of string */

printf("Type the entire input line to read or 999 \n");
gets(line);
while ((strcmp(line,"999") != 0))
{
/* We next use to sscanf to read from line */
/* dept1 and dept2 are used to accommodate embedded blanks
in department names. */

sscanf(line, "%s %s %s %s", firstname, lastname, dept1, dept2);
printf("%12s %12s %10s %s\n", firstname, lastname, dept1, dept2);
gets(line);
}
}

```

The above program with the sample input data will print:

First Name	Last Name	Department
John	Adams	Computer Science
Mary	Cooks	Mathematics
Todd	Black	English