

King Fahd University of Petroleum & Minerals
Information and Computer Science Department
ICS 103: Computer Programming in C (2-3-3) [Term 092]
Homework Assignment #3 [Due Friday June 4 before midnight]

➤ Instructions

- Create **two** program files named q1.cpp and q2.cpp (the files you save from Dev++ or turbo compiler).
- Zip the 2 files in one file named hw3_yourID.zip and upload it in WebCT. Make sure that you include cpp files containing source code and not executable files i.e. files with exe extension.
- **No group work is allowed. The homework solution has to be your own work. Any cheating will lead to severe consequences.**

Question 1: (15 points)

Write a c program that does the following:

- i) Asks the user to enter the number of rows and columns in an array
- ii) Reads an array of integers of the specified number of rows and columns
- iii) Display a menu for the user as follows:
 1. Print Array
 2. Print a Row
 3. Print a Column
 4. Swap Two Rows
 5. Swap Two Columns
 6. Exit the program
- iv) Implement each of the menu options 1 to 5 as separate functions. Print the array after swapping rows or columns.

```
D:\ics-103\092\workarea\hw3q1.exe
Enter the number of rows and columns: 2 3
Enter 2x3 array of inetegers:
1 2 3
4 5 6
Select one of the choices below:
1. Print Array
2. Print a Row
3. Print a Column
4. Swap Two Rows
5. Swap Two Columns
6. Exit the program
1
the 2 x 3 array is
1 2 3
4 5 6
Select one of the choices below:
1. Print Array
2. Print a Row
3. Print a Column
4. Swap Two Rows
5. Swap Two Columns
6. Exit the program
2
Enter a row to print >1
4 5 6
Select one of the choices below:
1. Print Array
2. Print a Row
3. Print a Column
4. Swap Two Rows
5. Swap Two Columns
6. Exit the program
3
Enter a column to print >2
3
6
Select one of the choices below:
1. Print Array
2. Print a Row
3. Print a Column
4. Swap Two Rows
5. Swap Two Columns
6. Exit the program
```

```
D:\ics-103\092\workarea\hw3q1.exe
Enter the number of rows and columns: 2 3
Enter 2x3 array of inetegers:
1 2 3
4 5 6
Select one of the choices below:
1. Print Array
2. Print a Row
3. Print a Column
4. Swap Two Rows
5. Swap Two Columns
6. Exit the program
4
Enter the two rows to swap: 0 1
Array after exchanging row 0 & row 1:
4 5 6
1 2 3
Select one of the choices below:
1. Print Array
2. Print a Row
3. Print a Column
4. Swap Two Rows
5. Swap Two Columns
6. Exit the program
5
Enter the two columns to swap: 1 2
Array after exchanging column 1 & column 2:
4 6 5
1 3 2
Select one of the choices below:
1. Print Array
2. Print a Row
3. Print a Column
4. Swap Two Rows
5. Swap Two Columns
6. Exit the program
6
Thank you for using the program
Press any key to continue . . .
```

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 10
void menu();
void read_array(int a[][SIZE],int n, int m);
void print_array(int a[][SIZE],int n, int m);
void swap_rows(int a[][SIZE], int m, int r1, int r2);
void swap_columns(int a[][SIZE], int n, int c1, int c2);
void print_row(int a[][SIZE], int cols,int rownum);
void print_col(int a[][SIZE], int rows,int colnum);
int main(void) {

    int a[SIZE][SIZE];
```

```

int row, col, v1, v2, choice;
int rownum,colnum;
printf("Enter the number of rows and columns: ");
scanf("%d %d",&row, &col);
read_array(a,row,col);

do {
    menu();
scanf("%d",&choice);
    switch(choice){
        case 1: printf("the %d x %d array is\n",row,col);
                print_array(a,row,col);
                break;
        case 2:printf("Enter a row to print >");
                scanf("%d",&rownum);
                print_row(a,col,rownum);
                break;
        case 3: printf("Enter a column to print >");
                scanf("%d",&colnum);
                print_col(a,row,colnum);
                break;
        case 4:
                printf("Enter the two rows to swap: ");
                scanf("%d%d",&v1,&v2);
                swap_rows(a,col,v1,v2);
                printf("Array after exchanging row %d & row
%d:\n",v1, v2);
                print_array(a,row,col);
                break;
        case 5:
                printf("Enter the two columns to swap: ");
                scanf("%d %d",&v1,&v2);
                swap_columns(a,row,v1,v2);
                printf("Array after exchanging column %d & column
%d:\n",v1, v2);
                print_array(a,row,col);
                break;
        case 6: printf("Thank you for using the program\n");
                break;
        default:
                printf("Invalid choice!!\n");
    }
}

```

```

    } while(choice!=6);

    system("pause");
    return 0;
}

void menu(){
    printf("Select one of the choices below:\n");
    printf("1. Print Array\n");
    printf("2. Print a Row\n");
    printf("3. Print a Column\n");
    printf("4. Swap Two Rows\n");
    printf("5. Swap Two Columns\n");
    printf("6. Exit the program\n");
}

void read_array(int a[][SIZE],int n, int m){
    printf("Enter %dx%d array of inetegers:\n",n,m);
    for (int i=0;i<n;i++)
        for (int j=0;j<m;j++)
            scanf("%d",&a[i][j]);
}

void print_array(int a[][SIZE],int n, int m){
    for (int i=0;i<n;i++){
        for (int j=0;j<m;j++)
            printf("%3d ",a[i][j]);
        printf("\n");
    }
}

void print_row(int a[][SIZE], int cols,int rownum){
    for (int j=0;j<cols;j++)
        printf("%d ",a[rownum][j]);
    printf("\n");
}

void print_col(int a[][SIZE], int rows,int colnum){
    for (int i=0;i<rows;i++)
        printf("%d\n",a[i][colnum]);
}

void swap_rows(int a[][SIZE], int m, int r1, int r2){
    int t;
    for (int j=0;j<m;j++){
        t=a[r1][j];
        a[r1][j]=a[r2][j];
        a[r2][j]=t;
    }
}

```

```

    }
}

void swap_columns(int a[][SIZE], int n, int c1, int c2){
    int t;
    for (int i=0;i<n;i++){
        t=a[i][c1];
        a[i][c1]=a[i][c2];
        a[i][c2]=t;
    }
}

```

Question 2: (15 points)

Write a program that does the following:

- i) Read from a file "scores.txt" the scores of 5 sections in an exam and store the results in a two-dimensional array. Assume that the maximum number of students in each section is 20 and that all sections have the same number of students.
- ii) Sort the scores of each section from highest to lowest.
- iii) For each of the sections, print the section number, sorted scores, highest, lowest and average. The results should be printed in the output file "results.txt".

Note. The input file "scores.txt" is organized as follows:

The first line contains 5 values corresponding to the number of students in each section. This is followed by the scores of each section i.e. scores of section 1 followed by scores of section 2 and so on. The selection_sort function of the slides can be modified to make the sorting decreasing, then it can be used to sort each row of the 2-D array by calling it with the name of the array plus 1 square bracket corresponding to the row index.

Examples of "scores.txt" file and generated "results.txt" files are shown below.

```

scores.txt - Notepad
File Edit Format View Help
3 4 5 6 7
56.6 78.9 34.6
23 99.5 67.8 77.5
67.6 78.3 81.2 97.4 87.5
55.4 41.3 38.9 79.6 66.4 78.6
56.7 48.7 89.5 35.4 88.7 95.3 98.5

```

```
results.txt - Notepad
File Edit Format View Help
section 1
scores: 78.90    56.60    34.60
highest=78.90
lowest=34.60
average=56.70
section 2
scores: 99.50    77.50    67.80    23.00
highest=99.50
lowest=23.00
average=66.95
section 3
scores: 97.40    87.50    81.20    78.30    67.60
highest=97.40
lowest=67.60
average=82.40
section 4
scores: 79.60    78.60    66.40    55.40    41.30    38.90
highest=79.60
lowest=38.90
average=60.03
section 5
scores: 98.50    95.30    89.50    88.70    56.70    48.70    35.40
highest=98.50
lowest=35.40
average=73.26
```

```
#include <stdio.h>
#include <stdlib.h>
#define ROWS 5
#define COLS 20
void selection_sort(double a[], int size);
int find_max(double a[], int start, int size);
void swap(double *a, double *b);
double get_average(double a[], int size);
void get_max_min(double a[], int size, double *max, double
*min);
int main() {
    double scores[ROWS][COLS],max,min,average;
    int students[ROWS],i,j;
    FILE *in, *out;
    in=fopen("scores.txt","r");
    out=fopen("results.txt","w");
    for(i=0;i<ROWS;i++)
        fscanf(in,"%d",&students[i]);
    for(i=0;i<ROWS;i++) {
```

```

    for(j=0;j<students[i];j++)
        fscanf(in,"%lf",&scores[i][j]);
    }

    for(i=0;i<ROWS;i++) {
        selection_sort(scores[i],students[i]);
        average=get_average(scores[i], students[i]);
        get_max_min(scores[i], students[i],&max,&min);
        fprintf(out,"section %d\nscores: ",i+1);
        for(j=0;j<students[i];j++)
            fprintf(out,"%0.2f\t",scores[i][j]);
        fprintf(out,"\n");
    }

    fprintf(out,"highest=%0.2f\nlowest=%0.2f\naverage=%0.2f\n",max,min,
average);
}
fclose(in);
fclose(out);
system("pause");
return 0;
}

void selection_sort(double a[], int size) {
    int i, max_pos;

    for (i = 0; i<=size-2; i++) {
        max_pos = find_max(a, i, size);
        swap(&a[i], &a[max_pos]);
    }
}

int find_max(double a[], int start, int size) {
    int i, max_index = start;

    for (i=start+1; i<size; i++)
        if (a[i] > a[max_index])
            max_index = i;

    return max_index;
}

void swap(double *a, double *b) {
    double temp = *a;

```

```
    *a = *b;
    *b = temp;
}
double get_average(double a[], int size) {
    int i;
    double sum = 0;

    for (i = 0; i < size; ++i)
        sum += a[i];

    return sum/size;
}
void get_max_min(double a[], int size,
                 double *max, double *min) {
    int i;
    *max = a[0];
    *min = a[0];
    for (i = 1; i < size; ++i) {
        if (a[i] > *max)
            *max = a[i];
        else if (a[i] < *min)
            *min = a[i];
    }
}
```