Wanner HernadezRiveron

November 24, 2019

CMIS 102

Week 8 Lab 8 Concerts

1. Demonstrate you successfully followed the steps in this lab by preparing screen captures of you running the lab as specified in the Instructions above.

a. Add a function to print welcome information, including your name and an introduction to the project, and, of course, call this function as the first instruction in main.

// C code

// // This code will compute the values of the sales ticket sales for concerts

// and sort the entries by those values

// Wanner HernandezR CMIS 102

// November 24, 2019

#include <stdio.h>

#define MAXN 100

// max characters in a group/concert name

#define MAXG 50

// max concerts/groups

#define MAXC 3

// max categories

char group [MAXG][MAXN];

int fans [MAXG][MAXC];

float prices [MAXC];

float sales [MAXG];

int count = 0;

void welcome () {

printf ("\nWelcome to the User Friendly Ticket Sales Calculator\n");

printf ("This application will aid you calculating ticket sales: \n");

printf ("Created By: Wanner HernandezRiveron \n\n\n");

} // end function welcome

void printArray () {

printf ("%15s%5s%5s%5s%10s\n", "Concert", "s1", "s2", "s3", "Sales");

for (int i = 0; i < count; i++) {

printf ("%15s", group [i]);

for (int j = 0; j < MAXC; j++) {

printf ("%5d", fans[i][j]);

} // end for each category

printf ("%10.2f\n", sales [i]);

} // end for each group

} // end function printArray

void computeSales () {

for (int i = 0; i < count; i++) {

sales [i] = 0;

for (int j = 0; j < MAXC; j++) {

sales [i] += prices [j] \* fans [i][j];

} // end for each category

} // end for each group

} // end function computeSales

void switchRows (int m, int n) {

char tc;

int ti;

float v;

// printf ("Switching %d with %d\n", m, n);

for (int i = 0; i < MAXN; i++) {

tc = group [m][i];

group [m][i] = group [n][i];

group [n][i] = tc;

} // end for each character in a group name

for (int i = 0; i < MAXC; i++) {

ti = fans [m][i];

fans [m][i] = fans [n][i];

fans [n][i] = ti;

} // end for each fan category

v = sales [m];

sales [m] = sales [n];

sales [n] = v;

} // end switch

int findMinSales (int m) {

float min = sales [m];

int target = m;

for (int i = m+1; i < count; i++)

if (sales [i] < min) {

min = sales [i];

target = i;

}// end new max found

return target;

}// end function findMinSales

void sortBySales () {

int target;

for (int i = 0; i < count; i++) {

target = findMinSales (i);

if (target > i)

switchRows (i, target);

}// for each concert

}// end function sortBySales

void getData () {

// for (int i = 0; i < MAXG; i++) sales [i] = 0;

printf ("Enter ticket prices in each of %d cateogories: ", MAXC);

for (int i = 0; i < MAXC; i++)

scanf ("%f", &prices [i]);

printf ("-- Enter group and fans in %d categories\n", MAXC);

printf (" . to finish entries:\n");

for (int i = 0; i < MAXG; i++) {

scanf ("%s", group[i]);

if (group [i][0] == '.')

break;

count++;

for (int j = 0; j < MAXC; j++)

scanf ("%d", &fans[i][j]);

} // end for each group

} // end function getData

int main(void) {

welcome ();

getData ();

computeSales ();

printArray ();

printf ("\n --- Sorted ---\n");

sortBySales ();

printArray ();

printf("... bye ...\n");

return 0;

}

|  |  |  |
| --- | --- | --- |
| Test Case | Input | Actual Output using repl.it – user input in red: |
| 1 | 1 2 3  a 1 2 3  b 3 3 1  c 5 3 1  d 3 3 5  e 1 1 2  f 9 4 3  g 4 5 6  . | Enter ticket prices in each of 3 cateogories: 1 2 3  -- Enter group and fans in 3 categories  . to finish entries:  a 1 2 3  b 3 3 1  c 5 3 1  d 3 3 5  e 1 1 2  f 9 4 3  g 4 5 6  .  ... bye ...  Concert s1 s2 s3 Sales  a 1 2 3 14.00  b 3 3 1 12.00  c 5 3 1 14.00  d 3 3 5 24.00  e 1 1 2 9.00  f 9 4 3 26.00  g 4 5 6 32.00  --- Sorted ---  Concert s1 s2 s3 Sales  e 1 1 2 9.00  b 3 3 1 12.00  c 5 3 1 14.00  a 1 2 3 14.00  d 3 3 5 24.00  f 9 4 3 26.00  g 4 5 6 32.00 |

Test case 1:

A screenshot of a computer

Description automatically generated

1. Modify the program to add a function to compute and display the total sales for all the concerts. Support your experimentation with screen captures of executing the new code.

The change I made to the program is that I added line 102 to 110 and that was able to calculate the total sales. I use the same Input to test my result.

void totalSalesCalculator (){

// function to calculate the total sales

float total = 0;

int salesSize = sizeof(sales);

for (int i = 0; i < salesSize; i++) {

total += sales[i];

}

printf("\nThe total Sale is %.2lf \n\n", total);

}// end function totalSalesCalculator

Test Sample:

A screenshot of a computer screen

Description automatically generated

1. Enhance the program to allow the user to enter 4 categories.  
   Explain the changes you made to the code, and create appropriate test data files. Support your experimentation with screen captures of executing the new code.

In this program I change line 11 and line 27 so that the program would max categories (had to increase this to 4).

Line 11 : #define MAXC 4 // max categories (Had to increase this to 4)

Line 27 : printf ("%15s%5s%5s%5s%10s\n", "Concert", "s1", "s2", "s3", "s4" "Sales"); // had to add (s4)

Test Sample:

A screenshot of a computer

Description automatically generated

1. Prepare a new test table with at least 2 more test cases listing input and expected output for the new code you created, supporting 4 categories.

|  |  |  |
| --- | --- | --- |
| Test Case | Input | Output |
| 1 | 1 2 3 4  a 1 2 3 4  b 3 3 1 4  c 5 3 1 4  . | The total Sale is 88.00 |
| 2 | 1 2 3 4  a 1 2 3 20  b 1 10 1 5  c 5 3 5 25  . | The total Sale is 264.00 |
| 3 | a b c d  a b c d  a c d f  a f d c | The total Sales are calculated as 0.00  This error is cased by entering abc in the category. |

Test case 1:

A screenshot of a computer screen

Description automatically generated

Test case 2:

A screenshot of a computer screen

Description automatically generated

Test case 3:

A screenshot of a computer

Description automatically generated

1. Create a variety of test cases focusing on the sorting algorithm, such as the final element is the smallest, the entire set is already sorted, etc.  
   Explain the purpose of each test case, and check your code against each of those cases.

|  |  |  |
| --- | --- | --- |
| Test Case | Input | Output |
| 1 | 1 2 3 4  a1 2 3 4 b1 2 3 4 c1 2 3 4  . | The total Sale is 90.00  Test case where the values entered are the same no sorting needed |
| 2 | 1 2 3 4  a1 -2 3 4 b1 2 -3 4 c1 2 3 -4  . | The total Sale is 32.00 2  Test case including a negative value |
| 3 | 1 2 3 4  a4 4 4 4 b3 3 3 3 c2 2 2 2  . | The total Sales are calculated as 90.00  Test case where the values entered are from largest to smallest sorting needed. |

Test case 1:

A screenshot of a computer screen

Description automatically generated

Test case 2:

A screenshot of a computer

Description automatically generated

Test case 3:

A screenshot of a computer

Description automatically generated

1. Try using different inputs:
   1. What changes would you suggest to handle larger, more realistic numbers?

A screenshot of a computer

Description automatically generated

* 1. What happens if any of the numbers, such as the ticket prices, are negative?

A screenshot of a computer screen

Description automatically generated

* 1. What are your recommendations concerning negative input values?

I’m unclear as to the intended question. If you use this program for all aspects negative numbers could be needed. As concerts are subject to refunds be that due to weather or whatever.

1. What changes should be made to the code if the customer wished to sort on the number of fans in category 1, the first of the three (or four) categories?  
   Make those changes, test your code and confirm that it is working correctly.

In this program I change line 66 to 73 so that can find a New max.

int findMinSales (int m) {

float min = fans [m][0];

int target = m;

for (int i = m+1; i < count; i++)

if (fans [i][0] < min) {

min = fans [i][0];

target = i;

}// end new max found

return target;

}// end function findMinSales

Final Code:

// C code

// // This code will compute the values of the sales ticket sales for concerts

// and sort the entries by those values

// Wanner HernandezR CMIS 102

// November 24, 2019

#include <stdio.h>

#define MAXN 100

// max characters in a group/concert name

#define MAXG 50

// max concerts/groups

#define MAXC 4

// max categories (Had to increase this to 4)

char group [MAXG][MAXN];

int fans [MAXG][MAXC];

float prices [MAXC];

float sales [MAXG];

int count = 0;

void welcome () {

printf ("\nWelcome to the User Friendly Ticket Sales Calculator\n");

printf ("This application will aid you calculating ticket sales: \n");

printf ("Created By: Wanner HernandezRiveron \n CMIS 102 \n\n\n");

} // end function welcome

void printArray () {

printf ("%15s%5s%5s%5s%10s\n", "Concert", "s1", "s2", "s3", "s4" "Sales");

// had to add (s4)

for (int i = 0; i < count; i++) {

printf ("%15s", group [i]);

for (int j = 0; j < MAXC; j++) {

printf ("%5d", fans[i][j]);

} // end for each category

printf ("%10.2f\n", sales [i]);

} // end for each group

} // end function printArray

void computeSales () {

for (int i = 0; i < count; i++) {

sales [i] = 0;

for (int j = 0; j < MAXC; j++) {

sales [i] += prices [j] \* fans [i][j];

} // end for each category

} // end for each group

} // end function computeSales

void switchRows (int m, int n) {

char tc;

int ti;

float v;

// printf ("Switching %d with %d\n", m, n);

for (int i = 0; i < MAXN; i++) {

tc = group [m][i];

group [m][i] = group [n][i];

group [n][i] = tc;

} // end for each character in a group name

for (int i = 0; i < MAXC; i++) {

ti = fans [m][i];

fans [m][i] = fans [n][i];

fans [n][i] = ti;

} // end for each fan category

v = sales [m];

sales [m] = sales [n];

sales [n] = v;

} // end switch

int findMinSales (int m) {

float min = fans [m][0];

int target = m;

for (int i = m+1; i < count; i++)

if (fans [i][0] < min) {

min = fans [i][0];

target = i;

}// end new max found

return target;

}// end function findMinSales

void sortBySales () {

int target;

for (int i = 0; i < count; i++) {

target = findMinSales (i);

if (target > i)

switchRows (i, target);

}// for each concert

}// end function sortBySales

void getData () {

// for (int i = 0; i < MAXG; i++) sales [i] = 0;

printf ("Enter ticket prices in each of %d cateogories: ", MAXC);

for (int i = 0; i < MAXC; i++)

scanf ("%f", &prices [i]);

printf ("-- Enter group and fans in %d categories\n", MAXC);

printf (" . to finish entries:\n");

for (int i = 0; i < MAXG; i++) {

scanf ("%s", group[i]);

if (group [i][0] == '.')

break;

count++;

for (int j = 0; j < MAXC; j++)

scanf ("%d", &fans[i][j]);

} // end for each group

} // end function getData

void totalSalesCalculator (){

// function to calculate the total sales

float total = 0;

int salesSize = sizeof(sales);

for (int i = 0; i < salesSize; i++) {

total += sales[i];

}

printf("\nThe total Sale is %.2lf \n\n", total);

}//End function totalSalesCalculator

int main(void) {

welcome ();

getData ();

computeSales ();

printArray ();

printf ("\n --- Sorted ---\n");

sortBySales ();

printArray ();

totalSalesCalculator();

printf("... bye ...\n");

return 0;

}

Test sample:

A screenshot of a computer screen

Description automatically generated