CSC 150 Homework

Chapter 7

John Akujobi – Fall 2022

**Q 1**

Identify an error in the following C statements: int x[8], i; for (i = 0; i <= 8; ++i) x[i] = i;

Will the error be detected? If so, when?

In the for loop the loop counter i is allowed to reach 8

We will get incorrect results from the code because the for loop will fetch an external value from the memory outside the array.

And use this wrong value as part of the operation

**Q 2**

Declare an array of type double values called **exper** that can be referenced by using any day of the week as a subscript, where 0 represents Sunday, 1 represents Monday, and so on.

double exper[7];

**Q 3**

The statement marked /\* this one */ in the following code is valid. True or false?* int counts[10], i; double x[5]; printf("Enter an integer between 0 and 4> "); i = 0; scanf("%d", &counts[i]); x[counts[i]] = 8.384; / this one \*/

True, the code is valid

**Q 4**

What are the two common ways of selecting array elements for processing?

Using the array index of the elements

E.g int array[4]

//This picks the first element of the array1

Using pointers

E.g int \*array1+4

//source: <https://www.wikiind.com/2011/05/what-are-two-common-ways-of-selecting/>

**Q 5**

Write a C program segment to display the index of the smallest and the largest numbers in an array x of 20 integers. Assume array x already has values assigned to each element.

//For loop to find the index of the smallest and largest numbers in the array

for (i = 0; i < 20; i++)

{

if (arr[i] < arr[smallest])

{

smallest = i;

}

if (arr[i] > arr[largest])

{

largest = i;

}

}

//Print the index of the smallest and largest numbers in the array

printf("The smallest number is %d and it is at index %d in the array.\\n", arr[smallest], smallest);

printf("The largest number is %d and it is at index %d in the array.\\n", arr[largest], largest);

**Q 6**

Write a C function called reverse that takes an array named x as an input parameter and an array named y as an output parameter. A third function parameter is n , the number of values in x . The function should copy the integers in x into y but in reverse order (i.e., y[0] gets x[n - 1], . . . y[n - 1] gets x[0] ).

//The function is called with reverse(x, y, n);

//where x is the array to be reversed

//y is the array that will be reversed

//and n is the number of elements in the array

void reverse(int x[], int y[], int n)

{

//This for loop copies the values of x into y in reverse order

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

{

y[i] = x[n - i - 1];

}

}

**Q 7**

Write a program segment to display the sum of the values in each row of a 5 × 3 type double array named table. How many row sums will be displayed? How many elements are included in each sum?

for (i = 0; i < 5; i++)

{

//This for loop calculates the sum of the values in each column

//runs 3 times as there are 3 columns

for (j = 0; j < 3; j++)

{

sum += table[i][j];

}

//Print the sum of the values in each row

printf("Sum of the values in row %d is %5.2f.\\n", i, sum);

//Reset the sum to 0 for the next row

sum = 0;

}

**There were five row sums displayed**

**Each of the sum displayed included 3 elements (from each column of the row)**

**Q 8**

Answer Review Question 7 for the column sums

//This for loop calculates the sum of the values in each column

//and prints the sum of each column

//runs 3 times as there are 3 columns

for (j = 0; j < 3; j++)

{

//This for loop calculates the sum of the values in each row

//runs 5 times as there are 5 rows

for (i = 0; i < 5; i++)

{

sum += table[i][j];

}

//Print the sum of the values in each column

printf("Sum of the values in column %d is %5.2f.\\n", j, sum);

//Reset the sum to 0 for the next column

sum = 0;

}

**There were three column sums displayed**

**Each of the sums displayed included 5 elements (from each row of the column)**

**Q 9**

Write a function for displaying (as a string) a value of enumerated type season\_t : typedef enum {winter, spring, summer, fall} season\_t;

//The function is called with display\_season(season);

//where season is the variable of the enumerated type season\_t

void display\_season(season\_t season)

{

//This switch statement prints the string value of the variable season

switch (season)

{

case winter:

printf("Season is winter.\\n");

break;

case spring:

printf("Season is spring.\\n");

break;

case summer:

printf("Season is summer.\\n");

break;

case fall:

printf("Season is fall.\\n");

break;

}

**Q 10**

Also show the following array after each pass using the bubble sort, then show the same array after each pass using the selection sort:

32 12 46 18 56 30 27

Using the bubble sort:

32 12 46 18 56 30 27 //Pass 1

12 32 18 46 30 27 56 //Pass 2

12 18 32 30 27 46 56 //Pass 3

12 18 30 27 32 46 56 //Pass 4

12 18 27 30 32 46 56 //Pass 5

12 18 27 30 32 46 56 //Pass 6 (No more passes needed as the array is sorted)

Using the selection sort:

32 12 46 18 56 30 27 //Unsorted array

12 32 46 18 56 30 27 //Smallest is 12 and is swapped with the first value

12 18 46 32 56 30 27 //18 is swapped with the second value

12 18 27 32 56 30 46 //27 is swapped with the third value

12 18 27 30 56 32 46 //30 is swapped with the fourth value

12 18 27 30 32 56 46 //32 is swapped with the fifth value

12 18 27 30 32 46 56 //46 is swapped with the sixth value

12 18 27 30 32 46 56 //56 and is swapped with the itself (No more passes needed as the array is sorted)

**Complete Programs for the Q5, 6, 7, 8 & 9 for testing (Compressed)**

//Question Q5

#include <stdio.h>

int main()

{

//Declaring the array

int arr[20];

//For loop to assign values to the array

//The values go from 1 to 20

for (int i = 0; i < 20; i++)

{

arr[i] = i + 1;

}

int i = 0; //The index of the array

int smallest = 0; //The index of the smallest number in the array

int largest = 0; //The index of the largest number in the array

//For loop to find the index of the smallest and largest numbers in the array

for (i = 0; i < 20; i++)

{

if (arr[i] < arr[smallest])

{

smallest = i;

}

if (arr[i] > arr[largest])

{

largest = i;

}

}

//Print the index of the smallest and largest numbers in the array

printf("The smallest number is %d and it is at index %d in the array.\\n", arr[smallest], smallest);

printf("The largest number is %d and it is at index %d in the array.\\n", arr[largest], largest);

return 0;

}

//Question Q9

#include <stdio.h>

//Declaring the enumerated type season\_t

typedef enum {winter, spring, summer, fall} season\_t;

//Declaring the function

void display\_season(season\_t season);

int main()

{

//Declaring the variable season

season\_t season;

//Assigning the value of season to winter

season = winter;

// //Receive the value of season from the user

// //As a string

// printf("Enter the season: ");

// scanf("%s", &season);

//Calling the function

display\_season(season);

return 0;

}

//The function is called with display\_season(season);

//where season is the variable of the enumerated type season\_t

void display\_season(season\_t season)

{

//This switch statement prints the string value of the variable season

switch (season)

{

case winter:

printf("Season is winter.\\n");

break;

case spring:

printf("Season is spring.\\n");

break;

case summer:

printf("Season is summer.\\n");

break;

case fall:

printf("Season is fall.\\n");

break;

}

}

//Question 6

#include <stdio.h>

void reverse(int x[], int y[], int n);

int main()

{

//Declaring the arrays

int x[5] = {1, 2, 3, 4, 5};

int y[5];

//Calling the function

reverse(x, y, 5);

//Printing the array y

for (int i = 0; i < 5; i++)

{

printf("%d ", y[i]);

}

return 0;

}

//The function is called with reverse(x, y, n);

//where x is the array to be reversed

//y is the array that will be reversed

//and n is the number of elements in the array

void reverse(int x[], int y[], int n)

{

//This for loop copies the values of x into y in reverse order

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

{

y[i] = x[n - i - 1];

}

}

//Questions Q7 and Q8

#include <stdio.h>

int main()

{

//Declaring the array called table

//And filling the array with values

double table[5][3] = {{1.1, 2.2, 3.3},

{4.4, 5.5, 6.6},

{7.7, 8.8, 9.9},

{10.1, 11.2, 12.3},

{13.4, 14.5, 15.6}}; //Arranged like this so that i can easily visualize

//how the rows and columns of the array are filled

//Concept taken from <https://beginnersbook.com/2014/01/2d-arrays-in-c-example/>

//Declare the variables i, j and sum

int i = 0; //Index of the rows of the table array

int j = 0; //Index of the columns of the table array

double sum = 0; //The sum of the values in each row

//This for loop calculates the sum of the values in each row

//and prints the sum of each row

//runs 5 times as there are 5 rows

for (i = 0; i < 5; i++)

{

//This for loop calculates the sum of the values in each column

//runs 3 times as there are 3 columns

for (j = 0; j < 3; j++)

{

sum += table[i][j];

}

//Print the sum of the values in each row

printf("Sum of the values in row %d is %5.2f.\\n", i, sum);

//Reset the sum to 0 for the next row

sum = 0;

}

printf("\\n"); //Print a new line to give space between the row sums and the column sums

//This for loop calculates the sum of the values in each column

//and prints the sum of each column

//runs 3 times as there are 3 columns

for (j = 0; j < 3; j++)

{

//This for loop calculates the sum of the values in each row

//runs 5 times as there are 5 rows

for (i = 0; i < 5; i++)

{

sum += table[i][j];

}

//Print the sum of the values in each column

printf("Sum of the values in column %d is %5.2f.\\n", j, sum);

//Reset the sum to 0 for the next column

sum = 0;

}

return 0;

}