Question **1**

Not complete

Marked out of 1.00

Flag question

Question text

Write a function **averageTemp** which takes a ***filename*** as string argument and returns the ***average temperature*** as float type for the week. Input file will contain temperature readings for each day of the week. Your function should read each line from the given filename, parse and process the data, and return the required information. Your function should return the average temperature for the whole week rounded-off to 2 digits of floating point precision.  Empty lines do not count as entries, and should be ignored. If the input file cannot be opened, return -1. Remember temperature readings can be decimal and negative numbers.

* Each line of the file contains two elements, separated by commas (,)
  + DAY,TEMPERATURE

### Question 2

Not complete

Marked out of 1.00

Flag question

#### Question text

Write a function that takes in a string array and its size and returns the number of times given character appeared in the whole array.

**int countCharacter(string array[ ], int size, char ch**

**)**

Example:

array = {"elephant", "english", "elegant", "america", "united", "apple"}

ch = 'e'

return value: **8**

### Question 3

Not complete

Marked out of 1.00

Flag question

#### Question text

Write a function ***countPrimes*** that takes in two parameters: an int array and the size of the array. The function returns the count of the number of prime elements in the array.

**int** ***countPrimes(int array[], int size)***

Given array: **{1, 2, 3, 4, 5, 7, 10, 12}**

Expected return: **4**

Explanation: The primes are 2, 3, 5, and 7; consequently, the function returns a count of 4.  
  
**Hint:** Create a function **IsPrime** to call from within your countPrimes function.

### Question 4

Not complete

Marked out of 1.00

Flag question

#### Question text

Write a function **diagonalSum** that takes in a 2-D array of **int** type and their row sizes and returns **sum of main diagonal**(left diagonal) elements. (Assume column size is always 3)

**int diagonalSum(int arr[][3], int row)**

Example:

arr[3][3]= {{1, 4, 6}, {7, 23, 3}, {5, 15, 25}}

diagonal sum = 1+23+25

return value: **49**

arr[3][3]= {{1, 4, -6}, {17, -23, 3}, {-5, 15, 5}}

diagonal sum = 1-23+5

return value: **-17**

arr[3][3] = {{-1, 4, 6}, {17, -3, 3}, {5, 15, 4}}

return value: **0**

### Question 5

Not complete

Marked out of 1.00

Flag question

#### Question text

Given an array of integers and the size of the array, write a function ***findDuplicate*** which prints the duplicate element from the array.  
The array consists of all distinct integers except one which is repeated. Find and print the repeated number.  If no duplicate is found, the function should print -1.

***void findDuplicate (int [ ], int)***

**Example 1:**   
     Given array: **{2,3,5,6,11,20,4,8,4,9}**   
     Output: **4**   
  
**Example 2:**   
     Given array: **{1,3,5,6,7,8,2,9}**   
     Output: **-1**

Question **6**

Not complete

Marked out of 1.00

Flag question

Question text

Write a function **maxTemp** which takes a ***filename*** as string argument and returns the ***maximum temperature*** as float type for the week. Input file will contain temperature readings for each day of the week. Your function should read each line from the given filename, parse and process the data, and return the required information. Your function should return the average temperature for the whole week.  Empty lines do not count as entries, and should be ignored. If the input file cannot be opened, return -1. Remember temperature readings can be decimal and negative numbers.

* Each line of the file contains two elements, separated by commas (,)
  + DAY,TEMPERATURE

Monday,52

Note: The function, int Split(string s, char sep, string words[], int max\_words) is provided; your function may call it. It behaves the same way as split from Homework 6 and Recitation.

Example

|  |
| --- |
| **File Contents** |
| Monday,52 Tuesday,55 Wednesday,50 Thursday,45 Friday,39 Saturday,20 Sunday,35 |
| **Return Value**    55 |

### Question 7

Not complete

Marked out of 1.00

Flag question

#### Question text

Create a function named ***PrintStudents***, which takes a string input filename and an integer minimum score value and a string output file name as a parameters.  The function will read the student scores and names from the file and output the names of the students with scores greater than or equal to the value given.  This function returns the integer number of entries read from the file.  If the input file cannot be opened, return -1 and do not print anything to the file.

Read each line from the given filename, parse the data, process the data, and write the required information to the file.

Each line of the file contains <FIRST-NAME LAST-NAME>, <SCORE>,  <SUBJECT> .  Read and parse the data, then write to the output file the names and classes for scores matching the criteria.

Example: With the following data and value of 80:

Constance Shelton, 67, APPM 2002

Charlotte Edwards, 85, CSCI 1300

Alyssa Hill, 78, MATH 1000

Pat Owens, 75, HUMN 1342

Shannon Jimenez, 96, LING 2000

Kristen Swanson, 80, PSYC 1001

Jim Schwartz, 60, CVEN 3241

Your function should return 7 and output to the file should contain:

Charlotte Edwards, CSCI 1300

Shannon Jimenez, LING 200

### Kristen SwQuestion 8

Not complete

Marked out of 1.00

Flag question

Question text

Write a function *printTotalMedals* that takes an array for olympic sports and arrays for gold, silver, and bronze medals and prints the total number of medals by discipline.  
  
Your function should take 5 parameters:

* an array of strings disciplines
* an array of ints representing the number of gold medals
* an array of ints representing the number of silver medals
* an array of ints representing the number of bronze medals
* an integer size for the size of the arrays

**Edge cases:**

If the size < 1, print "Invalid size. Size must be at least 1."

**For example:**

| **Test** | **Result** |
| --- | --- |
| const int SPORTS = 3; // number of sports/disciplines  string disciplines[] =  {  "Figure Skating",  "Speed Skating",  "Short track"  };  int gold\_medals[SPORTS] = {1, 0, 0};  int silver\_medals[SPORTS] = {1, 0, 0};  int bronze\_medals[SPORTS] = {1, 2, 1};  printTotalMedals(disciplines, gold\_medals, silver\_medals, bronze\_medals, 3); | Figure Skating: 3  Speed Skating: 2  Short track: 1 |
| const int SPORTS = 3; // number of sports/disciplines  string disciplines[] =  {  "Figure Skating",  "Speed Skating",  "Short track"  };  int gold\_medals[SPORTS] = {1, 0, 0};  int silver\_medals[SPORTS] = {1, 0, 0};  int bronze\_medals[SPORTS] = {1, 2, 1};  printTotalMedals(disciplines, gold\_medals, silver\_medals, bronze\_me |  |

anson, PSYC 1001

### Question 9

Not complete

Marked out of 1.00

Flag question

#### Question text

Write a function **similarity** that takes in 2 arrays of **int** type and their sizes and returns **true** if the two arrays are identical and returns **false** if two arrays are not same.

**bool similarity(int arr1[ ], int size1, int arr2[], int size2)**

Example:

arr1= {1, 4, 6, 7, 23, 3}

arr2 = {1, 4, 6, 7, 23, 3}

return value: **true**

arr1= {1, 4, 6, 7, 23, 3}

arr2 = {1, 14, 6, 7, 23, 3}

return value: **false**

arr1= {1, 4, 6, 7, 23, 3}

arr2 = {1, 17, 3}

return value: **false**

### Question 10

Not complete

Marked out of 1.00

Flag question

#### Question text

Write a function **similarity** that takes in two 2-D arrays of **int** type and their row sizes and returns **true** if the two arrays are identical and returns **false** if two arrays are not same. (Assume column size is always 3)

**bool similarity(int arr1[][3], int row1, int arr2[][3], int row2)**

Example:

arr1[3][3]= {{1, 4, 6}, {7, 23, 3}, {5, 15, 25}}

arr1[3][3]= {{1, 4, 6}, {7, 23, 3}, {5, 15, 25}}

return value: **true**

arr1[3][3]= {{1, 4, 6}, {17, 23, 3}, {5, 15, 25}}

arr2[3][3] = {{1, 4, 6}, {7, 23, 3}, {5, 75, 25}}

return value: **false**

arr1[2][3]= {{1, 4, 6}, {17, 23, 3}}

arr2[3][3] = {{1, 4, 6}, {17, 23, 3}, {5, 15, 25}}

return value: **false**

### Question 11

Not complete

Marked out of 1.00

Flag question

#### Question text

Create a function named**writeCountryDensity**, which takes a string input ***filename*** and an integer ***maximum density*** value and a string ***output file name*** as parameters.  The function will read the country name, population, and area from the file and output the ***names of the country*** with the ***density*** less than or equal to the value given.  This function returns the integer number of entries wrote in the output file.  If the input file cannot be opened, return -1 and do not print anything to the file.

Read each line from the given filename, parse the data, process the data, and write the required information into the output file.

Each line of the input file contains <COUNTRY NAME>, <POPULATION>,  <AREA>

Each line of the output file should contain <COUNTRY NAME>, <DENSITY VALUE>

**You only need to write the function code.**

HINT: We have provided a function that may make the parsing easier:

**int split(string s, char sep, string words[], int max\_words);**

**Example:** For Input File

Panama, 3292693, 78200

Papua New Guinea, 5931769, 462840

Paraguay, 6831306, 406750

Peru, 29180899, 1285220  
  
  
Output File should contain  
  
Panama, 42.106  
Papua New Guinea, 12.816  
Paraguay, 16.7949  
Peru, 22.705

### Question 12

Not complete

Marked out of 1.00

Flag question

#### Question text

Write a function named**writeVowels**, which takes three inputs : a string filename, an array of strings and an integer depicting the length of this array as parameters.  The function will read the strings from the array and output to the file the number of vowels contained in each string along-with the string itself. The format is shown below. The function should skip writing for empty strings and should return the total number of lines that were written to the file.

Eg : string words[] = {"hello","", "WORLD"}

Output format in the file:  
hello,2  
WORLD,1

The total number of lines written to the file would be 2 and thus your function should also return that value.

**For example:**

| **Test** | **Result** |
| --- | --- |
| string lines[] = {"hello","","WORLD"};  int writeCount = writeVowels("test1.txt", lines, 3);  cout<<"Number of lines written: "<<writeCount<<endl;  displayFile("test1.txt"); | Number of lines written: 2  hello,2  WO |

Monday,52

Note: The function, int Split(string s, char sep, string words[], int max\_words) is provided; your function may call it. It behaves the same way as split from Homework 6 and Recitation.

Example

|  |
| --- |
| **File Contents** |
| Monday,52 Tuesday,55 Wednesday,50 Thursday,45 Friday,39 Saturday,20 Sunday,35 |
| **Return Value**    42.29 |