**Test Script**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Function** | **#** | **Description** | **Sample Input Data** | | | | | | | **Expected Output** | **Actual Output** | **P/F** |
| **inputCheck** | **1** | The total values of the array to be checked is already 0 | **aTest contains:** 0, 0, 0, 0, 0, 0 | | | | | | | The function will return an integer 1, for the sum is equal to 0. | The function returns 1 | **P** |
|  | **2** | The values stored in the array contains decimal numbers | **aTest contains:** 2.3, 54.07, 28.96, 0.0, 9.8, 6.9 | | | | | | | The function will return an integer 0, for the total of the values in the array is not equal to zero | The function returns 0 | **P** |
|  | **3** | The values stored in the array contains a mix of negative and positive numbers. | **aTest contains:** 25, -78, 234, -29, 23, -58 | | | | | | | The function will return an integer 0, for the total of the values in the array is not equal to zero | The function returns 0 | **P** |
| **arrCheck** | **1** | The total values of the array to be checked is already 0 | **aTest contains:** 0, 0, 0, 0, 0, 0 | | | | | | | The function will return an integer 1, for the sum is equal to 0 | The function returns 1 | **P** |
|  | **2** | The values stored in the array contains decimal numbers | **aTest contains:** 3.54, 78.09, 6.76, 0.01, 2.3, 7.85 | | | | | | | The function will return an integer 0, for the total of the values in the array is not equal to zero | The function returns 0 | **P** |
|  | **3** | The values stored in the array contains a mix of negative and positive numbers. | **aTest contains:** 3, 32, -67, 89, 75, -1 | | | | | | | The function will return an integer 0, for the total of the values in the array is not equal to zero | The function returns 0 | **P** |
| **displayTable** | **1** | The parameters passed within the function is already in order | **Parameters passed:**  aSpinach, aRomaineLettuce, aMustardGreen, aCarrot, aPotato, aYam, aAsparagus, aBroccoli, aMelon, aStrawberry | | | | | | | The function will display a table containing the nutrient values of the Fruits /Vegetables in order by food type. | The function displays a table containing the nutrient values of the Fruits /Vegetables in order by food type. | **P** |
|  | **2** | The parameters passed within the function is in reverse order | **Parameters passed**:  aStrawberry, aMelon, aBroccoli, aAsparagus, aYam, aPotato, aCarrot, aMustardGreen, aRomaineLettuce, aSpinach | | | | | | | The function will still display a table containing the nutrient values of the Fruits /Vegetables in order by food type. | The function displays a table containing the nutrient values of the Fruits /Vegetables in order by food type. | **P** |
|  | **3** | The parameters passed within the function is in no particular order | **Parameters passed:** aStrawberry, aRomainelettuce, aBroccoli, aPotato, aSpinach, aYam, aMelon, aCarrot, aMustardGreen, aAsparagus | | | | | | | The function will still display a table containing the nutrient values of the Fruits /Vegetables in order by food type. | The function displays a table containing the nutrient values of the Fruits /Vegetables in order by food type. | **P** |
| **getServing** | **1** | The values passed are both integers | **fAmt** = 65(g)  **fServing** = 30(g) | | | | | | | 21.67 serving(s) | 21.67 serving(s) | **P** |
|  | **2** | The values passed are both floats | **fAmt** = 40.5(g)  **fServing** = 47.0(g) | | | | | | | 0.86 serving(s) | 0.86 serving(s) | **P** |
|  | **3** | The values passed is an integer and a float | **fAmt** = 125.5(g)  **fServing** = 72(g) | | | | | | | 1.74 serving(s) | 1.74 serving(s) | **P** |
| **computeServing** | **1** | The amount in grams entered is an integer | **fAmt** = 85(g)  **fServing** = 56(g) | | | | | | | 1.52 serving(s) | 1.52 serving(s) | **P** |
|  | **2** | The amount in grams entered is a float | **fAmt** = 90.05(g)  **fServing** = 30(g) | | | | | | | 3.00 serving(s) | 3.00 serving(s) | **P** |
|  | **3** | The amount in grams entered is a negative number | **fAmt** = -79(g)  **fServing** = 47(g) | | | | | | | The function will display an error containing a message saying “Please enter a valid amount in grams” | The function will display an error containing a message saying “Please enter a valid amount in grams” | **P** |
| **getCooked** | **1** | All of the values passed are floats | **fGrams** = 256.5(g)  **fServing** = 56.0(g)  **fNutrient** = 17.7(mg)  **fNutriLost** = 0.60 | | | | | | | 48.64(mg) | 48.64(mg) | **P** |
|  | **2** | The values passed are a mix of integers and floats | **fGrams** = 375 (g)  **fServing** = 47(g)  **fNutrient** = 0.46(mg)  **fNutriLost** = 0.90 | | | | | | | 3.30(mg) | 3.30(mg) | **P** |
|  | **3** | A base value is passed as the input (fGrams) | **fGrams** = 30(g)  **fServing** = 30(g)  **fNutrient** = 2813(IU)  **fNutriLost** = 0.95 | | | | | | | 2672.35(IU) | 2672.35(IU) | **P** |
| **getRaw** | **1** | All of the values passed are integers | **fGrams** = 235(g)  **fServing** = 138(g)  **fNutrient** = 52(mcg) | | | | | | | 88.55(mcg) | 88.55(mcg) | **P** |
|  | **2** | All of the values passed are floats | **fGrams** = 79.5(g)  **fServing** = 30.0(g)  **fNutrient** = 0.81(mg) | | | | | | | 2.15(mg) | 2.15(mg) | **P** |
|  | **3** | The values passed are a mix of integers and floats | **fGrams** = 115(g)  **fServing** = 64(g)  **fNutrient** = 1.37(mg) | | | | | | | 2.46(mg) | 2.46(mg) | **P** |
| **getNutriBase** | **1** | The parameters passed within the function is already in order | **Parameters passed:**  aSpinach, aRomaineLettuce, aMustardGreen, aCarrot, aPotato, aYam, aAsparagus, aBroccoli, aMelon, aStrawberry  **Sample selections:** | | | | | | | **aTotalBase contains:** 3.79, 0.83, 141.13, 12943.21, 283.15, 105.74, 2.64 | **aTotalBase contains:** 3.79, 0.83, 141.13, 12943.21, 283.15, 105.74, 2.64 | **P** |
| **Food** | | | | | **Grams** | |
| Spinach  Lettuce  Strawberry  Spinach | | | | | 45  30  125.5  65 | |
|  | **2** | The parameters passed within the function is in reverse order | **Parameters passed:**  aStrawberry, aMelon, aBroccoli, aAsparagus, aYam, aPotato, aCarrot, aMustardGreen, aRomaineLettuce, aSpinach  **Sample selections:** | | | | | | | **aTotalBase contains:** 3.87, 1.08, 157.47, 26545.89, 276.18, 85.75, 3.05 | **aTotalBase contains:** 3.87, 1.08, 157.47, 26545.89, 276.18, 85.75, 3.05 | **P** |
| **Food** | | | | **Grams** | | |
| Spinach  Carrot  Yam  Melon | | | | 120  67  34  120 | | |
|  | **3** | The parameters passed within the function is in no particular order | **Parameters passed**: aStrawberry, aRomainelettuce, aBroccoli, aPotato, aSpinach, aYam, aMelon, aCarrot, aMustardGreen, aAsparagus  **Sample selections:** | | | | | | | **aTotalBase contains:** 8.75, 2.43, 131.33, 8822.71, 320.54, 69.75, 4.35 | **aTotalBase contains:** 8.75, 2.43, 131.33, 8822.71, 320.54, 69.75, 4.35 | **P** |
| **Food** | | | **Grams** | | | |
| Potato  Broccoli  Lettuce  Asparagus | | | 57  45.5  68  345.15 | | | |
| **getPlan** | **1** | A float value is entered in the food selection | 1.0 | | | | | | | 1(Spinach) | 1(Spinach) | **P** |
|  | **2** | The character entered in the preparation selection is not within the choices (y, Y, n, N**)** | q | | | | | | | The function will display an error containing a message saying “Please input either Y or N only” | Please input either Y or N only | **P** |
|  | **3** | The amount in grams entered is a negative value | -67 | | | | | | | The function will display an error containing a message saying “Please enter a valid amount in grams” | Please enter a valid amount in grams | **P** |
| **getNutriLost** | **1** | All of the values passed are integers | **aTotalBase contains:** 11, 10, 9, 8, 7, 6  **aTotal contains:** 5, 4, 3, 2, 1, 0 | | | | | | | **aTotalLost contains:** 6, 6, 6, 6, 6, 6 | **aTotalLost contains**: 6, 6, 6, 6, 6, 6 | **P** |
|  | **2** | All of the values passed are floats | **aTotalBase contains:** 23.65, 46.6, 7.09, 2.0, 3.14, 56.15  **aTotal contains:** 8.05, 6.6, 5.4, 1.0, 0.78, 21.15 | | | | | | | **aTotalLost contains:** 15.60, 40.6, 1.69, 1.00, 2.36, 35.00 | **aTotalLost contains:** 15.60, 40.6, 1.69, 1.00, 2.36, 35.00 | **P** |
|  | **3** | The values passed are a mix of integers and floats | **aTotalBase contains:** 56, 55.15, 54.36, 53.01, 52, 51.24  **aTotal contains:** 28, 27.15, 26.30, 25, 24, 23.20 | | | | | | | **aTotalLost contains:** 28.00, 28.00, 28.06, 28.01, 28.00, 28.04 | **aTotalLost contains:** 28.00, 28.00, 28.06, 28.01, 28.00, 28.04 | **P** |
| **computeNutrients** | **1** | The parameters passed within the function is already in order | **Parameters passed:**  aSpinach, aRomaineLettuce, aMustardGreen, aCarrot, aPotato, aYam, aAsparagus, aBroccoli, aMelon, aStrawberry  **Sample selections:** | | | | | | | **aSpinach© contains**: 1.09, 0.22, 40.50, 4008.52, 56.55, 7.56, 0.92  **aRomaineLettuce contains:** 0.29, 0.07, 10.21, 2613.19, 40.85, 1.21, 0.04  **aStrawberry© contains**: 0.47, 0.16, 18.82, 14.12, 20.74, 51.61, 0.26  **aSpinach contains:** 1.76, 0.35, 65.00, 6094.83, 125.67, 18.20, 1.32 | **aSpinach© contains**: 1.09, 0.22, 40.50, 4008.52, 56.55, 7.56, 0.92  **aRomaineLettuce contains:** 0.29, 0.07, 10.21, 2613.19, 40.85, 1.21, 0.04  **aStrawberry© contains**: 0.47, 0.16, 18.82, 14.12, 20.74, 51.61, 0.26  **aSpinach contains:** 1.76, 0.35, 65.00, 6094.83, 125.67, 18.20, 1.32 | **P** |
| **Food** | | | | | | **Grams** |
| Spinach©  Lettuce  Strawberry©  Spinach | | | | | | 45  30  125.5  65 |
|  | **2** | The parameters passed within the function is in reverse order | **Parameters passed:**  aStrawberry, aMelon, aBroccoli, aAsparagus, aYam, aPotato, aCarrot, aMustardGreen, aRomaineLettuce, aSpinach  **Sample selection:** | | | | | | | **aSpinach contains:** 3.24, 0.64, 120.00, 11252.00, 232.00, 33.60, 2.44  **aCarrot© contains:**  0.18, 0.15, 19.77, 10074.05, 9.23, 2.77, 0.44  **aYam contains:**  0.17, 0.07, 5.00, 41.50, 5.50, 4.10, 0.12  **aMelon© contains:** 0.23, 0.19, 9.45, 3653.10, 17..85, 30.87, 0.04 | **aSpinach contains:** 3.24, 0.64, 120.00, 11252.00, 232.00, 33.60, 2.44  **aCarrot© contains:**  0.18, 0.15, 19.77, 10074.05, 9.23, 2.77, 0.44  **aYam contains:**  0.17, 0.07, 5.00, 41.50, 5.50, 4.10, 0.12  **aMelon© contains:** 0.23, 0.19, 9.45, 3653.10, 17..85, 30.87, 0.04 | **P** |
| **Food** | | **Grams** | | | | |
| Spinach  Carrot©  Yam  Melon© | | 120  67  34  120 | | | | |
|  | **3** | The parameters passed within the function is in no particular order | **Parameters passed:** aStrawberry, aRomainelettuce, aBroccoli, aPotato, aSpinach, aYam, aMelon, aCarrot, aMustardGreen, aAsparagus  **Sample selection:** | | | | | | | **aPotato© contains:** 0.33, 0.18, 5.20, 5.20, 15.03, 5.03, 0.02  **aBroccoli contains:** 0.33, 0.19, 21.50, 283.50, 28.50, 40.60, 0.35  **aRomaineLettuce©** contains: 0.60, 0.14, 20.83, 5627.07, 60.19, 1.65, 0.09  **aAsparagus contains:** 7.39, 1.89, 80.89, 2610.20, 177.97, 19.41, 3.88 | **aPotato© contains:** 0.33, 0.18, 5.20, 5.20, 15.03, 5.03, 0.02  **aBroccoli contains:** 0.33, 0.19, 21.50, 283.50, 28.50, 40.60, 0.35  **aRomaineLettuce©** contains: 0.60, 0.14, 20.83, 5627.07, 60.19, 1.65, 0.09  **aAsparagus contains:** 7.39, 1.89, 80.89, 2610.20, 177.97, 19.41, 3.88 | **P** |
| **Food** | **Grams** | | | | | |
| Potato©  Broccoli  Lettuce©  Asparagus | 57  45.5  68  345.15 | | | | | |
| **computeFeedback** | **1** | Calculated nutrient value of a particular food is lower than the recommended dosage | **Calculated amount of Iron =** 3.61(mg)  **Recommended dosage** = 17(mg) | | | | | | | Will display “Lacking” in the remarks and its corresponding advice on how to meet the recommended dosage | Lacking  /\*advisory message\*/ | **P** |
|  | **2** | Calculated nutrient value of a particular food is higher than the upper tolerable limit of the nutrient | **Calculated amount of Vitamin A** =12730.67(IU)  **Upper Tolerable Limit** **=** 2500(IU) | | | | | | | Will display “Too much” in the remarks and its corresponding advice on how to meet the recommended dosage | Too much  /\*advisory message\*/ | **P** |
|  | **3** | Calculated nutrient value of a particular food meets the recommended dosage | **Calculated amount of Vitamin C =** 78.58(mg)  **Recommended dosage =** 70(mg) | | | | | | | Will display “Good” in the remarks. | Good | **P** |
| **NewDay** | **1** | All of the variables of an integer array will be reset with an integer value | **aTest contains:** 1, 5, 23, 43, 237, 90  **Reset value** = 1 | | | | | | | **aTest contains:** 1, 1, 1, 1, 1, 1 | **aTest contains:** 1, 1, 1, 1, 1, 1 | **P** |
|  | **2** | All of the variables of a float array will be reset with a float value | **aTest contains:** 23.5, 67.01, 4.5, 7.89, 3.14, 1.02  **Reset value** = 1.00 | | | | | | | **aTest contains:** 1.00, 1.00, 1.00, 1.00, 1.00, 1.00 | **aTest contains:** 1.00, 1.00, 1.00, 1.00, 1.00, 1.00 | **P** |
|  | **3** | An array with integer and float variables will be reset with a float value | **aTest contains:** 1, 0, 23.149, 57, 2.16, 63  **Reset values** = 1.00 | | | | | | | **aTest contains:** 1.00, 1.00, 1.00, 1.00, 1.00, 1.00 | **aTest contains:** 1.00, 1.00, 1.00, 1.00, 1.00, 1.00 | **P** |