

Name:

This test worth 100 points in total. 1 hour and 30 minutes. Closed books; closed notes; No calculator is needed. There are 14 questions altogether.

- Please make sure to clearly specify your answer.
 - For each violated or missing punctuation or syntax, 0.25 POINTS WILL BE TAKEN OFF.
-

1. [3 pts] Define an array, named *weight*, to store (exactly) 25 floating point pay rates.

2. [3 pts] Initialize the array (named *primes*) *without using the size declaratory*; the values of the array would be: 2, 3, 5, 7, and 11.

3. [3 pts] Write an array initialization statement to have the following values in an array, named *numbers*, (note that you should not explicitly be writing zeros in your statement):

1	0
3	4
5	0

4. [3 pts] Define a two-dimensional array with 6 rows and 10 columns to maintain the sales (the price) of 10 items over 6 years. Define and use constants as necessary.

5. [3 pts] Consider the following values of an array, named *enrollment*. Predict the output.

40	15	60	77	65	22
----	----	----	----	----	----

cout << enrollment [1] << endl;.....

cout << enrollment [5] << endl;.....

enrollment [4] = enrollment [2] + enrollment [4];

cout << enrollment [4] << endl;.....

6. [3 pts] Consider the following values of array named *steps*.

17	13	19	43
11	26	24	63
90	67	51	48
73	70	65	72
55	67	47	34

cout << steps [1] [1] << endl;.....

cout << steps [3] [1] << endl;.....

cout << steps [2] [3] << endl;.....

7. [4 pts] When you apply the **Linear search** to find 40 in the array, what are the array elements that you compare the 40 with?

30	42	43	40	54	7	18	91	120
----	----	----	----	----	---	----	----	-----

1 st comparison	
2 nd comparison	
3 rd comparison	
4 th comparison	
5 th comparison	

8. [8 pts] When you apply the **Binary search** to find 45 in the array, what are the array elements that you compare the 45 with. Use the table to clearly show your work?

10	12	13	43	44	77	88	91	93	100	111	115
----	----	----	----	----	----	----	----	----	-----	-----	-----

1 st comparison					
2 nd comparison					
3 rd comparison					
4 th comparison					
5 th comparison					
6 th comparison					

9. [10 pts] Write a definition of a function that finds the **numeric average** of all the array elements (floats) in 1-dimensional array; name the function *findAverage()* ; return the result to the calling function; use parameters as needed.

Sample Test

10. [10 pts] Write a definition of a function that finds the **highest** and the **lowest** in a 1-dimensional array that maintains the total number of enrollments per semester. Return both results to the calling function. Name the function *findEnrolStats*; use parameters as needed.

Sample Test

11. [10 pts] Write a definition of a function that determines whether two 1-dimensional arrays are **equal**; assume they will be in the same size; return true if they are equal or false, otherwise. Name the function *isEqual*. Use parameters as needed.

Sample Test

12. [10 pts] Write a definition of a function that **resets the values** of a 2-dimensional array such that the values less than 50 will be set 0 and greater than or equal to 50 will be set to 1; 50 would be an argument to the function; name the function *reset*; use any other parameters as needed.

Sample Test

13. [10 pts] Write a function that takes a 2-dimensional array and **displays the content as given in the example**; Please note that the rows and columns are switched. Name the function: *transpose*. Select the parameters such that it could outputs the transpose of any sized 2-dimensional array. You may assume that the array entirely filled with numbers.

array

17	13	19	43
11	26	24	63
90	67	51	48
73	70	65	72
55	67	47	34

output

17	11	90	73	55
13	26	67	70	67
19	24	51	65	47
43	63	48	72	34

14. [20 pts] Write a **FULL** C++ program that reads a list of students' letter grades into an array; The grades would be read as key board inputs. Then the program should display the student letter grades, one per each line, in the reverse order of their entry. Define and use at least two functions: one to fill the array with user inputs (*readGrades()*) and another to display the letter grades in the reverse order (*displayRevGrades()*).

Sample Test

Sample Test