2020/03/31

Program Design
First Mid-Test - 2020

I. Matching: choose the correct meaning. (10%)

b 1. &	h 2. Dereferencing operator
a 3. Subscript	g 4. Scope
f 5. *sizeof* operator	a 6. Null character
j 7. Pass-by-value	e 8. Pointer-offset notation
c 9. Search key	i 10. Pass-by-reference

a) String termination character.
b) Unary operator that returns the address of its operand.
c) Value that the program attempts to find by searching.
d) Provides a called function with the ability to access the caller's data directly.
e) ptr + 3 (ptr is a pointer).
f) Can be used to determine the size of an array in bytes.
g) The portion of a program in which an identifier may be referenced.
h) *.
i) Refers to a particular location or element in an array.
j) Passes a copy of an argument's value to a function.

II. Close the sentences. (15%)

a) A function that calls itself either directly or indirectly (i.e., through another function) is a(n) _____ function.
b) The process of placing the elements of an array in order is called _____ the array.
c) A pointer is a variable that stores the _____ of another variable.
d) The _____ operator can be used to determine the size of an array in bytes.
e) For a recursion to terminate, it must reduce the original problem to the simplest form, called the _____.

III. The following **strcopy** function uses the C++ traditional **pointer-based** array declaration for its parameters, which copies the character string in **src[]** into the character array **dest[]**. Note: For all the following questions, do not worry about the size of the strings. (a:5%, b:10%, c:10 %, **total:25%**)

void strcopy(char dest[], const char src[])

a) Please implement the **strcopy** function using index operator [] on the array elements.
b) Please using just pointer notations to declare the parameters of the **strcopy** function and to copy the characters from **src** string to **dest** string.
c) Please overload the **strcopy** function with:

void strcopy(char dest[], const char src[], int n)

that copies the first **n** characters of string **src** *into the tail of* the string **dest**. Note: **n** might be larger than the size of string **src**. (hint: remember to terminate string **dest** after copy operations.)

IV. I have a course with a small group of 5 students. During the course time there are 5 tests with different weights for the final grade of the course. The following two tables list the weight of each tests and the names and tests results of the students. (a:5%, b:5%, c:10%, d:10%, e:10%. **total: 40%**)

Table 1.

Tests	Test1	Test2	Test3	Test4	Test5
weights	0.1	0.3	0.1	0.1	0.4

Table 2.

Names/Test Results	Test1	Test2	Test3	Test4	Test5
Bob	56	67	83	81	70
John	76	89	81	42	66
Joe	65	69	91	89	82
Mary	70	78	90	86	63
ShauMin	86	81	91	85	69

a) Declare a two-dimensional array **scores** and a one-dimensional array **weights,** initializing them with the test results and the weights of the tests.
b) Declare an array of pointers **names** and initialize it with the names of the students.
c) Define a function ***double studentGrade*** that receives **weights** array and 5 scores of one student (use a *one-dimensional array* and its size as parameters), then returns the final grade of the student.
d) Define another function ***void finalGrade*** that can receive arrays **scores**, **weights** and **names**, and the function pointer of ***studentGrade***, then print out the names and the final grades, got by calling ***studentGrade*** function pointer, of all the 5 students.
e) Define two C structures *TestWeight*, which includes *testname* and *weight* fields, and *StudentScores*, which includes *studentName* and *Scores[5]* fields. Declare proper structure variables and initialize them with data in Table 1 and Table 2.

V. An integer mathematic sequence is defined as:(10%)

f(1)=A
f(n)=f(n−1)+B

where one can define the value of A and B to set the start value of a sequence and the difference between two terms. Write a recursive function ***int mathSeq*** that returns the value of ***f(n)*** with A and B as parameters too.

VI.