1) (10 pts) DSN (Dynamic Memory Management in C)

Consider the following structures and the main function shown below:

```
typedef struct StringType {          typedef struct Employee {
    char *string;                        StringType *ename;
    int length;                          double salary;
} StringType;                         } Employee;

#include <string.h>
#include <stdio.h>
int main() {
  //array of employees' names
  char nameList[][50] = {"Adam", "Josh", "Kyle", "Ali", "Mohammed"};
  //array of salaries, where 15.80 is the salary of Adam, 13.50 is
  // the salary of Josh, etc.
  double salaries[5] = {15.80, 13.5, 20.9, 12.99, 10.5};
  Employee *employees = createEmployees(nameList, salaries, 5);
   // Other code here…
   return 0;
}
```

Write a function createEmployees() that takes the list of employees' names, list of their salaries, and length of the list (empCount) as the parameters, **and returns a pointer to a dynamically allocated array of Employee storing the relevant information for empCount employees**. The function dynamically allocates memory for empCount number of employees and assigns the name and salaries for each of them from the input parameters. During this process, the names are stored in the dynamically allocated memory of StringType, and also make sure you assign the length of the name appropriately. Your code should use exact amount of memory needed to store the corresponding names. You may assume no name is longer than 49 characters.

```
Employee* createEmployees(char names[][50], double *salaries, int empCount) {

    Employee *employees = malloc (empCount  * sizeof(Employee));  // 2 pts

    for (int i = 0; i < empCount; i++) {                          // 1 pt

            employees[i].ename = malloc(sizeof(StringType));     // 1 pt
            int length = strlen(names[i])+1;                     // 0 pts

            employees[i].ename->string = malloc(length * sizeof(char)); // 2 pts

            strcpy(employees[i].ename->string, names[i]);        // 1 pt

            employees[i].ename->length = length-1;               // 1 pt

            employees[i].salary = salaries[i];                   // 1 pt
        }
      return employees;                                          // 1 pt
}
```