QUIZ TIME!!!

Structures

Will the following code work?

```
#include<stdio.h>
typedef struct stdt{
    char name[7];
    int roll;
} student;
void main(){
    student s1,s2 = {"Lianne",20};
    struct student s3;
    struct stdt s4;
    scanf("%s%d", &s1.name, s1->roll);
    scanf("%s%d", s1.name, s1->roll);
    if(s1==s2) printf("Equal \n"); /*Duplicate entry*/
    s1.name = "Clarissa"; /*Change the name to Clarissa*/
}
```

Pointers

Will the following code work?

```
#include<stdio.h>
void swap(int *x, int *y){
/* For exchanging the values of two variables*/
    int *tmp;
    tmp = x; x = y; tmp = y;
}

void main(){
    int x=20,y=-10;
    swap(x,y);
}
```

Dynamic Memory Allocation

Complete the program by filling up the missing bodies of the functions

```
typedef struct pt{
    /*Define a structure to hold x and y coords of a 2D point*/
    int x,y;} point;

int allocate(point **mat){
    /*Dynamically allocate space for storing a 10×10 array of
points and return 1 on success, else 0*/
}

void read(point **mat){
    /*Read the values of the 10×10 array of points*/
}

void print-special(point **mat){
    /*Print the points whose x and y coordinates are equal*/
}
```

Structures

Structure Definition

Define a data type vector which consists of a dynamically allocated array to store the elements of a vector and an integer to designate the size (dimension) of the vector. Assume that we deal with vectors of floating-point numbers.

Structures

Structure Definition

Define a data type vector which consists of a dynamically allocated array to store the elements of a vector and an integer to designate the size (dimension) of the vector. Assume that we deal with vectors of floating-point numbers.

Write a function that

For two input vectors $v = (v_0, v_1, \ldots, v_{n-1})$ and $w = (w_0, w_1, \ldots, w_{n-1})$, returns the dot product $v \cdot w = v_0 w_0 + v_1 w_1 + \ldots + v_{n-1} w_{n-1}$. If the input vectors are not of the same dimension, your function should return some appropriate error value.

2D Array

Write a C function which

Takes a square matrix A and its order n as two parameters and prints each of the n left diagonals of A starting from the top and ending at the bottom. The diagonals are treated in a wrap-around fashion as shown in the following 4 x 4 matrix.

$$M = \begin{bmatrix} X & + & - & * \\ * & X & + & - \\ - & * & X & + \\ + & - & * & X \end{bmatrix}$$

Malloc

Write a C function which

Write a function that accepts as argument an array A of integers together with its size n and a non-negative integer k. The function should return another array, allocated dynamically within the function, which is obtained by cyclically shifting the input array A by k positions to the right. For example, upon the input of $A = \{2, 4, 6, 1, 3, 9, 5\}$ of size $n = 7$ and $k = 3$, your function should return $\{3, 9, 5, 2, 4, 6, 1\}$.