

Name,Surname:

CMPE108 Group:

Signature

Please print this homework, and solve all questions on the printout using a pencil or pen. Do not forget to write your student number on every page. Late submissions, printout solutions, and photocopies will not be graded. The following information will be available on final exam sheets.

Precedence and Associativity Table

Operators	Associativity
() [] -> .	Left to right
! ++ -- + - * & (type)	Right to left (Unary)
* / %	Left to right
+ -	Left to right
< <= > >=	Left to right
== !=	Left to right
&&	Left to right
	Left to right
? :	Right to left (ternary)
= += -= *= /= %=	Right to left
,	Left to right

% specifiers in ANSI C:

- | | |
|---|--|
| %c char single character | %o int unsigned octal value |
| %d (%i) int signed integer | %p pointer address stored in pointer |
| %e (%E) float or double exponential format | %s array of char sequence of characters |
| %f float or double signed decimal | %u int unsigned decimal |
| %g (%G) float or double use %f or %e as required | %x (%X) int unsigned hex value |

String Control Codes:

- | | | | | |
|---------------------------|-------------------------|---------------------|----------------------|----------------------------|
| \a alert, | \b backspace, | \f formfeed, | \n new line, | \r carriage return, |
| \t horizontal tab, | \' single quote, | \0 null. | \\ backslash. | |

Q1.

Given that `int i;` is already declared at the beginning of the `main` function.

Write necessary C statements for the following descriptions.

If necessary, write declarations and initializations in your solutions.

- a) Declare a one-dimensional integer array `arr` of length 5 and initialize all elements to zero.

.....

- b) Assign 2 and 4 to the first and last element of the array `arr` in part (a) respectively, and display these two elements on the screen.

.....

- c) For the array `arr` defined in part (a), write a for-loop to enter new values from the keyboard to the array `arr`.

.....

- d) Write a for-loop to print elements of the array `arr` as a single column on the screen.

.....

- e) Declare a one-dimensional `int` array `arr1` of length 5 and copy all elements of `arr` to `arr1`.

.....

Q2.

Consider a program that consists of a function named `incircle` that will take two `doubles` as input formal parameters. This function should decide whether the point (x, y) lies inside the unit circle (the circle with center at the origin and radius 1), that is, if $(x^2 + y^2) < 1$. In case the point is inside the unit circle, return 1, and otherwise return 0.

The main program reads points (x,y) , and for each point it writes either "Point is inside the circle" or "Point is outside the circle"

Fill the empty parts of the program specified with dots (.....) below:

```
#include <stdio.h>
/* function definition below */
int incircle(.....) {    //a

    if (.....)          //b

        return .....;    //c
    else
        return .....;    //d
}
int main() {
    float x, y;
    while (1) {
        scanf("%f%f", &x, &y);
        /* call the function here to test the point*/

        if (.....)      //e
            printf("Point is inside the circle.\n");
        else
            printf("Point is outside the circle.\n");
    }
    return 0;
}
```

Q3.

The aim of the following C program is to initialize 4x4 matrix, change the diagonal elements in the matrix to 20, and display the average of all elements on the first and last columns of the matrix. Please, complete the missing parts in the following C program by considering both the comments given in the program, and the aim.

Note that: In the program, the array "A" must be used as your 4x4 matrix; you must NOT use additional variables. and you must fill only the missing parts.

```
#include <stdio.h>
#define L 4
int main()
{
    // In the following statement the array "A" is declared as 4x4 matrix
    // and initialized as 4x4 matrix
    int A[L][L] = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16};

    // Variables are declared and one of them is also initialized
    int i;
    int j;
    int sum=0;

    // The following loop assigns 20 to the diagonal elements.
    for(i=0; i<L; i++)

    // The following nested loops find the sum of all elements //a
    // on the first and last columns of the array "A".
    for(i=0; i<L; i++)

        for(j=0; j<L; _____ ) //b

            sum = _____; //c
    // In the following statement, the average will be displayed.
    printf("Average of the first and last columns is %f\n,
           (float) sum / 8 );
    return 0;
}
```


Q9.

The dot product of two vectors $\mathbf{a} = [a_1, a_2, \dots, a_n]$ and $\mathbf{b} = [b_1, b_2, \dots, b_n]$ is defined as.

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$$

where \sum denotes summation of all terms, and n is the dimension of the vector space. For instance, in three-dimensional space, the dot product of vectors $\mathbf{a}=[1, 2, -6]$ and $\mathbf{b}=[4, 2, -1]$ is:

$$[1,2,-6] \cdot [4,2,-1] = (1)(4) + (2)(2) + (-6)(-1) = 14$$

Fill in the missing parts in the following program such :

- that the user will enter two 5 dimensional vectors \mathbf{a} and \mathbf{b} into the arrays A and B respectively
- the dot product of the two vectors will be calculated and printed as the output.

Note: Use only the spaces provided. Do not add any extra lines or delete any given code.

```
#include <stdio.h>
int main (void)
{
    int i, sum=0;
    /*declare your arrays A and B below*/

    int _____ //a

    /* Use the loop below to enter elements of the two arrays */
    for ( _____ ) //b
        _____ ; //c

    /* Use the loop below to calculate the dot product here */
    for ( _____ ) //d
        _____ ; //e

    /* print the dot product below */

    _____ ; //f

    return 0;
}
```