COMPUTER PROGRAMMING I

GROUP ( ) LAB (5)

BY

NAME:

Student ID:

DATE:

LAB INSTRUCTION

- LAB Mode: **INDIVIDUAL**.
- Submitıon:-
  - Hard copy.
  - Softcopyas PDF format to Google Classroom.
  - ProgramıngCode.
- The font must be **ARIAL**andthe size must be **12. JUSTIFY. SPACING 1.5**.
- Use**YOUR NAME, SURNAME**, and**DATE**as a variable in youranswer.
- Complete thecoversheetandattach it toyourlab as (firstpage).
- Answerlabwithownwork (***NO PLAGIARISM***).
- Yourmarkswill be deducted in thecase of:
  - Latesubmission.
  - Plagiarism.

LAB QUESTIONS

1. Answer each of the following. Assume that single-precision floating-point numbers are stored in 4 bytes, and that the starting address of the array is at location 1002500 in memory. Each part of the exercise should use the results of previous parts where appropriate.

a) Define an array of type float called numbers with 10 elements, and initialize the elements

to the values 0.0, 1.1, 2.2, …, 9.9. Assume the symbolic constant SIZE has been defined as 10.

b) Define a pointer, nPtr, that points to an object of type float.

c) Print the elements of array numbers using array index notation. Use a for statement. Print each number with 1 position of precision to the right of the decimal point.

d) Give two separate statements that assign the starting address of array numbers to the pointer variable nPtr.

e) Print the elements of array numbers using pointer/offset notation with the pointer nPtr.

f) Print the elements of array numbers using pointer/offset notation with the array name as the pointer.

g) Print the elements of array numbers by indexing pointer nPtr.

h) Assuming that nPtr points to the beginning of array numbers, what address is referenced by nPtr + 8? What value is stored at that location?

i) Refer to element 4 of array numbers using array index notation, pointer/offset notation with the array name as the pointer, pointer index notation with nPtr and pointer/offset notation with nPtr.

2. Find the error in each of the following program segments. Assume

```
int *zPtr; // zPtr will reference array z
int *aPtr = NULL;
void *sPtr = NULL;
int number;
int z[5] = {1, 2, 3, 4, 5};
sPtr = z;
```

a) ++zptr;

b) // use pointer to get first value of array; assume zPtr is initialized

number = zPtr;

c) // assign array element 2 (the value 3) to number;

assume zPtr is initialized

d) // print entire array z; assume zPtr is initialized

```
for (size_t i = 0; i <= 5; ++i) {
printf("%d ", zPtr[i]);
}
```