

Student ID:

Full Student Name: Signature:

(XMUT) NWEN241 Quiz
System Programming
Wednesday, 14 October 2026

Time: 60 minutes

Type: Closed Book

Instruction:

- This Quiz will not contribute to your final score
- Analogue calculators and dictionaries are permitted.
- Write all answers clearly in the spaces provided
- If there is a question that is unclear, ask for clarification
- A spare page for extra answers is provided
- You may write notes and work on the paper, but make sure your answers are clear.

Part 1 - True or False

Circle the answer

1. The size of a pointer variable depends on the data type it points to (for example, an int* is larger than a char*)

TRUE FALSE

2. Dereferencing a NULL pointer is safe and simply returns 0.

TRUE FALSE

3. When you perform pointer arithmetic, such as ptr++, the memory address increases by the size of the data type the pointer points to.

TRUE FALSE

4. An uninitialized pointer contains a 'garbage' value that can point to arbitrary locations in memory

TRUE FALSE

5. In C, the name of an array can be used as a pointer to its first element.

TRUE FALSE

Student ID:

Part 2 – Simple Answers

1. What is the result of this code? Explain your answer

```
int main() {  
    int arr[] = {10, 20, 30, 40};  
    int *p = arr;  
    p++;  
    printf("%d", *p);  
    return 0;  
}
```



2. What is the result of this code? Explain your answer

```
int main() {  
    int a = 10, b = 20;  
    int *p = &a;  
    int *q = &b;  
    *p = *q;  
    printf("%d, %d", a, b);  
    return 0;  
}
```



Student ID:

3. What is the result of this code? Explain your answer

```
int main() {  
    int val = 15;  
    int *ptr = &val;  
    *ptr = *ptr + 5;  
    printf("%d", val);  
    return 0;  
}
```

Student ID:

4. Consider this C program

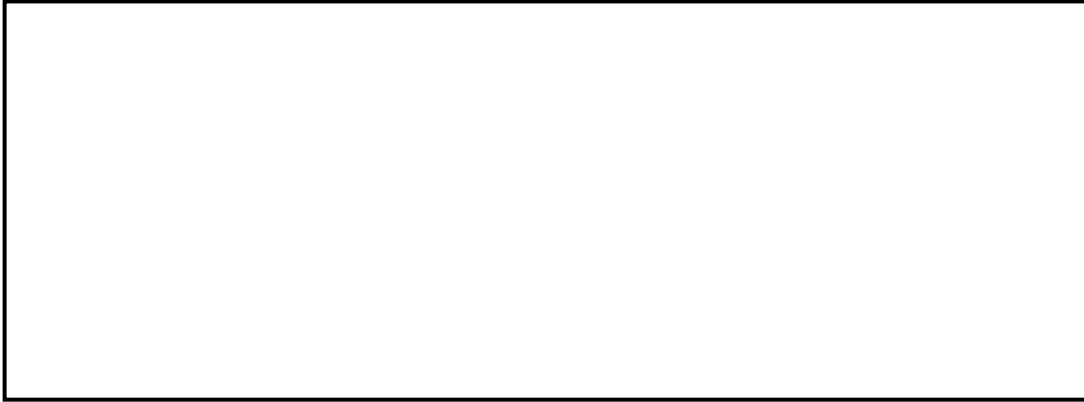
```
#include <stdio.h>
int main() {
    int a[] = {10, 20, 30, 40, 50};
    int len = sizeof(a) / sizeof(int);
    int sum = 0;
    for (int *ip = a; ip < a + len; ip++) {
        *ip = *ip + 5;      // Line 1
        sum = sum + *ip;    // Line 2
    }
    printf("%d, %d", a[2], sum);
    return 0;
}
```

- a. What is the result of the code? Trace each iteration of the loop

--

Student ID:

b. Why is the condition `ip < a + len` used instead of `ip <= a + len`?

A large, empty rectangular box with a black border, intended for the student to write their answer to the question.

Student ID:

5. Consider this C program

```
int x[5];
int *sp;
int **spp;
for (int i = 0; i < 5; i++) {
    x[i] = 100 - (i * 10);
}
sp = x;
spp = &sp;
```

- a) What is the value of *sp? Explain your answer
- b) What is the value of *(sp + 2)? Explain your answer
- c) What is the value of **spp? Explain your answer
- d) What is the value of *(*spp + 4)? Explain your answer

Student ID:

6. Consider the C program below.

```
#include <stdio.h>
void processArray(int *arr, int n, int factor) {
    for (int i = 0; i < n; i++) {
        if (i % 2 == 0) {
            *(arr + i) += factor;
        } else {
            *(arr + i) -= factor;
        }
    }
}
int main() {
    int myData[] = {10, 20, 30, 40, 50};
    int size = 5;
    int k = 5;
    processArray(myData, size, k);
    printf("%d, %d, %d", myData[0], myData[2], myData[4]);
    return 0;
}
```

What is the result? Trace the execution of the code carefully. Assume standard C compiler behaviour.