Name	Adrian Leo Pradana
NPM	2106718344

1. A pointer is a variable whose value is the memory address of another variable. A pointer variable is created with the * operator and points to a data type such as int of the same type. The pointer is given the address of the variable that the user working with. The pointer have a specific purpose in C because some C programming tasks are performed more easily with pointers, and other tasks, such as dynamic memory allocation, cannot be performed without using pointers.

   References:

   - Fundamentals of Digital Logic with Verilog Design: Third Edition Page 203-211

2. In C, string is a sequence of characters which is save in an array and as we know ‘\0’ null character marks the end of a string. Example of string pointer :

```
// pointer variable to store string
 char *strPtr = "test";

 // temporary pointer variable
 char *temp = strPtr;

 // print the string
 while(*temp != '\0') {
  printf("%c", *temp);

  // move the temp pointer to the next memory location
  temp++;
 }
```

Value	t	e	s	t	\0
Address	1000	1001	1002	1003	1004

From the above code that means the string “test” is saved in the memory location 1000 to 1004. That’s why to print the value of string with pointer we need using loops to move the pointer until it met ‘\0’ (NULL).

References:

- https://www.geeksforgeeks.org/storage-for-strings-in-c/

3. For `int * const ptr1;` it means ptr1 const pointer to int, this implies that the pointer shouldn’t point to some other address. Const qualifier doesn’t affect the value of integer in this scenario so the value being stored in the address is allowed to change. while `const int * ptr2;` it means ptr2 pointer to int const , this implies that the pointer is pointing to a value that shouldn’t be changed. Const qualifier doesn’t affect the pointer in this scenario so the pointer is allowed to point to some other address.

   References:

   - https://developerinsider.co/what-is-the-difference-between-const-int-and-int-const/

4. The output is 6487568, we knew that *p is null before and we assign 10 to x and we assign p variable with reference x variable and it goes to the variable that appoints to it and get the address of its variable.

   References:

   - https://developerinsider.co/what-is-the-difference-between-const-int-and-int-const/

5.
```
#include <stdio.h>

void multiply(int *x , int*y){
        int result;

        result = *x * *y;
        printf("result is : %d ",result);
}

int main(void) {

        int a, b;
```

```
        printf("Input first number :");
        scanf("%d", &a);
        printf("Input second number : ");
        scanf("%d", &b);

        printf("%d X %d \n", a , b);

        multiply(&a,&b);

    return 0;
}
```

References:

- https://www.geeksforgeeks.org/difference-between-call-by-value-and-call-by-reference/