Pointer Exercises

Section A: Multiple Choice Questions (MCQs)

1. What will be the output of the following code?

   ```
   int x = 5;
   int *ptr = &x;
   *ptr += 2;
   printf("%d", x);
   ```

   a) 5
   b) 7
   c) Memory address of x
   d) Undefined behavior

2. Which of the following statements is **true** about pointers in C?
   a) A pointer can store both address and value at the same time.
   b) The `&` operator is used to dereference a pointer.
   c) `*ptr` accesses the value stored at the address `ptr` is pointing to.
   d) A pointer can point only to `int` variables.

3. Given:

   ```
   int arr[] = {1, 2, 3};
   int *p = arr;
   p++;
   ```

   What does `p` now point to?
   a) arr[0]
   b) arr[1]
   c) arr[2]
   d) Memory address after arr[2]

4. Which of the following can cause **undefined behavior** in C?
   a) Dereferencing an uninitialized pointer
   b) Assigning NULL to a pointer
   c) Incrementing a pointer to an array element
   d) Passing a pointer to a function

5. If `int *p;` is declared but not initialized, what will happen if you do `printf("%d", *p);`?
   a) Prints 0
   b) Prints garbage value
   c) Causes undefined behavior
   d) Compilation error

Section B: Debug This Code

1. The following code intends to swap two integers using pointers, but it is **not working correctly**. Identify and fix the errors.

```
#include <stdio.h>
void swap(int a, int b) {
    int temp = a;
    a = b;
    b = temp;
}
int main() {
    int x = 5, y = 10;
    swap(&x, &y);
    printf("x = %d, y = %d", x, y);
    return 0;
}
```

2. This program is intended to print the sum of all array elements using pointers. Fix the logic and syntax errors.

```
#include <stdio.h>
int main() {
    int arr[5] = {1, 2, 3, 4, 5};
    int *p = &arr;
    int sum;
    for(int i = 0; i <= 5; i++) {
        sum += *(p + i);
    }
    printf("Sum = %d", sum);
    return 0;
}
```

Section C: Predict the Output

Question 1

```
int x = 10, y = 20;
int *p1 = &x, *p2 = &y;
*p1 = *p2;
*p2 = 50;
printf("%d %d", x, y);
```

Question 2

```
char str[] = "Hello";
char *p = str;
p += 2;
printf("%s", p);
```

Question 3

```
int num = 5;
int *ptr = &num;
printf("%p %d", ptr, *ptr);
```

Section D: Programming Challenges

1. **Pointer-based String Reverse**
   Write a program that takes a string as input and reverses it **using pointers only** (no array indexing allowed).

2. **Dynamic Array Sum using malloc**
   Write a program that:

   - Dynamically allocates memory for n integers
   - Takes n integers as input
   - Calculates and prints the sum using pointer arithmetic
   - Frees the allocated memory

Section E: Real-World Scenario

Hospital Bed Allocation System

A hospital has a fixed number of beds. Each bed stores a patient's ID (integer).
Write a program that:

- Uses a pointer to store patient IDs in an array
- Allows the user to assign IDs to beds
- Prints all occupied beds and their patient IDs using pointer traversal

---