Chapter 9 – Pointers

Short Questions with Easy Answers

Q1. What is a Pointer?

👉 A pointer is a variable that stores the **address of another variable** instead of storing a value.

- Example:

```
int a = 10;
int *p = &a;   // p stores the address of a
```

Q2. What is a Void pointer?

👉 A pointer that does not have a specific data type is called a **void pointer**. It can point to any type.

- Example:

```
void *ptr;
int x = 5;
ptr = &x;
```

Q3. Write some valid pointer declarations.

👉 Examples:

```
int *p1;
char *p2;
double *p3;
float *p4;
```

Q4. What is a Null Pointer?

👉 A pointer that does not point anywhere (contains value 0).

```
int *ptr = NULL;
```

Q5. How to declare a function pointer?

👉 Syntax:

```
return_type (*ptr)(parameter_list);
```

Example:

```
double (*p2f)(double, char);
```

Q6. Logic of statement p = &val[0];

👉 Means pointer p points to the **first element of array val**.

- *p == val[0]
- *(p+1) == val[1]
- *(p+n) == val[n]

ExamChamber.com

Q7. What is call by reference in pointers?

👉 Passing **address of variable** to function. Any change inside function affects original variable.

Q8. What are pointer and character strings?

👉 A string can be stored in a char pointer.

```
char *str = "Hello";

printf("%s", str);
```

Q9. What is a double pointer?

👉 A pointer that stores the address of another pointer.

```
int x = 10;

int *p = &x;

int **dp = &p;
```

Q10. Write disadvantages of Pointers.

- Complex to understand
- Can cause memory corruption
- May lead to memory leakage
- Slower than normal variables

✅ **Long Questions with Simple Answers**

Q1. How pointers work in C? (With program)

👉 Example:

```
#include <stdio.h>

int main() {
  int x = 10;
  int *p;   // declare pointer
  p = &x;   // store address of x in p

  printf("Value of x = %d\n", x);
  printf("Address of x = %p\n", &x);
  printf("Pointer p stores = %p\n", p);
  printf("Value at address p = %d\n", *p);  // dereferencing

  return 0;
```

ExamChamber.com

```
}
```

Q2. Explain types of pointers with examples.

1. **Null Pointer** → Points nowhere.
2. int *p = NULL;
3. **Void Pointer** → Can point to any type.
4. void *p;
5. **Wild Pointer** → Uninitialized pointer.
6. int *p; // dangerous
7. **Dangling Pointer** → Points to a deleted memory location.
8. **Double Pointer** → Stores address of another pointer.

Q3. Program: Swap two numbers using pointers.

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

Q4. Advantages & Disadvantages of pointers.

✅ **Advantages:**

- Efficient in handling arrays, structures, strings
- Used in dynamic memory allocation (malloc, free)
- Useful in call by reference
- Used in system programming (OS, device drivers)

❌ **Disadvantages:**

- Complex to use
- Wrong usage causes memory leakage or crashes

ExamChamber.com

- Slower than normal variables

Q5. Write a note on Pointers and Strings.

👉 In C, strings are stored as **character arrays** and can be accessed using pointers.

- Example:

```
#include <stdio.h>
int main() {
  char *str = "Hello";
  printf("%s\n", str);
  printf("First char = %c\n", *str);
  printf("Second char = %c\n", *(str+1));
  return 0;
}
```

🔷 **Extra Practice Questions**

Q1. What is difference between pointer and array?

- Array → fixed size, stores multiple values.
- Pointer → stores address of a single variable but can traverse array.

Q2. Write a program to print array elements using pointer.

```
#include <stdio.h>
int main() {
  int arr[5] = {10, 20, 30, 40, 50};
  int *p = arr;

  for (int i = 0; i < 5; i++) {
    printf("%d ", *(p+i));
  }
  return 0;
}
```

Q3. Program: Use double pointer to access variable.

```
#include <stdio.h>
int main() {
  int x = 100;
```


ExamChamber.com

```
    int *p = &x;
    int **dp = &p;

    printf("Value of x = %d\n", **dp);
    return 0;
}
```

ExamChamber.com