

Pointers

16-11-25

Aim – Write a program to demonstrate pointers, pointer to pointer and pointer to array

Theory –

Pointers store memory addresses of variables, enabling efficient data access. A pointer to pointer holds the address of another pointer, allowing multiple indirection levels. A pointer to array stores the address of an entire array, helping manage multi-element data structures and multidimensional arrays effectively.

A1.

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

```
void main() {  
    int n = 10;  
    int *ptr;  
    ptr = &n;  
    printf("Value of n      : %d\n", n);  
    printf("Value via *ptr   : %d\n", *ptr);  
}
```

```
"C:\Users\tarun\Downloads\T" × + ∨  
Value of n      : 10  
Value via *ptr   : 10  
  
Process returned 23 (0x17)   execution time : 0.025 s  
Press any key to continue.
```

A2.

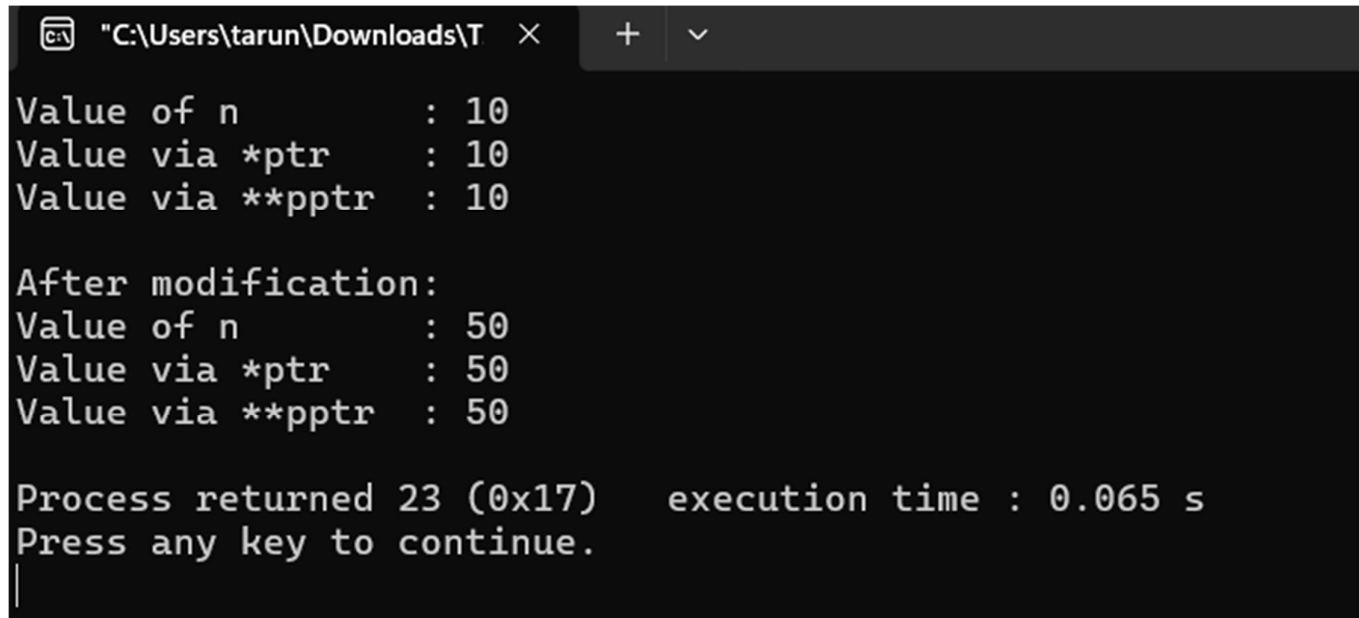
```
#include <stdio.h>
```

```
void main() {  
    int n = 10;  
    int *ptr;  
    int **pptr;  
    ptr = &n;  
    pptr = &ptr;  
    printf("Value of n      : %d\n", n);
```

```

printf("Value via *ptr   : %d\n", *ptr);
printf("Value via **pptr : %d\n", **pptr);
**pptr = 50;
printf("\nAfter modification:\n");
printf("Value of n      : %d\n", n);
printf("Value via *ptr   : %d\n", *ptr);
printf("Value via **pptr : %d\n", **pptr);
}

```



```

C:\Users\tarun\Downloads\T >
Value of n      : 10
Value via *ptr   : 10
Value via **pptr : 10

After modification:
Value of n      : 50
Value via *ptr   : 50
Value via **pptr : 50

Process returned 23 (0x17)   execution time : 0.065 s
Press any key to continue.
|

```

A3.

```

#include <stdio.h>

void main() {
    int arr[5] = {10, 20, 30, 40, 50};
    int (*p)[5] = &arr;
    printf("Accessing array using pointer to array:\n");
    for (int i = 0; i < 5; i++) {
        printf("(p)[%d] = %d\n", i, (*p)[i]);
    }
}

```

```
"C:\Users\tarun\Downloads\T  ×  +  ∨  
Accessing array using pointer to array:  
(*p)[0] = 10  
(*p)[1] = 20  
(*p)[2] = 30  
(*p)[3] = 40  
(*p)[4] = 50  
  
Process returned 13 (0xD)    execution time : 0.029 s  
Press any key to continue.
```

Conclusion

I learnt that pointers store variable addresses. Pointer to pointer stores another pointer's address for multi-level access. Pointer to array stores the address of an entire array, helping manage multiple elements efficiently.