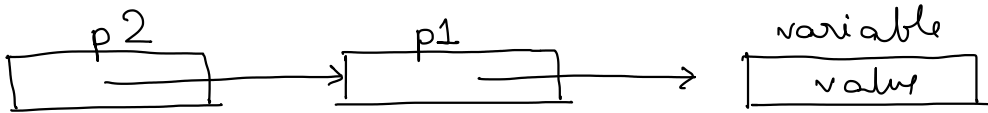


10d - Double Pointer

Friday, 14 August 2026

18:22

Pointer to pointer



```
int var = 100;  
int *p1 = &var;  
int **p2 = &p1;
```

Ways to access variable var

Print:

- i) var
- ii) *p1
- iii) **p2

pointers/double_pointers.c

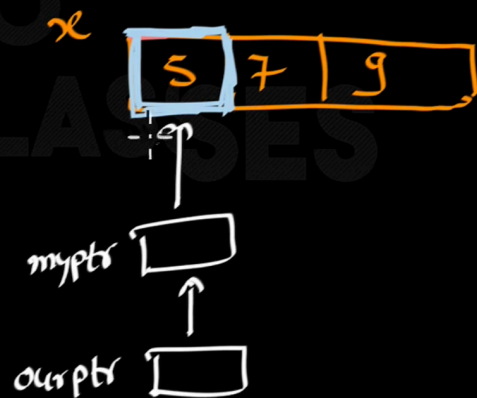
```
1  #include <stdio.h>  
2  
3  int main(){  
4  int var, *p1, **p2;  
5      var = 100;  
6      p1 = &var;  
7      p2 = &p1;  
8      printf("p1: %d\n", *p1);  
9      printf("p2: %d\n", **p2);  
10     return 0;  
11 }  
12
```

pointers - zsh

```
arnesh@Arneshs-MacBook-Air pointers % ./a.out
p1: 100
p2: 100
arnesh@Arneshs-MacBook-Air pointers %
```

Question :

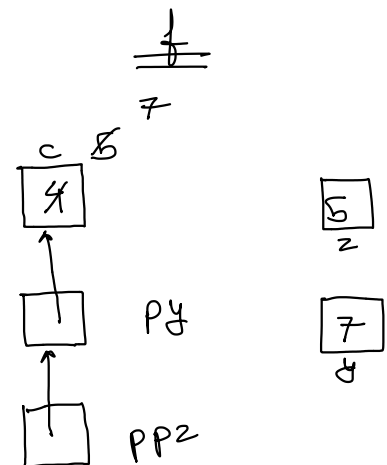
```
- int k, x[3] = {5, 7, 9};
- int *myptr, **ourptr;
- myptr = x;
- ourptr = &myptr;
- k = *myptr;           // k=? 5
- k = (**ourptr) + 1;  // k=? 6
- k = *(*ourptr + 1);  // k=? 7
```



GATE 2008

```
int f(int x, int *py, int **ppz)
{
    int y, z;
    **ppz += 1; z = **ppz;
    *py += 2; y = *py;
    x += 3;
    return x+y+z;
}

void main()
{
    int c, *b, **a;
    c = 4; b = &c; a = &b;
```



The above modify `c`, but `x` remains unchanged

```
printf("%d", f(c, b, a));
```

```
}
```

So $x += 3; \rightarrow 7$

o/p $\rightarrow 7 + 7 + 5$
 $\Rightarrow 19.$