Advance Pointers

Pointer Declarations

- Int *p;
- Int *p[];
- Int **p;
- Int (*)p[];

How Pointer Dereference ?

- Assume location 100
- Int a[]={2 , 4 , 6, 8 }
- Print a , *a
- Print &a[0]

- Int a[2][2] = {

  {2 , 4 },

  {6 , 8 }

  }

Print a , *a , **a

- Assume location 100
- Int a[2][2] = {

{2 , 4 },

{6 , 8 }

}

Print a , *a , **a

- Int a[2][2] = {

  {2 , 4 },

  {6 , 8 }

  }

Print a+1 , *a + 1 , **a + 1

- Int a[2][2] = {
  {2 , 4 },
  {6 , 8 }
  }
- Print a+1 , a[0] + 1 , a[0][0] + 1

Think This

- Assume location 100
- Int a[2][2] = {

  {2 , 4 },

  {6 , 8 }

  }
- Print &a+1

3-D Deferencing

- Int a[2][2][2] = {

```
                {
                    {2 , 3 }
                    { 4,  5 }
                },
               {    {6, 7 },
                    { 8, 9 }
               }
          }
```

- Print a , *a , **a , ***a

3D Dereferencing

- Int a[2][2][2] = {

```
{    2 , 3,
       4,  5
  },
{  6, 7 ,
     8, 9
}
}
```

- Print a +1 , *a +1 , **a +1 , ***a

- Int a[] = { 0 , 1 , 2 , 3 , 4 }
- Int *ptr;
- for (ptr = &a[0] ; p <= &a[4] ; p++)
  printf *ptr

Questions

```
static int arr[]={0,1,2,3,4};
int *p[]={arr,arr+1,arr+2,arr+3,arr+4};
int **ptr=p;
**ptr++;
print(ptr-p,*ptr-a,**ptr);
*++*ptr; //print same above
++**ptr; //print same above
```

- if array begins at 100 what is the output

```
int arr[]={ 1, 2,  3, 4 }
print(arr,&arr,arr+1,&arr+1)
```

- Are arr and &arr same for an array of 10 integers

How to Read Complex expression ?

- Rule 1. Assign the priority to the pointer declaration considering precedence and associative according to following table.

Operator	Precedance	Associative
( ) , [ ]	1	Left to right
* , Identifier	2	Right to left
Data type	3	

- (): This operator behaves as bracket operator or function operator.
- []: This operator behaves as array subscription operator.
- *: This operator behaves as pointer operator not as multiplication operator.
- Identifier: It is not an operator but it is name of pointer variable. You will always find the first priority will be assigned to the name of pointer.
- Data type: It is also not an operator. Data types also includes modifier (like signed int, long double etc.)

- char (* ptr)[3]
- float (* ptr)(int)
- void (*ptr)(int (*)[2],int (*) void))

- ptr is pointer to such function which first parameter is pointer to one dimensional array of size two which contentint type data and second parameter is pointer to such function which parameter is void and return type is int data type and return type is void.

Contact Info

- trainers@finaldesk.com
- rishabh@finaldesk.com
- nilesh@finaldesk.com
- jignesh@finaldesk.com
- yash@finaldesk.com
- anand@finaldesk.com