Complicated Statement Reading

- **int arr[4];**

  arr is a one dimensional array of 4 elements and each element is of type integer

- **float arr[8];**

  arr is a one dimensional array of 8 elements and each element is of type float

- **struct demo obj[5];**

  obj is a one dimensional array of 5 elements and each element is of type struct demo

- **int arr[3][5];**

  arr is a two dimensional array which contains 3 one dimensional array and each one dimensional array contains 5 elements and each element is of type integer.

- **double arr[4][2];**

  arr is a two dimensional array which contains 4 one dimensional array and each one dimensional array contains 2 elements and each element is of type double.

- **int arr[3][2][4];**

  arr is a three dimensional array which contains 3 two dimensional array and each
  two dimensional array contains 2 one dimensional array and each one dimensional
  array contains 4 elements and each element is of type integer.

- **char arr[6][4][8];**

  arr is a three dimensional array which contains 6 two dimensional array and each
  two dimensional array contains 4 one dimensional array and each one dimensional
  array contains 8 elements and each element is of type character.

- **char ** p1;**
  p1 is a pointer to pointer to char

- **const char **p2;**
  p2 is a pointer to pointer to const char

- **char * const * p3;**
  p3 is a pointer to const pointer to char

- **const char * const * p4;**
  p4 is a pointer to const pointer to const char

- **char ** const p5;**
  p5 is a const pointer to pointer to char

- **const char ** const p6;**
  p6 is a const pointer to pointer to const char

- **char * const * const p7;**
  p7 is a const pointer to const pointer to char

- **const char * const * const p8;**
  p8 is a const pointer to const pointer to const char

- **const int i=10;**
  i is a integer constant which contains value 10

- **const int *p = &i;**
  p is a pointer to integer constant

- **int const *p = &i;**
  p is a pointer to constant integer (same as above)
  ```
  p++;            //allowed
  (*p)++;         //error
  ```

- **int arr[6]={1,2,3};**
  arr is a array of six integers out of which three are initialized

- **int *p = arr;**

p is a pointer to integer which pointing at base address of array arr

```
p++;            //allowed
(*p)++;         //allowed
```

- **int const arr[3]={1,2,3};**
  arr is a array of three integer constant

- **int *p = arr;**
  p is a pointer to integer which pointing at base address of array arr

  ```
  p++;            //allowed
  (*p)++;         //allowed
  (arr[2])++;     //error
  ```

- **int const arr[3]={1,2,3};**
  arr is a array of three integer constant

- **int const * const p = arr;**
  p is a constant pointer to constant integer which sores address of array arr

  ```
  p++;            //allowed
  (*p)++;         //allowed
  (arr[2])++;     //error
  ```

- **int no=10;**
  no is a integer which stores value 10

- **int *p = &no;**
  p is a pointer to integer which stores address of no

- **int **s = &p;**
  s is a pointer to integer pointer which stores address of p

- **char c = 'A';**
  c is a character which stores character A

- **char *d= &c;**

d is a pointer to character which stores address of c

- **union demo**
- **{**

  int i;
  char c;
- **};**

  demo is union which contains one integer and one character

- **union demo obj;**

  obj is a object of union demo

- **union demo *g = &obj;**

  g is a pointer to union demo which stores address of object obj

- **struct demo**
- **{**

  int i;
  struct demo *next;
- **};**

  demo is a structure which contains a integer and self referencing pointer

- **struct demo obj;**

  obj is object of struct demo

- **struct demo *g = &obj;**

  g is a pointer to struct demo which stores address of obj

- **int *(arr[3]);**

  arr is a array three integer pointers

- **int (*arr)[3];**

  arr is a pointer to array of three integer

- **int (*(arr[3])[2])**

arr is a array of three pointers to array of two integers

- **void fun(void);**

  fun is a function which accept nothing and return nothing

- **int fun(int,char);**

  fun is a function which accept two parameters first is integer
  and second is character and function returns integer.

- **int* fun(int);**

  fun is a function which accept single parameter which is integer
  and it returns integer pointer(address of integer)

- **int fun(int *, char *)**

  fun is a function which accept two parameters first is integer pointer
  (address of integer) and second is character pointer(address of character) which return integer.

- **void fun(int a[2]);**

  fun is a function which accept address of integer array which contains 2 elements
  and function returns nothing.

- we can get same effect as above by writing function as
- **void fun(int a[ ]);**
- **void fun(int *a);**

- **int **fun(int);**

  fun is a function which accept integer and return pointer to integer pointer

- **void (*fptr)();**

  fptr is a pointer to function which accept nothing and return nothing

Complicated Statement Reading

- **int (*fptr)(int , char);**

  fptr is a pointer to function which accept two parameters first is integer and second is character and it return integer.

- **int *(*fptr)(int*,char*);**

  fptr is a function pointer which a accept two parameters first is integer pointer
  second in character pointer and function returns integer pointer.

- **int (*daytab)[13]**

  daytab is pointer to array of 13 integers.

- **void (*f[10]) (int, int)**

  f is an array of 10 of pointer to function which takes 2 arguments of type int
  returning void

- **char (*(*x())[]) ()**

  x is a function returning pointer to array of pointers to function returning char

- **char (*(*x[3])())[5]**

  x is an array of 3 pointers to function returning pointer to array of 5 char's

- **int *(*(*arr[5])()) ()**

  arr is an array of 5 pointers to functions returning pointer to function returning pointer to integer

- **void (*bsd_signal(int sig, void (*func)(int)))(int);**

  bsd_signal is a function that takes integer & a pointer to a function
  that takes integer as argument and returns void and returns pointer to
  a function(that take integer as argument and returns void)

- **float (*(*b())[])();**

b is a function that returns a pointer to an array of pointers
to functions returning floats.

- **void *(*c)(char,int(*)());**
  c is a pointer to a function that takes two parameters a char and a pointer to a
  function that takes no parameters and returns an int
  and returns a pointer to void.

- **void* (*a[5])(char * const, char * const);**
  a ia a array of 5 pointers to functions that receive two const pointers
  to chars and return a void pointer

- **float (*(*e[10])(int &))[5];**
  e is an array of 10 pointers to functions that take a single
  reference to an int as an argument and return
  pointers to an array of 5 floats.

- **void ** (*d) (int &,char **(*)(char *, char **));**
  d is a pointer to a function that takes two parameters: a reference to an int and a
  pointer to a function that takes two parameters: a pointer to a char and a pointer
  to a pointer to a char and returns a pointer to a pointer to a char
  and returns a pointer to a pointer to void