Complicated declarations in C

December 9, 2019
Complicated declarations in C

- https://www.geeksforgeeks.org/complicated-declarations-in-c/
- Convert C declaration to postfix format and read from right to left.
  1. Start from declaration’s name
  2. Parse operations closed to which have been parsed based on precedence table
  3. Read postfix format from left to right.
Complicated declarations in C

- int (*fp) ();
- int (*daytab)[13];
- void (*f[10])(int, int);
- char (*(*x()))[]();
- char (*(*x[3]()))[5];
- int *(arr[5]());
- void (*bsd_signal(intsig, void(*func)(int)))(int);
Complicated declarations in C

- **int (*fp) ();**  
  fp * () int  
  Meaning: fp is pointer to function returning int

- **int (*daytab)[13];**  
  daytab * [13] int  
  Meaning: daytab is pointer to array of 13 integers.

- **void (*f[10]) (int, int);**  
  f[10] * (int, int) void  
  Meaning: f is an array of 10 pointer to function (which takes 2 arguments of type int) returning void

- **char (*(*x())[])) ();**  
  x () * [] * () char  
  Meaning: x is a function returning pointer to array of pointers to function returning char

- **char (*(*x[3])(()))[5];**  
  Meaning: x is an array of 3 pointers to function returning pointer to array of 5 char’s
Complicated declarations in C

- `char (**x[3])(())[5];`  
  Meaning: `x` is an array of 3 pointers to function returning pointer to array of 5 char’s

- `int (**arr[5])(());`  
  Meaning: `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 (int, func * (int) void ) * (int) void  
  Meaning: `bsd_signal` is a function that takes integer and 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)
Functions can not be put into array, though pointers to functions can. Example:

- `float *ptr[5](int);`  
  Postfix: `ptr [5] (int) * float`
- `float *(ptr[5])(int);`  
  Postfix: `ptr [5] (int) * float`
- `float (*ptr)[5](int);`  
  Postfix: `ptr * [5] (int) float`
- `float (((*ptr)[5])(int));`  
  Postfix: `ptr * [5] (int) float`

These are all illegal declarations such that they put functions to array