



- **Exercice 2.4, point 3)**, interpréter : **double (\*c) (double\*);**
  1. (\*c) (double\*) est un double
  2. (Règle fonction)  
(\*c) est une fonction prenant en paramètre un pointeur sur double et livrant un double
  3. (Règle parenthèses) :  
\*c est une fonction prenant en paramètre un pointeur sur double et livrant un double
  4. (Règle pointeur) :  
**c est un pointeur sur une fonction prenant en paramètre un pointeur sur double et livrant un double**



- **Exercice 2.4, point 6)**, interpréter : **int (\*(f[5])(void))[10];**
  1.  $(*(f[5])(void))[10]$  est un int
  2. (Règle tableau)  
 $(*(f[5])(void))$  est un tableau de 10 int
  3. (Règle parenthèses) :  
 $*(f[5])(void)$  est un tableau de 10 int
  4. (Règle pointeur) :  
 $(f[5])(void)$  est un pointeur sur un tableau de 10 int
  5. (Règle fonction) :  
 $(f[5])$  est une fonction sans paramètre livrant un pointeur sur un tableau de 10 int



6. (Règle parenthèses) :  
\*f[5] est une fonction sans paramètre livrant un pointeur sur un tableau de 10 int
7. (Règle pointeur)  
f[5] est un pointeur sur une fonction sans paramètre livrant un pointeur sur un tableau de 10 int
8. (Règle tableau)  
**f est un tableau de 5 pointeurs pointant chacun sur une fonction sans paramètre livrant un pointeur sur un tableau de 10 int**



## ***Solution exercice 2.4***

- 1) a est un tableau de 10 pointeurs pointant chacun sur un pointeur sur `char`
- 2) b est un pointeur sur une fonction sans paramètre renvoyant un pointeur sur un `double`
- 3) c est un pointeur sur une fonction prenant en paramètre un pointeur sur `double` et livrant un `double`
- 4) d est un tableau de 10 pointeurs sur `int`
- 5) e est un pointeur sur un tableau de 10 `int`
- 6) f est un tableau de 5 pointeurs pointant chacun sur une fonction sans paramètre et livrant un pointeur sur un tableau de 10 `int`
- 7) g est un pointeur contenant l'adresse d'un pointeur sur un tableau de 5 `double`



- **Exercice 2.5, point 4)**, déclarer en C :  
**p est un pointeur constant sur un tableau de 10 pointeurs sur double**
- 1. (Règle pointeur)  
\* const p est un tableau de 10 pointeurs sur double
- 2. (Règle tableau)  
(\* const p)[10] est un pointeur sur double
- 3. (Règle pointeur) :  
\* (\* const p)[10] est un double
- 4. (Finalisation) :  
double\* (\* const p)[10] = <initialisation>;      (car p = pointeur constant)



- **Exercice 2.5, point 6)**, déclarer en C :  
**f est une fonction prenant en paramètre un pointeur constant sur char et renvoyant un pointeur constant contenant l'adresse d'un pointeur sur char**
- 1. (Règle fonction)  
f(char\* const) est un pointeur constant contenant l'adresse d'un pointeur sur char
- 2. (Règle pointeur) :  
\* const f(char\* const) est un pointeur sur char
- 3. (Finalisation) :  
**char \*\* const f(char\* const);**



## ***Solution exercice 2.5***

- 1) `const int* t[10];`
- 2) `int* const t[10] = <initialisation>;`  
N.B. équivalent à dire que t est tableau constant de pointeurs
- 3) `int* (*p) (double (*) (double));`
- 4) `double* (* const p)[10] = <initialisation>;`
- 5) `char* (*t[10]) (double);`
- 6) `char** const f(char* const);`
- 7) `char* const (*f(void))[10];`