

EXEMPLE 1

```
length)
{
{
{
```

croissant ?

PROBLÈMES

```
void tri_decroissant(int *tab, int l)
{
    int i, j;
    for (i = length - 1; i > 0; i--)
        for (j = 0; j < i - 1; j++)
            if (tab[j] < tab[j + 1])
                int temp = tab[j];
                tab[j] = tab[j + 1];
                tab[j + 1] = temp;
        }
    }
}
```

EXEMPLE 2

```
}
```

POINTEUR D

- Permet de référencer direc

- float * p_float
- char * p_char

POINTELS

stocké dans une zone mémoire
(ext)

autre objet du programme, a
mémoire

naître son "type", comme pour

RÉCUPÉRATION DE L'AD

- utilisation temporaire du

```
void foo(void) {
    print("foo\n");
}

int main(void) {
    void * ptr_foo = &(foo);
    void * ptr_foo2 = foo;
    printf("@foo:%p - %p\n", ptr_foo, ptr_foo2);
    return 0;
}
```

ond à un pointeur statique

e la fonction conduit à son

RÉCUPÉRATION DE L'AD

- Soucis pointeur générique

PLES

fonctions ne retournant un

```
1){...}
```

```
_t);
```

```
r_2,ptr_3);
```

```
}
```

RAPPEL: APPEL

- L'appel d'une fonction s'effectue par son nom suivi entre parenthèses
- Le compilateur va remplacer l'exécution du code correspondant (à son adresse) et en positionner les arguments et variables locales

L DE FONCTION

APPEL DE

- Il est possible de procéder de

la déclaration d'un pointeur de
de la taille qui fait toute la

type de retour et mêmes t

POINTEUR DE FONCTION

retourner un pointeur de

en renvoyant le pointeur (disons
pointées s'entremêlent ainsi :

```
foo(l_args_foo) )  
s_fct);
```

paramètre une liste

foo) et retourne un pointeur
prototype est type_ret_fct

EXEMPLE

```
void foo_1(void){ printf(", World!\n");  
int foo_2(char * s){ return printf("%s", s);  
void (* foo_3(int i))(void){  
    printf("%d\n", i);  
    return &(foo_1);  
}  
int (* foo_4(int i))(char *){  
    printf("%d\n", i);  
    return &(foo_2);  
}  
int main(void){  
    void (*ptr_1)(void)=foo_3(1);  
    void (*ptr_2)(char *)=foo_4(2);  
    (*ptr_2)("Hello");  
    (*ptr_1)();  
    return 0;  
}
```

Que ce code

POINTEUR D'UN TYPEDEF

déclarant un nouveau type

POINTEUR DE FONCTION

- Pour passer un pointeur de f

UTILITÉ: EXEMPLE

implémentées comme suit et
 e pointeurs de fonctions

```
{
...
}

e) = {&(add), &(subtract),
```

RETOUR SUR L'U

- Le choix de l'opéra

```
double (* selectOperation(char choice)
int i=-1;
switch(choice){
    case '+': i=0; break;
    case '-': i=1; break;
    case 'x': i=2; break;
    case '/': i=3; break;
    default: return NULL;
}
return operations[i];
}
```

UTILITÉ: EXEMPLE

ations manquent pour que le

CONCL

- Les pointeurs de fonctions p