



## Systèmes Embarqués 1 & 2

Classes T-2/I-2 // 2018-2019

### a.08 – C - Pointeurs de fonctions *Exercices*

#### Exercice 1

Développer un petit programme permettant d'appeler un jeu de commandes quelconque avec leurs arguments à partir d'une chaîne de caractères. Le programme doit être généralisé et utiliser les pointeurs de fonctions.

Par exemple, le programme doit fonctionner correctement pour les lignes de commande ci-dessous :

```
1) "command1 10 20 30"  
2) "command2 10 un_string"  
3) "command3 30 40.20 50e-2"  
4) "command4"
```

#### Exercice 2

Soit les 4 fonctions suivantes

```
/* 1 */ int fnct_a (int *ptr, int max);  
/* 2 */ int fnct_b (int *ptk, int min);  
/* 3 */ int fnct_c (int *trd, int moy);  
/* 4 */ int fnct_d (int *utr, int val);
```

- Déclarer un tableau de pointeurs de fonctions capable de stocker ces fonctions, puis initialiser le tableau avec ces valeurs.
- Appeler chaque fonction à l'aide du tableau préalablement initialisé.

**Exercice 3**

Quelles seront les valeurs contenues dans le tableau « out » après l'exécution du code ci-dessous?

```
#define ARRAY_SIZE(x) (sizeof(x)/sizeof(x[0]))

typedef int (*fnct_t) (int);

static int f1 (int a) { return (25 + (3 * a)); }
static int f2 (int a) { return (192 - a); }
static int f3 (int a) { return (a - 100); }
static int f4 (int a) { return (a % 8); }

static fnct_t tabf[] = {f1, f2, f3, f4};

static int fnct(int x)
{
    unsigned k = (x >> 6) % ARRAY_SIZE(tabf);
    return tabf[k](x);
}

int main()
{
    int in[] = {10, 90, 120, 160, 180, 250};
    int out[ARRAY_SIZE(in)];

    for(unsigned i=0; i<ARRAY_SIZE(in); i++) {
        out[i]=fnct(in[i]);
    }

    return 0;
}
```

i	fnct	a	out[i]
0			
1			
2			
3			
4			
5			