TD n°1 de langage C : scalaires, fonctions, pointeurs

Vous devez faire tous les exercices d'acquisition. Lorsque ces exercices sont maîtrisés (si applicable), vous pouvez tenter de faire un ou plusieurs exercices de consolidation. Choisissez les exercices qui présentent pour vous une difficulté raisonnable et qui sont donc susceptibles de vous faire progresser.

En cas de difficulté, demandez un indice ou une explication à l'enseignant.

Exercice 1 (acquisition)

Réalisez une fonction `void abs(int* p)` qui remplace la valeur de la variable pointée par p par sa valeur absolue.

Exercice 2 (acquisition)

Réalisez une fonction `void hms(int t)` qui affiche à l'écran la durée t, exprimée en secondes, sous la forme heure, minutes, secondes.

Par exemple hms(3662) doit afficher 1h 1m 3s.

Exercice 3 (acquisition)

Soient les 2 fonctions suivantes :

```
int f(int x)
{
    return x*x;
}

int fpn(int n, int x)
{
    if(n==0) return x;
    else return f(fpn(n-1,x));
}
```

Quelle est la valeur retournée par **fpn(3,2)** ?

Exercice 4 (acquisition)

Soient les initialisations suivantes :

```
int x=1, y=1;
int* p1 = &x ;
int* p2 = &y ;
```

Dessinez la représentation en mémoire des variables x, y, p1 et p2, avec les lien entre variables pointeurs et valeurs pointées.

Donnez les valeurs des variables x et y après l'exécution du code suivant :

```
(*p1)++; *p2 = *p1; (*p2)++;
```

Exercice 5 (consolidation)

Modifiez la fonction `f` de l'exercice 3 de manière à ce que la fonction **fpn** du même exercice permette de calculer 2 à la puissance **x**, pour toute valeur positive ou nulle de **x**.

Indiquer les paramètres à transmettre à **fpn** pour permettre ce calcul et donner en exemple un programme qui calcul et affiche 2 à la puissance 5.

Exercice 6 (consolidation)

On suppose qu'on dispose d'une fonction `int isPrime(int x)` qui teste si x est un nombre premier. Cette fonction retourne une valeur Booléenne.

Réalisez une fonction `void fact(int n)` qui affiche la décomposition de l'entier n en facteurs premiers, c'est à dire sous la forme d'un produit de nombres premiers.

Par exemple, **fact(44)** doit afficher 2 * 2 * 11.

Exercice 7 (consolidation)

On souhaite utiliser une variable de type int comme une pile de bits.

Réalisez une fonction `void push(int* p, int b)` qui empile dans la variable pointée par p le bit 0 si b vaut 0 et le bit 1 sinon.

Réalisez une fonction `int pop(int* p)` qui retourne la valeur du dernier bit empilé.

Indice : Multiplier un entier par deux revient à décaler vers la gauche tous les bits de sa représentation en base 2. La division par deux réalise un décalage vers la droite.

Le bit le plus à droite représente le dernier bit empilé.