A2 Prog – IIM

TD - Pile

Exercice 1 :

```
void A(int remainingTurn, int playerId);
void B(int remainingTurn, int playerId);
void C(int remainingTurn, int playerId);

int main(){
    int remainingTurn = 100;
    int playerId = 0;
    A(remainingTurn , playerId);
    return 0;
}


void A(int remainingTurn, int playerId) {
    std::cout << "Starting turn for player : " << playerId << "." <<
std::endl;
    B(remainingTurn, playerId);
}

void B(int remainingTurn, int playerId) {
    std::cout << "Player " << playerId << " has played. Switching player."
<< std::endl;
    playerId = (playerId+1)%2;
    C(remainingTurn, playerId);
}

void C(int remainingTurn, int playerId) {
    --remainingTurn;
    std::cout << "End of turn. " << remainingTurn << " turns left." <<
std::endl;
    if(remainingTurn > 0){
        A(remainingTurn, playerId);
     }
}
```

Écrire un code équivalent appelant bien trois fonctions A, B et C mais n'ayant pas recours à la récursion.

Exercice 2 :

Résultat final :

0	8	7	1	5
3	5	2	8	9
0	4	0	1	0
2	4	3	6	1
1	4	3	1	0

```
struct Coordinate {
    int i;
    int j;
};

int main() {
    Coordinate start = {2, 2};
    int tab[5][5] = {{0,8,7,1,5}, {3,5,2,8,9}, {0,4,0,1,0}, {2,4,3,6,1},
{1,4,3,1,0}};

    /*
    *
    */
}
```

Écrire une fonction “FindHighestAdjacent” donnant la direction (di, dj) de la case adjacente (hors diagonales) avec la plus grande valeur à une case d’index (i, j) sur un tableau donné. Par exemple, la case adjacente à la case d’index (2, 2) avec la plus grande valeur est la case d’index (1,2). Elle a pour valeur 4 et est en direction (-1,0).

- Paramètres : tableau, taille du tableau, coordonnées (i, j).
- Valeur de retour : Direction vers la case de plus grande valeur (type Coordinate).

Attention aux index en limites de tableau.

Écrire une fonction récursive recherchant un chemin sur le tableau en suivant les cases adjacentes de plus grandes valeurs. Cette fonction doit :

- Utiliser la fonction programmée précédemment pour se diriger.
- Écrire 0 dans la case du tableau visitée avant de continuer sa progression.
- Se limiter à un nombre de déplacement donné.
- S’arrêter si toutes les cases adjacentes ont pour valeur 0.

Elle répondra à la signature suivante :

- Paramètres : tableau, taille du tableau, coordonnées (i, j) de la case à visiter, nombre de déplacements restants.
- Valeur de retour : Coordonnées de fin

Réécrire cet algorithme avec la fonction sans récursion.