Corrigé TD1 ASD UPEC

```
bool Permuter(int tableau[ ], int taille, int index1, int index2)
{
	bool permutationReussie = false;
	if(tableau != null)
	{
	if( (index1 >= 0) &&  (index1 < taille) && (index2 >= 0) &&  (index2 < taille))
	{
		int tmp = tableau[index1];
		tableau[index1] = tableau[index2];
		tableau[index2] = tmp;
		permutationReussie = true;
	}
	}
	return permutationReussie;
}

// avec optimisation  :
// - si aucune permutation lors d'un parcours
// - alors tableau trié, donc arrêt des parcours (cf. variable booléenne 'fin').
void TriBulles(int tableau[ ], int taille)
{
	bool fin = false;
	// on peut le faire aussi avec un do-while
	for(int i = taille-1 ; (!fin) && (i >= 1) ; i--)
	{
		fin = true;  // supposons qu'il n'y aura pas de permutation

		// échanger 2 éléments consécutifs si dans mauvais ordre
		for( int j = 0 ; j < i ; j++)
		{
			if( tableau[j] > tableau[j+1] )
			{
				Permuter(tableau, j, j+1);
				fin = false;
			}
		}
	}
}
```

```
// - on suppose le tableau trié jusqu'à la position i-1
// - on positionne l'élément tableau[i]à sa position en décalant tous les éléments plus grands vers la droite.
void TriInsertion(int tableau[ ], int taille)
{
for( int i = 1 ; i < taille ; i++ )
  {
    int tmp = tableau[i];
    int j = i-1;

    while( (j >= 0) && (tmp < tableau[j] ) )
    {
      // décalage des valeurs plus grandes que tableau[j]
      tableau[j+1] = tableau[j];
      j--;
    }

    // puis positionnement de l'élément tableau[j] à sa place
    tableau[j+1] = tmp;
  }
}

// Principe
// - on cherche l'élément min dans tout le tableau et on le permute avec le 1er élément.
// - on recommence la recherche du min mais dans la portion du tableau entre la 2nde position
// et la dernière,  puis on permute ce min avec le 2nd élément. - etc.
void TriSelection(int  tableau [ ])
{
        for( int i = 0 ; i < taille -1 ; i++ )
        {int posMin = i;

        // recherche position de l'élément min
        // on peut faire une fonction séparée  indiceMin(int tab[], int taille, int i)
        // qui recherche l'indice min à partir de l'indice i et l'appeler directement ici
        for( int j = i+1 ; j < taille ; j++ )
        {
                if( tableau[j] < tableau[posMin] )
                {
                        posMin = j; // mise à jour (de la position) du min
                }
        }
        Permuter(tableau, i, posMin);
        }
}
```