TD5 — Gestion d'erreurs, robustesse

On considère le paquetage `type_pile` du TD/TP n° 4, dont voici ci-dessous une implémentation.

```c
#include "type_pile.h"
#include <stdio.h>

/* Créer une pile vide */
void creer_pile(PileEntiers *p) { p->n = 0; }

/* Retourne vrai ssi p est vide */
int est_vide(PileEntiers *p) { return (p->n == 0); }

/* Renvoie l'entier en haut de la pile */
int sommet(PileEntiers *p) { return p->tab[p->n - 1]; }

/* Renvoie le nombre d'éléments dans la pile */
int taille(PileEntiers *p) { return p->n; }

/* Vider la pile p */
void vider(PileEntiers *p) { p->n = 0; }

/* Empiler un entier x */
void empiler(PileEntiers *p, int x) {
  p->tab[p->n] = x;
  p->n = p->n + 1;
}

/* Supprimer et renvoyer l'entier en haut de la pile */
int depiler(PileEntiers *p) {
  p->n = p->n - 1;
  return p->tab[p->n];
}
```

Exercice 1. Donner les préconditions requises pour l'utilisation de chaque fonction de ce paquetage.

Exercice 2. Ces préconditions peuvent ne pas être satisfaites, dans le cas d'une mauvaise utilisation de ce paquetage. Quel sera alors le comportement des fonctions ?

Exercice 3. Modifiez le paquetage pour permettre la gestion des erreurs.