Atelier 4 : Les piles en C

On voudrait gérer une pile d'entiers représentée par une liste simplement chaînée.

Une pile est représentée par les structures de données suivantes :

```
typedef struct cellule{
int val;
struct cellule * suivant ;
}cellule ;

typedef struct cellule *pile ;
```

1. Ecrire la fonction **void creer_pile(pile *p)** permettant de créer une pile.
2. Ecrire la fonction **int vide_pile(pile p)** permettant de tester si une pile est vide ou non.
3. Ecrire la fonction **int sommet_pile(pile p)** permettant de retourner le sommet de la pile.
4. Ecrire la fonction **void empiler_pile(pile *p, int x)** permettant d'empiler l'entier x au sommet de la pile (ajout en tête d'une liste chaînée).
5. Ecrire la fonction **void depiler_pile(pile *p)** permettant de supprimer le sommet de la pile (suppression en tête d'une liste chaînée).
6. Ecrire la fonction **void afficher_pile(pile p)** permettant d'afficher les éléments de la pile dans l'ordre inverse de leur empilement. (utiliser seulement les fonctions de manipulation des piles **creer_pile, vide_pile, sommet_pile, emplier_pile et depiler_pile) .**
7. Ecrire la fonction **void afficher_pile_Rev(pile p)** permettant d'afficher les éléments de la pile dans leur ordre d'empilement.
8. Ecrire la fonction **void ranger_pair_impair(pile *p)** permettant de ranger les entiers pairs au- dessous des entiers impairs. Pour ce faire, on peut se servir de deux piles intermédiaires pPair et pImpair contenant respectivement les entiers pairs et les entiers impairs.
9. Ecrire la fonction principale **main** où vous :
    - créez une pile d'entiers et de la charger par des entiers positifs autant de fois que l'utilisateur le désire.
    - Afficher le contenu de la pile
    - Afficher le contenu de la pile en sens inverse
    - Ranger les entiers pais au-dessous des entiers impairs et vérifier le rangement