```c
void ABVersTabRec(PArbre a, int pos, TArbBin *T) {
	if (! estVide(a)) {
		T[pos].elmt = a->elmt;
		T[pos].equilibre = a->equilibre;
		ABVersTabRec(filsGauche(a), 2 * pos + 1, T);
		ABVersTabRec(filsDroit(a), 2 * pos + 2, T);
	}
}

TArbBin* ABVersTab(PArbre a) {
	TArbBin *T;
	if (estVide(a)) return NULL;
	if ((T = (TArbBin *) calloc((int) pow(2, hauteur(a) + 1) - 1, sizeof(TArbBin))) ==
NULL) {
		puts("Erreur allocation Arbre vers Tableau");
		exit(1);
	}
	ABVersTabRec(a, 0, T);
	return T;
}



void affArbreGraphique(PArbre a) {
	TArbBin *Tarb;
	int iTarb;
	int tailleAff = 7; // noeud=" xx,xx "
	int largeur; // largeur d'affichage maximum
	int h;
	int nbNoeuds;
	int p, i, j;
	int ecart;

	puts("");
	if (estVide(a)) {
		puts("Arbre vide");
		return;
	}
	h = hauteur(a);
	largeur = tailleAff * ((int) pow(2, h)); // taille element * nbFeuillesMax;
	Tarb = ABVersTab(a);

	iTarb = 0;
	for (p = 0, nbNoeuds = 1; p <= h; p++, nbNoeuds *= 2) {	// p = profondeur,
nbNoeuds par niveau
		ecart = (largeur / (nbNoeuds * 2));
		for (j = 1; j <= nbNoeuds; j++) {
			for (i = 0; i < ecart - 3; i++) putchar(' '); // -3 car 3 caracteres
avant milieu
			if (Tarb[iTarb].elmt == 0)
				printf(" ..... ");
			else
				printf(" %2d,%+1d ", Tarb[iTarb].elmt, Tarb[iTarb].equilibre);
			iTarb++;
			if (j < nbNoeuds && p < h) // entre 2 noeuds sauf au dernier niveau
				for (i = 0; i < ecart - 4; i++) putchar(' '); // -4 car 4
caracteres apres milieu
		}
```

```
                puts("");
        }
        free(Tarb);
        puts("");
}
```