

*Durée : 55'**Calculatrice autorisée*

Question 1

32 (=4+6+12+10) points

Pour implémenter les polynômes à coefficients entiers, on définit le type `poly` par :

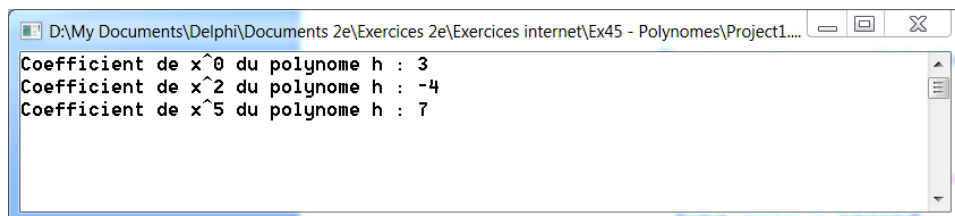
```

type poly = record
    nom:string;
    c:array[0..100] of integer;
    d:integer
end;

```

où le champ `nom` contient le nom, le champ `d` le degré et le champ `c` la liste des coefficients du polynôme suivant les puissances croissantes de la variable : `p.c[i]` est le coefficient de x^i du polynôme stocké dans la variable `p`.

- (1) Ecrire les affectations du programme principal qui permettent de définir complètement la variable `h` de type `poly` contenant le polynôme $h(x) = 7x^5 - 4x^2 + 3$. (On ne demande pas d'écrire une procédure saisie !)
- (2) Ecrire la procédure `afficheCoeff`, qui prend en entrée un polynôme `p` de type `poly` et qui écrit tous les coefficients non nuls de ce polynôme à l'écran. La fenêtre d'exécution suivante doit être produite à l'appel `afficheCoeff(h)`, où `h` est le polynôme de la question (1).



- (3) a) Ecrire la fonction `produit` qui prend en entrée deux polynômes `a` et `b` de type `poly` et qui retourne le polynôme produit `a.b`.
b) On veut stocker le polynôme h^4 dans une variable `q` de type `poly`, `h` étant le polynôme de la question (1). Quelle(s) ligne(s) doit-on ajouter au programme principal pour y arriver ? Que contiennent alors `q.nom` et `q.d` ?
- (4) a) Ecrire la fonction `horner` qui prend en entrée un polynôme `p` de type `poly` et un réel `x` de type `extended` et qui retourne la valeur numérique du polynôme `p` en `x` à l'aide de l'algorithme de Horner.
b) Faire un tableau d'exécution de la fonction `horner` montrant comment elle calcule $h(-2)$, `h` étant le polynôme de la question (1).

Tournez s.v.p.

Question 2

18 (=5+6+7) points

Pour implémenter des listes **d'entiers**, on définit le type `list` par :

```
type list = record
    items:array[1..100] of integer;
    length:integer
end;
```

où le champ `length` contient le nombre d'éléments (≤ 100) de la liste et le champ `items` le tableau avec les éléments de la liste.

- (1) Ecrire une procédure `randomFill`, qui prend en entrée une liste de type `list` et qui remplit cette liste aléatoirement : le nombre d'éléments de la liste est choisi aléatoirement entre 10 et 100 (bornes comprises) et les éléments de la liste sont des entiers aléatoires compris dans $[-20,20]$.
- (2) Ecrire une fonction `countIn` qui prend en entrée deux entiers `a` et `b` (avec $a < b$) et une liste de type `list` et qui retourne le nombre d'éléments de la liste compris dans $[a,b]$.
- (3) Ecrire une fonction `averageP` qui prend en entrée une liste de type `list` et qui retourne la moyenne arithmétique **des éléments positifs** (≥ 0) de cette liste.

Question 3

10 points

Ecrire la **fenêtre d'exécution** du programme suivant :

```
program devoirIII;
{$APPTYPE CONSOLE}

uses
    SysUtils;

var x,y:integer;

procedure titi(var a:integer;b:integer);
begin
    while a mod b<3 do
        begin
            writeln('a=',a,' et b=',b);
            a:=3*a;
            b:=b+1;
        end
    end;

begin
    writeln('Exemple 1');
    x:=6; y:=1;
    titi(x,y);
    writeln('x=',x,' et y=',y);
    writeln;
    writeln('Exemple 2');
    x:=2; y:=15;
    titi(y,x);
    writeln('x=',x,' et y=',y);
    readln
end.
```