

TD 2 (Algorithmique et programmation)

Exercice 1 : soit le développement limité de e^x :

$$e^x = 1 + (x/1!) + (x^2/2!) + (x^3/3!) \dots + (x^i/i!)$$

Il s'agit d'écrire une action permettant de calculer la valeur de e^x en sommant les $n+1$ premiers termes du développement précédent. Il faudra au préalable trouver les différentes équations de récurrences, et donner ensuite les deux versions récursive et itérative d'une telle fonction.

Exercice 2 : Donnez un algorithme récursif qui prend un entier a produit la représentation binaire de a .

Exercice 3 : Ecrire deux fonctions récursives calculant $P(x, n) = x^n$, n entier positif, en utilisant les définitions récurrentes suivantes :

1. *Base* : $P(x, 0) = 1$; *Récurrance* : $P(x, n) = x * P(x, n-1)$.

2. *Base* : $P(x, 0) = 1$; *Récurrance* : $P(x, 2*n) = P(x, n) * P(x, n)$; et $P(x, 2*n+1) = P(x, n) * P(x, n) * x$

Laquelle préférez vous ? pourquoi?

Exercice 4 :

L'objectif ici est la réalisation d'un programme complet en Pascal permettant d'effectuer différentes opérations sur des ensembles d'entiers (un ensemble ne doit pas contenir d'éléments redondants).

Le type ensemble d'entiers est défini comme suit :

```
Const Taille_Max      =      ... ;

Type ensembleEntiers  =      record
                                cardinal : integer;
                                elements array[1.. Taille_Max] of integer ;
                                end ;
```

Ecrire les actions suivantes, nécessaires à la gestion d'un ensemble.

- affectation d'un ensemble à vide ;
- test de vacuité d'un ensemble ;
- cardinal d'un ensemble ;
- test d'appartenance d'un élément à un ensemble ;
- suppression d'un élément ;
- ajout d'un élément ;
- saisie des éléments d'un ensemble ;
- affichage des éléments d'un ensemble ;
- copie d'un ensemble dans un autre. ;
- réunion de deux ensembles ;
- intersection de deux ensembles ;
- différence entre deux ensembles ;
- différence symétrique ;
- test d'inclusion d'un ensemble dans un autre ;
- tester l'égalité de deux ensembles ;

Exercice 5 : déroulez le programme suivant et indiquez les valeurs affichées à l'écran ainsi que les états de la pile d'exécution.

```
PROGRAM  PROG1 ;
  VAR I, J, K, L : INTEGER ;

  PROCEDURE  P (VAR X : INTEGER ; Y : INTEGER ) ;
```

```

VAR L : INTEGER ;

PROCEDURE EXC (VAR J, I : INTEGER) ;
    VAR X : INTEGER ;

    BEGIN (* EXC *)
        X := I ;
        I := J ;
        J := X ;
    END ; (* EXC *)

BEGIN (* P *)
    L := 0 ;
    EXC(X,Y) ;
    X :=10*X ;
    Y :=100*Y ;
    K :=1000*K+L
END ; (* P *)

BEGIN (* PROG1 *)
    I :=1 ; J :=2 ; K :=3 ; L :=4 ;
    P(I,J) ;
    WRITELN(I ,J,K,L) ;
    I :=1 ; J :=2 ; K :=3 ;L :=4 ;
    P(K,L) ;
    WRITELN(I,J,K,L) ;
END.

```

Exercice 6 : déroulez le programme suivant et indiquez les valeurs affichées à l'écran.

```

Program prog2 ;
const aff = 4 ;
var a,b,c,d :integer ;
    procedure P (a, b : integer ; var c, d : integer);
    begin
        a := b ;
        b := c + d ;
        d := b - d ;
        c := b - c
    end ;
begin
    a := 1 ; b := 2 ; c := 3 ; d := 4 ;
    P(b, c, d, a) ;
    writeln(a :aff, b :aff ; c : aff , d :aff) ;
end.

```