***Les outils de base de l’algorithmique***

***Exercice 1 :***

Ecrire un **algorithme** qui demande les coordonnées de deux points dans le plan, calcule et affiche à l’écran la distance entre ces deux points.

N.B.

la distance entre deux points A(x1,y1) et  B(x2,y2)    est :    AB= sqrt((x2-x1)^2  + (y2-y1)^2)

on donne la fonction sqrt(x) qui renvoie la racine carrée d’un nombre réel  x .

**Solution :**

***Algorithme  calcul\_distance;***

Var

      X1,x2,y1,y2,s : réels ;

Debut

            Ecrire(‘entrer la valeur de x1 : ‘) ;

            Lire(x1) ;

            Ecrire(‘’ entrer la valeur de y1 : ‘’) ;

            Lire(y1) ;

Ecrire(‘’ entrer la valeur de x2 : ‘’) ;

            Lire(x2) ;

Ecrire(‘’ entrer la valeur de y2 : ‘’) ;

            Lire(y2) ;

            S <-- sqrt((x2-x1)^2  + (y2-y1)^2) ;

            Ecrire(‘la distance entre A(‘ ,x1,’,’,y1,’) et  B(‘,x2,’,’,y2,’) est :  ‘,s) ;

fin

***Exercice 2 :***

Elaborer un **algorithme** permettant de demander les valeurs de trois résistances r1,r2 et r3 et de calculer et afficher leurs résistances équivalente dans les deux cas suivants :

N.B.

Lorsque ces résistances sont branchées en série :

                                     Rser = r1+r2+r3

Lorsque ces résistances sont branchées en parallèle :

                           Rpar=(r1\*r2\*r3)/(r1\*r2+r1\*r3+r2\*r3)

***Solution :***

***Algorithme  calcul\_resistance;***

Var

      r1,r2,r3,Rpar,Rser: réels ;

Debut

                Ecrire(‘entrer la valeur de r1 : ‘) ;

                Lire(r1) ;

                Ecrire(‘ entrer la valeur de r2 : ’) ;

                Lire(r2) ;

Ecrire(‘ entrer la valeur de r3 :  ’) ;

                Lire(r3) ;

                Rser <--  r1+r2+r3;

                Rpar <--  (r1\*r2\*r3)/(r1\*r2+r1\*r3+r2\*r3) ;

                Ecrire(‘ la résistance équivalente a r1 ,r2 et r3 en série est : ‘,Rser) ;

Ecrire(‘ la résistance équivalente a r1 ,r2 et r3 en parallèle  est : ‘,Rpar) ;

fin

***Exercice 3 :***

Ecrire un **algorithme** qui permet de calculer la surface d’un triangle quelconque  dont les cotés  ont une longueur donnée : a,b et c .

N.B.

           S= sqrt(r\*(r-a)\*(r-b)\*(r-c))      avec :     r =(a+b+c)/2

***Solution :***

***Algorithme  calcul\_surface;***

Var

      a,b,c,r,s : réels ;

Debut

                Ecrire(‘entrer la longueur du côté a  : ‘) ;

                Lire(a) ;

                Ecrire(‘entrer la longueur du côté b  : ‘) ;

                Lire(b) ;

    Ecrire(‘entrer la longueur du côté c  : ‘) ;

                Lire(c) ;

                r  <--     (a+b+c)/2 ;

                S <--  sqrt(r(r-a)(r-b)(r-c)) ;

                Ecrire(‘la surface du triangle abc est :  ‘,s);

Fin

***Exercice 4 :***

Dans une école un étudiant passe quatre matières à l’examen :

1ère  matiere écrite   :   coefficient =3

2ème matiere ecrite :     coefficient =2

1ère matiere orale :       coefficient =4

2ème matière orale :     coefficient =5

Le coefficient de l’ecrit est 6, celui de l’orale est 1 dans le calcul de la moyenne generale.

Ecrire un **algorithme** permettant d’entrer toutes les notes de calculer et d’affficher la moyenne de l’ecrit ,celle de l’orale et la moyenne génerale.

***Solution :***

***Algorithme  calcul\_note;***

Var

             me1,me2,mo1,mo2,moy : réels ;

const

             cme1=3 ;

             cme2=2 ;

              cmo1=4 ;

             cmo2=5 ;

Debut

                Ecrire(‘entrer la  note du 1ère matiere écrite : ‘) ;

                Lire(me1) ;

                Ecrire(‘entrer la  note du 2ème matiere écrite : ‘) ;

                Lire(me2) ;

    Ecrire(‘entrer la  note du 1ère matiere orale : ‘) ;

                Lire(mo1) ;

    Ecrire(‘entrer la  note du 2ème matiere orale: ‘) ;

                Lire(mo2) ;

                moy<--  (me1\*cme1+me2\*cme2+mo1\*cmo1+mo2\*cmo2)/(cme1+cme2+cmo1+cmo2) ;

   Ecrire(‘ la moyenne generale est   :  ‘,moy) ;

fin

***Exercice 5 :***

Ecrire un **algorithme** qui lit trois nombres dans trois variables A ,B et C , puis fait la permutation circulaire de ces trois nombres (sens trigonométrique) et affiche les nouveaux contenus des variables A,B et C .

***Solution :***

***Algorithme  calcul\_permutation;***

Var

            A,b,c,aux : réels ;

Debut

                Ecrire(‘entrer la  de a : ‘) ;

                Lire(a) ;

                Ecrire(‘entrer la  de b : ‘) ;

                Lire(b) ;

    Ecrire(‘entrer la  de c : ‘) ;

                Lire(c) ;

    Ecrire(‘a = ‘,a,’     b= ‘,b,’         c=’,c) ;

                Auxß c ;

                a<--  b ;

                b<--  a ;

                a<--  aux ;

Ecrire(‘a = ‘,a,’     b= ‘,b,’         c=’,c) ;

fin

***Exercice 6 :***

Soit N un nombre entier. Proposer une opération avec laquelle nous pourrons conclure si le nombre N est pair ou impair.

***Solution :***

***Algorithme  parite;***

Var

      N :entier;

Debut

                Ecrire(‘entrer  un entier : ‘) ;

                Lire(N) ;

                Si  N mod 2 =0 alors

        Ecrire(‘le nombre est pair);

    Else

         Ecrire(‘le nombre est impair .’);

   Finsi

fin

***Exercice 7 :***

Ecrire un **algorithme** qui calcule le périmetre d’un cercle : p=2\*π\*R

***Solution :***

***Algorithme  calcul\_perimetre;***

Const

           Pi=3.14 ;

Var

            R,p: réels ;

Debut

                Ecrire(‘entrer le rayon R : ‘) ;

                Lire(R) ;

                P<--  2\*pi\*R

Ecrire(‘ le perimetre du cercle R=’,R,’ est   :’,p) ;

fin

***Instructions conditionnelles et alternatives***

***Exercice 1 :***

Ecrire un **algorithme** qui calcule la valeur absolue d’un nombre réel .

|x|= x  si    x>0

|x|= -x  si   x<0

***Solution :***

***Algorithme  valeur\_absolue;***

Var

      X :réels;

Debut

                Ecrire(‘entrer  un nombre : ‘) ;

                Lire(X) ;

                Si  X>0 alors

        Ecrire(‘la valeur absolue de X=’,X);

    Sinon

         Ecrire(‘la valeur absolue de X=’,X);

    Finsi

fin

***Exercice 2 :***

Ecrire un **algorithme** qui permet d’afficher la valeur absolue de la différence entre deux nombres réels saisis au clavier.

|x-y|= x –y  si    x>y

|x-y|= -(x –y)  si   x<y

***Solution :***

***Algorithme  valeur\_absolue;***

Var

      x,y :réels;

Debut

                Ecrire(‘entrer  x : ‘) ;

                Lire(x) ;

                Ecrire(‘entrer  y : ‘) ;

                Lire(y) ;

    Si  x>y  alors

        Ecrire(‘la valeur absolue de x-y=’,x-y);

    Sinon

         Ecrire(‘la valeur absolue de x-y =’,y-x);

    Finsi

fin

***Exercice 3 :***

On désire écrire un **algorithme** qui permet d’afficher le jour correspondant à un chiffre allant de 1 à 7 , entré au clavier.

Résoudre ce problème avec deux méthodes :  (si imbriquée , primitive cas).

***Solution 1 :***

***Algorithme  affichage\_jour;***

Var

      jour :entier;

Debut

                Ecrire(‘entrer  un chiffre de 1 a 7 : ‘) ;

                Lire(jour) ;

Si  jour=1 alors

        Ecrire(‘ lundi ’);

Sinon Si  jour=2  alors

             Ecrire(‘ mardi ’);

           Sinon Si  jour=3  alors

                        Ecrire(‘ mercredi ’);

                 Sinon Si  jour=4  alors

                               Ecrire(‘ jeudi ’);

                         Sinon Si  jour=5  alors

                                        Ecrire(‘vendredi’);

                                   Sinon Si  jour=6  alors

                                                 Ecrire(‘samedi’);

                                               Sinon  Si  jour=2  alors

                                                                  Ecrire(‘dimanche’);

                                                           Sinon

                                                                 Ecrire(‘ce n’est pas un jour de semaine’);

                                                            Finsi

                                               Finsi

                                     Finsi

                          Finsi

                  Finsi

Finsi

      Finsi

fin

***Solution 2 :***

***Algorithme  affichage\_jour;***

Var

      jour :entier;

Debut

                Ecrire(‘entrer  un chiffre de 1 a 7 : ‘) ;

                Lire(jour) ;

                Cas  jour

                               1 : Ecrire(‘LUNDI ‘) ;

               2 : Ecrire(‘MARDI ‘) ;

               3 : Ecrire(‘MERCREDI ‘) ;

                4 : Ecrire(‘JEUDI ‘) ;

                5 : Ecrire(‘VENDREDI ‘) ;

                6 : Ecrire(‘SAMEDI ‘) ;

                7 : Ecrire(‘DIMANCHE ‘) ;

             Sinon

                         Ecrire(‘ il faut choisir un nombre entre 1 et 7 !!!! ‘) ;

FinCas

fin

***Exercice 4 :***

Ecrire un **algorithme** qui permet de saisir deux nombres entiers x ,y et les afficher à l’écran dans l’ordre croissant.

***Solution :***

***Algorithme  tri\_deux\_nombre;***

Var

      x,y :réels;

Debut

                Ecrire(‘entrer  x : ‘) ;

                Lire(x) ;

                Ecrire(‘entrer  y : ‘) ;

                Lire(y) ;

    Si  x>y  alors

        Ecrire(y,’          ‘,x);

   Sinon

         Ecrire(y,’         ‘,x);

   Finsi fin

***Exercice 5 :***

Ecrire un **algorithme** qui teste si une année est bissextile ou non.

N.B.

Une année est bissextile si elle est divisible par 4 et pas par 100 ou si elle est divisible par 400.

***Solution :***

***Algorithme  annne\_bissextile;***

Var

      annee :réels;

Debut

                Ecrire(‘entrer  l’année : ‘) ;

                Lire(annee) ;

    Si  ((annee mod 4 =0 et annee mod 100 <> 0) ou annee mod 400 =0 )  alors

        Ecrire(‘l’année que vous avez entrer est bissextile .’);

    Sinon

         Ecrire(‘l’année que vous avez entrer n’ est pas  bissextile .’);

    Finsi

fin

***Exercice 6 :***

Ecrire un **algorithme** permettant de résoudre une équation de deuxième degré :  ax²+bx+c=0 .

***Solution :***

***Algorithme  calcul\_permutation;***

Var

            A,b,c,d : réels ;

Debut

                Ecrire(‘entrer le coefficient a : ‘) ;

                Lire(a) ;

                Ecrire(‘entrer le coefficient  b : ‘) ;

                Lire(b) ;

Ecrire(‘entrer le coefficient  c : ‘) ;

                Lire(c) ;

                Si a=0 alors

                     Si b=0 alors

                            Si c=0 alors

                                  Ecrire(‘ la solution est   : S = R’) ;

                               sinon

                                    Ecrire(‘ l’equation n’a pas de solution ’) ;

                            Finsi

                     sinon

                           Ecrire(‘la solution est   : S = ’,-c/b) ;

                     Finsi

                sinon

                     D<-- b\*b-4+a\*c ;

                      Si d=0 alors

                           Ecrire(‘la solution est   : S = ’,-b/(2\*a)) ;

                      Sinon si d>0 alors

                                       Ecrire(‘l’equation a deux solution:  S1= ’,(-b- /(2\*a) ,’ et  S2 = ‘,(-b+ /(2\*a)) ;

                                   Sinon

                                         Ecrire(‘l’equation n’a pas de solution dans R ’) ;

                                    Finsi

                        Finsi

               Finsi

fin

***Exercice 7 :***

Une librairie décide de faire des remises sur les prix d’abonnement à une revue scientifique selon le menu suivant :

Ancien abonné : -15%

Etudiant              : -20%

Nouvel abonné : 00%

Etranger             : + 25%

Le calcul du prix d’**abonnement** de fait en fonction du tarif normal d’**abonnement** (TN) et de la qualité de l’abonné (Q). (une seule qualité est acceptée par abonné).

Ecrire un **algorithme** permettant de calculer le prix à payer.

***Solution :***

***Algorithme  librairie;***

Var

      Q : entier ;

      TN,TR,R,RAP :réels;

Debut

                Ecrire(‘Entrer le tarif normal : ‘) ;

                Lire(TN) ;

              Ecrire(‘--------------- MENU -----------------------‘) ;

              Ecrire(‘----------Ancien Abonné------------------ 1‘) ;

              Ecrire(‘-----------------Etudiant------------------- 2‘) ;

Ecrire(‘-----------------Nouvel abonné--------------- 3‘) ;

                Ecrire(‘-------------------Etranger---------------------- 4‘) ;

               Ecrire(’Entrer la qualité de l’abonné ? ‘’) ;

                Lire(Q) ;

                Cas Q

                               1 : TR <--  -0.15 ;

                               2 : TR <--  -0.20 ;

                               3 : TR <--  -0.00 ;

                               4 : TR <--  +0.25 ;

                Fincas

                Si (Q<1) ou (Q>4) alors

                      Ecrire(‘Erreur de choix’) ;

               Sinon

                           R <--  TN \*TR ;

                           PAR <--  TN+R ;

                             Ecrire(‘Le prix à payer est :’,PAP) ;

                Finsi

Fin

***Exercice 8 :***

Ecrire un **algorithme** de résolution.

On dispose de trois nombres réels saisis au clavier.

Selon un choix effectué a partir d’un menu affiché a l’ecran, on désire calculer la somme ou le produit ou la moyenne ou chercher le minimum de ces trois nombres.

Le menu doit se présenter a l’ecran de la manière suivante :

------------------ MENU ---------------------

1 --------------somme ----------------------

2---------------produit ----------------------

3---------------Moyenne -------------------

4---------------Minimum -------------------

5---------------Maximum ------------------

-------------------------------------------------

Entrez votre choix ?

***Solution :***

***Algorithme  calcul;***

Var

        X,Y,Z,R:réels;

        Choix : caractère ;

Debut

                Ecrire(‘Entrer le premier nombre : ‘) ;

                Lire(X) ;

    Ecrire(‘Entrer le deuxième nombre : ‘) ;

                Lire(Y) ;

    Ecrire(‘Entrer le troisième nombre : ‘) ;

                Lire(Z) ;

                Ecrire(‘--------------- MENU -----------------------‘) ;

                Ecrire(‘------------------Somme------------------- 1‘) ;

                Ecrire(‘----------------- Produit-------------------- 2‘) ;

    Ecrire(‘------------------Moyenne----------------- 3‘) ;

                Ecrire(‘-----------------Minimum------------------ 4‘) ;

                Ecrire(‘-----------------Maximum------------------ 5‘) ;

                Ecrire(’Entrer  votre choix ? ‘’) ;

                Lire(choix) ;

                Cas choix

                               1 : R <--  X+Y+Z;

                               2 : R <--  X\*Y\*Z;

                               3 : R <--  (X+Y+Z)/3;

                               4 :          Si  X<Y   Alors

                                        R ß X ;

                                Sinon

                                   R <--  Y ;

                                 Finsi

                                Si  Z<R  Alors

                                    R <--  Z ;

                                finsi

                 5 :         Si X>Y  alors

                                   R <--  X ;

                                Sinon

                                    R <--  Y ;

                              Finsi

                              Si  Z>R  Alors

                                  R <--  Z ;

                               finsi

                Fincas

                Si (choix>=1 ) et (choix<=5) alors

                    Ecrire(‘Le resultat est : ’,R) ;

                Sinon

                    Ecrire(‘votre choix est mauvais … !’) ;

              Finsi

          Fincas

               Si (Q<1) ou (Q>4) alors

                      Ecrire(‘Erreur de choix’) ;

               Sinon

                           R <--  TN \*TR ;

                           PAR <--  TN+R ;

                             Ecrire(‘Le prix à payer est :’,PAP) ;

                Finsi

Fin

***Exercice 9 :***

Ecrire un **algorithme** qui compare deux dates représentées chacune par trois entiers.

***Solution :***

***Algorithme  compare\_date;***

Var

            j1,m1,a1,j2,m2,a3: réels ;

Debut

                Ecrire(‘\*\*\*\*\*\*\*\*\*\*\*\*date 1 \*\*\*\*\*\*\*\*\*\*\*‘) ;

                Ecrire(‘jour: ‘) ;

                Lire(j1) ;

                Ecrire(‘Mois : ‘) ;

                Lire(m1) ;

    Ecrire(‘Année : ‘) ;

                Lire(a1) ;

                Ecrire(‘\*\*\*\*\*\*\*\*\*\*\*\*date 2 \*\*\*\*\*\*\*\*\*\*\*‘) ;

                Ecrire(‘jour: ‘) ;

                Lire(j2) ;

                Ecrire(‘Mois : ‘) ;

                Lire(m2) ;

    Ecrire(‘Année : ‘) ;

                Lire(a2) ;

                Si  a1>a2  alors

                     Ecrire (‘ la date 1 est plus grande que la date 2 ’) ;

                Sinon si a1<a2 alors

                                 Ecrire(‘la date 2 est plus grande que la date 1 ’) ;

                          Sinon Si  m1>m2  alors

                                              Ecrire (‘ la date 1 est plus grande que la date 2 ’) ;

                                       Sinon si m1<m2 alors

                                                       Ecrire(‘la date 2 est plus grande que la date 1 ’) ;

                                                 Sinon Si  j1>j2  alors

                                                                         Ecrire (‘ la date 1 est plus grande que la date 2 ’) ;

                                                               Sinon si j1<j2 alors

                                                                        Ecrire(‘la date 2 est plus grande que la date 1 ’) ;

                                                             Finsi

                                                     Finsi

                                         Finsi

                               Finsi

              finsi

fin

***Instructions répétitives : les boucles***

***Exercice 1 :***

Ecrire un **algorithme** permettant de lire une suite de nombres réels sur le clavier. Le dernier élément à lire est un zéro.

L’algorithme doit afficher le plus petit élément de la suite ainsi que la somme des éléments lus.

***Solution :***

***Algorithme  lire\_une\_suite;***

Var

            a,s : réels ;

Debut

              Repeter

                        Ecrire(‘entrer un  nombre: ‘) ;

                        Lire(a) ;

                        S<-- s+a ;

              Jusqu'à (a=0)

              Ecrire(‘la somme des nombres est :  ‘,s) ;

fin

***Exercice 2 :***

Ecrire un **algorithme** qui lit les noms des athlètes désirant participer à un marathon et s’arrête lorsqu’il rencontre deux noms consécutifs identiques.

L’algorithme doit afficher , à la fin , le nombre de participants.

***Solution :***

***Algorithme  lire\_une\_suite;***

Var

            Nom1,nom2 : chaine de 30 caractères ;

            Compteur  i  :entier ;

Debut

                Ecrire(‘entrer un nom d’athlète   : ‘) ;

                Lire(nom1) ;

                Ecrire(‘entrer un nom d’athlète   : ‘) ;

                Lire (nom2) ;

                i <--  2 ;

              Tantque(nom1 <> nom2)

                        Nom1 <--  nom2 ;

                        Ecrire(‘entrer un  nom d’athlète  : ‘) ;

                        Lire(nom2) ;

                        i<-- i+1;

              Fintantque

              Ecrire(‘le nombre des participants est :  ‘,i) ;

fin

***Exercice 3 :***

Dans un cinéma , on désire calculer la moyenne  d’âges de personnes s’intéressant à voir un film particulier.

Pour finir, l’utilisateur doit répondre par ‘N’ à la question posée par l’**algorithme** :

‘’ Encore une autre personne (O/N) ? ’’

Et on doit afficher la moyenne d’âges des personnes à l’ecran.

***Solution :***

***Algorithme  moyenne\_age;***

Var

            i ,age,som :entier ;

Debut

                i <--  0 ;

                age <--  0 ;

                Repeter

                        Ecrire(‘ entrer l’age du personne ’) ;

                        Lire(‘age’ ) ;

                          i<--  i+1 ;

                        Som <--  som+ age ;

                          Ecrire(‘Encore une autre personne (O/N) ?’)

                          Lire(rep) ;

                Jusqu'à (rep=’N’)

              Ecrire(‘la moyenne d’âges des personnes  est :  ‘,som/i) ;

fin

***Exercice 4 :***

Ecrire un algorithme permettant de calculer la factorielle d’un nombre entier positif N.

N ! = 1\*2\*3\*……..\*N

***Solution :***

***Algorithme  factorielle;***

Var

              i ,N,fact:entier ;

Debut

                fact<--  1 ;

                pour i=1 à N faire

                             fact<-- fact\*i ;

                Finpour  i

              Ecrire(‘ la factorielle de  ’ ,N,’  est   : ‘,fact  ) ;

fin

***Exercice 5 :***

Pour avoir une idée sur le niveau des éleves d’une classe, on a décidé de calculer la moyenne  de la classe à partir des moyennes générales de tous les élèves qui sont au nombre de 30.

         Moy =

Ecrire un **algorithme** de résolution.

***Solution :***

***Algorithme  moyenne\_classe;***

Var

              i   :entier ;

                NT ,som: réels

const

                nb=30 ;

Debut

                Som <--  0 ;

                pour i=1 à N faire

                            Ecrire(‘ entrer la note N°’,i,’ :’) ;

                            Lire(NT) ;

                            Som<-- som+NT ;

                Finpour  i

              Ecrire(‘ la moyenne de la classe est : ‘,som/nb ) ;

fin

***Exercice 6 :***

Ecrire un algorithme qui permet de retrouver le maximum, le minimum ainsi que la somme d’une liste de nombres positifs saisis par l’utilisateur. La fin de la liste est indiquée par un nombre négatif. La longueur  de la liste n’est pas limitée.

Exemple :              si la liste des éléments est : 7   3   20     15    2      6    5   -1

                                Le maximum est 20  , le minimum est 2 .

***Solution :***

***Algorithme  moyenne\_age;***

Var

            i ,n,som :entier ;

Debut

                Ecrire(‘ entrer un nombre ’) ;

              Lire(‘n’ ) ;

                max<--  n ;

                min <--  n ;

                som <--  n ;

                Tantque n>0  faire

                        Ecrire(‘ entrer  un nombre’) ;

                        Lire(‘n’ ) ;

                         Si min>n  alors

                                Min <--  n ;

                         Finsi

             Si max<n  alors

                                 Max <--  n ;

                          Finsi

                fintantque

              Ecrire(‘Le maximum est  ‘,max ,’ le minimum est ‘,min) ;

fin

***Les variables dimensionnées (Les tableaux)***

***Exercice 1 :***

Ecrire un **algorithme** permettant  d’entrer cinq valeurs réelles au clavier,les stocker dans un tableau, calculer leur somme et les afficher avec leur somme à l’ecran.

***Solution :***

***Algorithme  tableau\_somme;***

Var

              V   : tableau [1..5] de réels ;

              S   :  réel ;

i  :entier;

Debut

**(\*lecture  des élements  du tableau\*)**

                Pour i <--  1 à 5 faire

                               Ecrire(‘entrer l’element  N° ’,i);

                               Lire(V[i]) ;

                Finpour i

**(\*calcul de la somme des élements du tableau \*)**

                S <--  0 ;

Pour i <--  1 à 5 faire

                               S <--  S + V[i] ;

                Finpour i

**(\*afficher des éléments du tableau \*)**

                Pour i <--  1 à 5 faire

                               Ecrire(‘l’’element  N° ’,i,’est : ‘,V[i]);

                Finpour i

                Ecrire(‘la somme des éléments du tableau est :‘,S) ;

fin

***Exercice 2 :***

Ecrire un algorithme permettant  de saisir et d’afficher N éléments d’un tableau.

***Solution :***

***Algorithme  saisie\_affichage;***

Var

              T   : tableau [1..100] de réels ;

              N,i   :  réel ;

Debut

                Ecrire(‘entrer le nombre d’éléments du tableau :’) ;

                Lire(N) ;

**(\*lecture  des élements  du tableau\*)**

                Pour i <--  1 à N faire

                               Ecrire(‘entrer l’element  N° ’,i);

                               Lire(T[i]) ;

                Finpour i

**(\*afficher des éléments du tableau \*)**

                Pour i <--  1 à N faire

                               Ecrire(‘l’’element  T[’,i,’] est : ‘,T[i]);

                Finpour i

Fin

***Exercice 3 :***

Ecrire un **algorithme** permettant  de calculer la somme,produit et moyenne des  éléments d’un tableau.

***Solution :***

***Algorithme  somme\_produit\_moyenne;***

Var

              T   : tableau [1..100] de réels ;

              N,i   :  entiers ;

                S,P,M :  réels ;

Debut

                Si N=0 alors

                               Ecrire(‘le tableau est vide ‘) ;

                Sinon

**S <--  0 ;**

**P <--  1 ;**

                         Pour i <--  1 à N faire

                               S <--  S+T[i] ;

                               P <--  P \* T[i] ;

                           Finpour i

                           M <--  S/N ;

                            Ecrire(‘la somme des éléments du tableau est : ‘,S);

                Ecrire(‘le produit des éléments du tableau est : ‘,P);

                Ecrire(‘la  moyenne des éléments du tableau est : ‘,M);

Finsi

Fin

***Exercice 4 :***

Ecrire un **algorithme** permettant  de consulter un  élément d’un tableau.

***Solution :***

***Algorithme  consultation;***

Var

              T   : tableau [1..100] de réels ;

              N,P   :  entiers ;

Debut

                Si N=0 alors

                        Ecrire(‘le tableau est vide ‘) ;

                Sinon

                        Ecrire(‘entrer l’’indice de l’’élément à consulter :’) ;

                        Lire(P) ;

                        Si (P<1) ou (P>N) alors

                                Ecrire(‘Position hors limites du tableau ‘)

                        Sinon

                                Ecrire(‘l’’élément à consulter est :’,T[P]) ;

                         Finsi

                Finsi

Fin

***Exercice 5 :***

Ecrire un **algorithme** permettant  de chercher toutes les occurrences d’un  élément  dans un tableau.

***Solution :***

***Algorithme  recherche\_toutes\_occurences ;***

Var

              T   : tableau [1..100] de réels ;

              N,i   :  entiers ;

                X : réel ;

                Existe : booléen ;

Debut

                Si N=0 alors

                               Ecrire(‘le tableau est vide ‘) ;

                Sinon

                        Ecrire(‘entrer la valeur de l’’élément à chercher  :’) ;

                        Lire(X) ;

                        Existe <--  Faux ;

                        Pour i <--  1  à N  Faire

                                Si T[i] = X alors

                                               Existe <--  Vrai ;

                                               Ecrire(‘l’’élément à chercher apparait à la position : ‘,i) ;

                                Finsi

                         Finpour i

                         Si Existe = Faux alors

                                   Ecrire(‘l’’élément n’’apparait pas dans ce tableau ‘) ;

                           Finsi

                 Finsi

Fin

***Exercice 6 :***

Ecrire un **algorithme** permettant  de chercher  la première occurrence d’un élément dans un tableau.

***Solution :***

***Algorithme  recherche\_première\_occurence ;***

Var

T   : tableau [1..100] de réels ;

             P, N,i   :  entiers ;

                X : réel ;

                Existe : booléen ;

Debut

                Si N=0 alors

                               Ecrire(‘le tableau est vide ‘) ;

                Sinon

                        Ecrire(‘entrer la valeur de l’’élément à chercher  :’) ;

                        Lire(X) ;

                        Existe <--  Faux ;

                        i <--  1 ;

                        tantque  (i<=N) et (Existe=Faux) Faire

                                Si T[i] = X alors

                                               Existe <--  Vrai ;

                                               P<--  i ;

                                Sinon

                                               i ßi+1 ;

                                Finsi

                       Fintantque

                       Si Existe = vrai alors

                                Ecrire(‘la première occurrence de l’élément dans ce tableau est :’,P) ;

                       Sinon

                                Ecrire(‘l’’élément n’’apparait pas dans ce tableau ‘) ;

                      Finsi

                Finsi

Fin

***Exercice 7 :***

Ecrire un **algorithme** permettant  de chercher  la dernière occurrence d’un élément dans un tableau.

***Solution :***

***Algorithme  recherche\_derniere\_occurence ;***

*Var*

*T   : tableau [1..100] de réels ;*

*P, N,i   :  entiers ;*

*X : réel ;*

*Existe : booléen ;*

*Debut*

*Si N=0 alors*

*Ecrire(‘le tableau est vide ‘) ;*

*Sinon*

*Ecrire(‘entrer la valeur de l’’élément à chercher  :’) ;*

*Lire(X) ;*

*Existe <--  Faux ;*

*i <--  N ;*

*tantque  (i>=1) et (Existe=Faux) Faire*

*Si T[i] = X alors*

*Existe <--  Vrai ;*

*P<--  i ;*

*Sinon*

*i <-- i-1 ;*

*Finsi*

*Fintantque*

*Si Existe = vrai alors*

*Ecrire(‘la dernière occurrence de l’élément dans ce tableau est :’,P) ;*

*Sinon*

*Ecrire(‘l’’élément n’’apparait pas dans ce tableau ‘) ;*

*Finsi*

*Finsi*

*Fin*

***Exercice 8 :***

Ecrire un **algorithme** permettant  de calculer le nombre de fois pour lesquelles un  élément  apparait dans un tableau.

***Solution :***

***Algorithme  frequence;***

Var

              T   : tableau [1..100] de réels ;

              N,i,F   :  entiers ;

                X : réel ;

Debut

                Si N=0 alors

                               Ecrire(‘le tableau est vide ‘) ;

                Sinon

                        Ecrire(‘entrer la valeur de l’’élément à chercher  :’) ;

                        Lire(X) ;

                        F <--  0 ;

                        Pour i <--  1  à N  Faire

                                Si T[i] = X alors

                                               F <--  F+1 ;

                                Finsi

                         Finpour i

                                   Ecrire(‘l’’élément  apparait : ‘, F,’fois dans ce tableau ‘) ;

                 Finsi

Fin

***Exercice 9 :***

Ecrire un **algorithme** permettant  d’ajouter un élément a la fin d’un tableau.

***Solution :***

***Algorithme  Ajout;***

Var

              T   : tableau [1..100] de réels ;

              N :  entiers ;

                X : réel ;

                Rep : caractère ;

Debut

                Ecrire(‘entrer la valeur de l’’élément à ajouter   :’) ;

                Lire(X) ;

                 Ecrire(‘Confirmer l’ajout (O/N) :’) ;

                 Lire(Rep) ;

                Si Rep =’O’ alors  **(\*la valeur ‘o’ pour ‘oui’ ! et ‘N’ pour ‘Non’ \*)**

                               N<--  N+1 **;(\* il y aura un élément en plus \*)**

                                T[N] <--  X ;

                 Finsi

Fin

***Exercice 10 :***

Ecrire un **algorithme** permettant  de modifier  un élément  dans un tableau.

***Solution :***

***Algorithme Modification;***

Var

              T   : tableau [1..100] de réels ;

              N ,P:  entiers ;

                X : réel ;

                Rep : caractère ;

Debut

                Si N=0 alors

                               Ecrire(‘le tableau est vide ‘) ;

                Sinon

                Ecrire(‘entrer l’indice de l’’élément à  modifier :’) ;

                Lire(P) ;

                      Si (P<1) ou (P>N) alors

                                Ecrire(‘Position hors limites du tableau ‘)

                       Sinon

                                Ecrire(‘L’ancienne valeur dans cette position est :’,T[p]) ;

                                Ecrire(‘Entrer la nouvelle valeur :’) ;

                                Lire(X) ;

                                Ecrire(‘Confirmer  la modification (O/N) ‘) ;

                                Lire(Rep) ;

                                Si Rep=’O’ Alors

                                          T[P]<--  X ;

                                Finsi

                       Finsi

                Finsi

Fin

***Exercice 11 :***

Ecrire un **algorithme** permettant  d’insérer  un élément  dans un tableau  (au début , au milieu ou  à la fin).

***Solution :***

***Algorithme  insertion;***

Var

              T   : tableau [1..100] de réels ;

              N ,P,i:  entiers ;

                X : réel ;

                Rep : caractère ;

Debut

                Si N=0 alors

                               Ecrire(‘le tableau est vide ‘) ;

                Sinon

                Ecrire(‘entrer la valeur de l’’élément à insérer  :’) ;

                Lire(X) ;

                 Ecrire(‘ Entrer la position d’insertion :’) ;

                 Lire(P) ;

                      Si (P<1) ou (P>N) alors

                                Ecrire(‘Position hors limites du tableau ‘)

                       Sinon

                                Ecrire(‘Confirmer l’insertion (O/N) ‘) ;

                                Lire(Rep) ;

                                Si Rep=’O’ Alors

                                               N <--  N+1 ;

                                               Pour i <--  N à P+1 Faire

                                                               T[i] <--  T[i-1] ;

                                               Finpour i

                                               T[P]<--  X ;

                                Finsi

                       Finsi

                Finsi

Fin

***Exercice 12 :***

Ecrire un **algorithme** permettant  de supprimer  un élément  dans un tableau.

***Solution :***

***Algorithme  suppression;***

Var

              T   : tableau [1..100] de réels ;

              N ,P,i:  entiers ;

                Rep : caractère ;

Debut

                Si N=0 alors

                               Ecrire(‘le tableau est vide ‘) ;

                Sinon

                                Ecrire(‘entrer l’indice  de l’’élément à supprimer  :’) ;

                                Lire(P) ;

                               Si (P<1) ou (P>N) alors

                                        Ecrire(‘Position hors limites du tableau ‘)

                              Sinon

                                         Ecrire(‘la valeur dans cette position est :‘,T[P]) ;

                                         Ecrire(‘Confirmer la suppression  (O/N) ‘) ;

                                         Lire(Rep) ;

                                         Si Rep=’O’ Alors

                                                 N <--  N+1 ;

                                                 Pour i <--  P à N-1 Faire

                                                               T[i] <--  T[i+1] ;

                                                 Finpour i

                                                  N <--  N-1; **(\*il y aura un élément en moins\*)**

                                           Finsi

                            Finsi

                Finsi

Fin

***Exercice 13 :***

Ecrire un **algorithme** permettant  de trier  par ordre croissant les éléments  d’un tableau.

***Solution :***

***Algorithme  tri\_Croissant;***

Var

              T   : tableau [1..100] de réels ;

              N ,i,j:  entiers ;

                Aux: réel ;

Debut

                Si N=0 alors

                               Ecrire(‘le tableau est vide ‘) ;

                Sinon

                          Pour i <--  1 à N-1 Faire

                                               Pour j <--  i+1 à N  Faire

                                                           Si T[i] > T[j]  alors

                                                                    Aux <--  T[i] ;

                                                                    T[i] <--  T[j] ;

                                                                    T[j] <--  Aux ;

                                                               Finsi

                                             Finpour j

                           Finpour i

                Finsi

Fin

***Exercice 14 :***

Ecrire un **algorithme** permettant  de trier  par ordre décroissant les éléments  d’un tableau.

***Solution :***

***Algorithme  tri\_Décroissant;***

Var

              T   : tableau [1..100] de réels ;

              N ,i,j:  entiers ;

                Aux: réel ;

Debut

                Si N=0 alors

                               Ecrire(‘le tableau est vide ‘) ;

                Sinon

                          Pour i <--  1 à N-1 Faire

                                               Pour j <--  i+1 à N  Faire

                                                           Si T[i] < T[j]  alors

                                                                    Aux <--  T[i] ;

                                                                    T[i] <--  T[j] ;

                                                                    T[j] <--  Aux ;

                                                               Finsi

                                             Finpour j

                           Finpour i

                Finsi

Fin

***Exercice 15 :***

Ecrire un **algorithme** permettant  de fusionner les éléments  de deux tableaux T1 et T2 dans un autre tableau T.

N.B :

 N : nombre des éléments  du tableau T1

M : nombre des éléments du tableau T2

***Solution :***

***Algorithme  fusion\_deux\_tableaux;***

Var

              T1,T2   : tableau [1..100] de réels ;

                T   : tableau [1..200] de réels ;

              N ,M,i:  entiers ;

Debut

                Si  (N=0)  et (M=0) alors

                               Ecrire(‘le tableau est vide ‘) ;

                Sinon

                          Pour i <--  1 à N Faire

                                T[i] <--  T1[i] ;

                           Finpour i

                           Pour i <--  1 à M Faire

                                T[N+i] <--  T2[i] ;

                           Finpour i

                Finsi

Fin

***Exercice 16 :***

Ecrire un **algorithme** permettant  de saisir les données d’un tableau à deux dimensions (10,4), de faire leur somme, produit et moyenne et de les afficher avec les résultats de calcul à l’écran.

***Solution :***

***Algorithme  tableau\_deux\_dimension;***

Var

              T : tableau [1..10,1..4] de réels ;

              I,j:  entiers ;

                S,P,M : réels ;

Debut

**(\*saisie des éléments du tableau \*)**

                 Pour i <--  1 à 10 Faire

                                 Pour j<--  1 à 4 Faire

                                              Ecrire(‘entrer l’’element T[‘,i,’,’,j,’] :’;

                                                Lire(T[i,j]) ;

                                Finpour j;

                Finpour i

**(\*calcul de la somme ,produit et moyenne \*)**

S <-- 0 ;

P <--  1 ;

                Pour i <--  1 à 10 Faire

                                Pour j<--  1 à 4 Faire

                                             S <--  S+T[i,j] ;

                                             P <--  P + T[i,j] ;

                                Finpour j;

                Finpour i

                M <--  S/40 ; **(\*40 : nombre d’élément du tableau = 10x4 \*)**

**(\* Affichage des éléments du tableau et des résultats \*)**

                Pour i <--  1 à 10 Faire

                                Pour j<--  1 à 4 Faire

                                             Ecrire(‘l’’élément T[‘,i,’,’,j,’] = ‘, T[i,j]);

                                Finpour j;

                Finpour i

                Ecrire(‘la somme des éléments du tableau est :’,S) ;

                Ecrire(‘le produit des éléments du tableau est :’,P) ;

                Ecrire(‘la moyenne  des éléments du tableau est :’,M) ;

Fin

***Exercice 17 :***

Ecrire un **algorithme** qui calcule la somme des éléments de la diagonale d’une matrice carrée M(n,n) donnée.

***Solution :***

***Algorithme  diagonale\_de\_matrice;***

Const

                N=8 ;

Var

              M : tableau [1..8,1..8] d’entiers;

              i,j:  entiers ;

                Sdiag: entiers;

Debut

**(\*saisie des éléments de la matrice\*)**

                 Pour i <--  1 à n  Faire

                                 Pour j<--  1 à n Faire

                                              Ecrire(‘entrer l’’element M[‘,i,’,’,j,’] :’;

                                                Lire(M[i,j]) ;

                                Finpour j;

                Finpour i

**(\*calcul de la somme  des éléments de la diagonale  \*)**

Sdiag <--  0 ;

                Pour i <--  1 à n Faire

                                Sdiag <--  Sdiag +M[i,j] ;

                Finpour i

                Ecrire(‘la somme des éléments  de la diagonale est :’,Sdiag) ;

Fin

***Exercice 18 :***

Ecrire un **algorithme** permettant d’effectuer le produit  des matrices A(n,m) et B(m,p) .

n ,m et p données (par exemple n=4,m=5,p=3 ).

N.B :

            Pour pouvoir faire le produit de deux matrices, il faut absolument que le nombre de colonnes

            de la première soit égal au nombre de lignes de la deuxième.

***Solution :***

***Algorithme  produit\_matrices;***

Const

            n=4;

m=5;

p=3;

Var

              A : tableau [1..n,1..p] de réels;

              B : tableau [1..p,1..m] de réels;

              C : tableau [1..n,1..m] de réels;

  i,j:  entiers ;

Debut

**(\*lecture  des éléments des deux matrices\*)**

                 Pour i <--  1 à n  Faire

                                 Pour j<-- 1 à p Faire

                                              Ecrire(‘entrer l’’element A[‘,i,’,’,j,’] :’);

                                                Lire(A[i,j]) ;

                                Finpour j;

                Finpour i

                 Pour i <--  1 à p  Faire

                                 Pour j<--  1 à m Faire

                                              Ecrire(‘entrer l’’element B[‘,i,’,’,j,’] :’);

                                                Lire(B[i,j]) ;

                                Finpour j;

                Finpour i

**(\*calcul de produit des deux matrices\*)**

                Pour i <--  1 à n  Faire

                                 Pour j<--  1 à m Faire

                                              C[i,j] <--  0 ;

                                               Pour k <--  1 à p faire

                                                               C[i,j] <--   C[i,j] + A[i,k] \*B[k,j] ;

                                                Finpour k

                                Finpour j;

                Finpour i

**(\*Affichage de la matrice produit\*)**

                 Pour i <--  1 à n  Faire

                                 Pour j<--  1 à m Faire

                                              Ecrire(‘l’’element C[‘,i,’,’,j,’] =’,C[i,j]);

                                Finpour j;

                Finpour i

Fin

***Exercice 19 :***

**Ecrire un algorithme permettant  de construire dans une matrice carrée P et d’afficher le triangle de PASCAL  de degré N.**

**N.B :**

**On poura utiliser cette relation pour les éléments de triangle de PASCAL :**

**Pi,j = Pi-1,j-1+ Pi-1,j**

**Exemple : triangle de pascal de degré 5 :**

**N=0              -        1**

**N=1              -        1     1**

**N=2              -        1     2       1**

**N=3              -        1     3       3       1**

**N=4              -        1     4       6       4       1**

**N=5              -        1     5      10     10    15    1**

***Solution :***

***Algorithme  triangle\_pascal;***

Var

               P: tableau [1..100,1..100] de réels;

  i,j,n:  entiers ;

Debut

                Ecrire(‘entrer l’ordre du  triangle de pascal que vous voulez :’) ;

                Lire(n) ;

**(\*remplissage du triangle de Pascal\*)**

                P[1,1] <--  1 ;

                 Pour i <-- 2 à n+1  Faire

                               P[i,1] <--  1 ;

                                 Pour j<-- 2 à i-1 Faire

                                           P[i,j]  <-- P[i-1,j-1] + P[i-1,j]

                                Finpour j;

                                P[i,j] <-- 1 ;

                Finpour i

**(\* affichagedu triangle de Pascal\*)**

                 Pour i <--  1 à n+1  Faire

                                 Pour j<--  1 à i Faire

                                              Ecrire(P[i,j] ,’           ‘);

                                Finpour j;

**(\*retour a la ligne \*)**

                Finpour i

Fin

***Les enregistrements***

***Exercice 1 :***

Dans la recherche d’employés, un organisme a exigé d’identifier les candidats par les caractéritiques suivantes :

-Nom

-Prénom

-Date de naissance : jour,mois,année

-Lieu de naissance : ville,province,pays

-Etat civil (marié ou non)

-Nombre d’enfants

-Nationalité

-Adresse : avenue,ville,pays,numero,code postal,téléphone

-Diplôme

-Stage

-Etablissement : Etablissement1,Etablissement2, Etablissement3,Etablissement4.

a-      Ecrire l’**enregistrement** Identité.

b-      Ecrire une parte d’**algorithme** permettant de déclarer une variable de type identité et lui affecter l’identité d’une personne.

***Solution :***

**A-**

Type

**Date=enregistrement**

                               Jour : entier ;

                               Mois : entier ;

                               Année : entier ;

**FinEnregistrement**

**Lieu=enregistrement**

                               Ville : chaine de 15 caractères ;

                               Province : chaine de 25 caractères ;

                               Pays : chaine de 20 caractères ;

**FinEnregistrement**

**Domicile= enregistrement**

                               Avenue,ville,pays : chaine de 20 caractères ;

                               Numero,code\_postale entier long ;

                               Telephone : chaine de 10 caractères ;

**FinEnregistrement**

                Entreprise = enregistrement

                               Etab\_1, Etab\_2, Etab\_3, Etab\_4 : chaine de 25  caractères ;

**FinEnregistrement**

**Identite=enregistrement**

                               Nom,prenom,nationalité,diplôme : chaine de 40 caractères ;

                               Date\_naissance : date ;

                               Lieu\_naissance : lieu ;

                               Etat\_civil,stage : booléen ;

                               Adresse : domicile ;

                               Etablissement :entreprise ;

**FinEregistrement**

**B-**

Type

Var

                Employe :identite ;

Début

**Avec** Employe **faire**

                               Nom ß ‘’Adil’’ ;

               Prenomß ‘’Adil’’ ;

                nationalite ß ‘’Marocaine’’ ;

**avec**  Date\_naissance **Faire**

                               jour ß 20 ;

                               mois ß 5 ;

                               annee ß 1964 ;

**Fin avec**

**avec** Lieu\_naissance **Faire**

                               villeß ‘’Tanger’’ ;

                               paysß ‘’Maroc’’ ;

                               provinceß ‘’Tanger’’ ;

**Fin avec**

                Etat\_civil ß vrai ;

                Stage ß vrai ;

                Nbr\_enf ß 3 ;

**Avec**adresse **faire**

                               Code\_postale ß 32500 ;

                               Avenue ß ‘’Mohamed V’’ ;

                               Ville ß ‘’casablanca’’ ;

                               Pays ß ‘’Maroc’’ ;

                               Numero ß 71 ;

                               Telephone ß ‘’0665222222’’ ;

**Fin avec**

**Avec** etablissement **faire**

                               Etab\_1 ß ‘’Sony’’ ;

                               Etab\_2 ß ‘’Samsung’’ ;

                               Etab\_3 ß ‘’Thomson’’ ;

                               Etab\_4 ß ‘’Motorola’’ ;

**Fin avec**

**Fin avec**

Fin

***Exercice 2 :***

Dans un repère orthonormé, un point est connu  par deux coordonnées X et Y. on peut le                       représenter en programmation par un enregistrement a deux champs.

-Ecrire un **algorithme** permettant de :

-Déclarer un **enregistrement** « point »,

-Introduire les coordonnées de deux points P1 et P2,

-Afficher les deux points P1 et P2,

-Calculer et afficher la distance entre ces deux points,

-Introduire au clavier un déplacement dx sur l’axe  des X et dy sur l’axe des y ,

-Déplacer les deux points en utilisant les déplacements dx et dy,

-Afficher les deux points P1 et P2 dans les nouvelles positions.

***Solution :***

***Algorithme points ;***

Type

**Point=enregistrement**

                               X  : entier ;

                               Y  : entier ;

**FinEnregistrement**

Var

                P1,P2 :point ;

                dx,dy : entier ;

                d : réel ;

Debut

**(\* Saisie des coordonnées des deux points  \*)**

Ecrire(‘Entrer l’abscisse  du premier point P1 : ‘) ;

                Lire(P1.X) ;

     Ecrire(‘Entrer l’ordonnée  du premier point P1 : ‘) ;

                Lire(P1.Y) ;

                Ecrire(‘Entrer l’abscisse  du deuxième point P2 : ‘) ;

                Lire(P2.X) ;

    Ecrire(‘Entrer l’ordonnée  du deuxième  point P2 : ‘) ;

                Lire(P2.Y) ;

**(\*Affichage des deux points P1 et P2\*)**

                Aller à (P1.X,P1.Y) ;

                Ecrire(‘\*’) ;

                Aller à (P2.X,P2.Y) ;

                Ecrire(‘\*’) ;

**(\*calcul de la distance entre les points P1 e P2\*)**

                d<-- sqrt((P2.X – P1.X)^2 + (P2.Y – P1.Y)^2) ;

                Ecrire (‘La distance entre les deux points P1 et P2 est :’,d) ;

**(\*Saisie des déplacements selon X et Y\*)**

                Ecrire (‘entrer le deplacement selon OX :’) ;

                Lire(dx) ;

                Ecrire (‘entrer le deplacement selon OY :’) ;

                Lire(dy) ;

**(\*Déplacement et affichage des  deux points \*)**

                P1.X <--  P1.X + dx ;

    P1.y <--  P1.y + dy ;

                P2.X <--  P2.X + dx ;

                P2.y <--  P2.y+ dy ;

**(\*Affichage des deux points dans les nouvelles positions\*)**

Aller à (P1.X,P1.Y) ;

                Ecrire(‘\*’) ;

                Aller à (P2.X,P2.Y) ;

                Ecrire(‘\*’) ;

Fin

***Exercice 3 :***

Ecrire un **algorithme** permettant de :

Déclarer un tableau nommé « courbe » de N point et chaque point défini sous forme d’enregistrement.

Saisir les coordonnées de tous les points de la courbe,

Tracer la courbe à l’ecran point par point en utilisant des étoiles

***Solution :***

***Algorithme points ;***

Type

**Point=enregistrement**

                               X  : entier ;

                               Y  : entier ;

**FinEnregistrement**

Var

                Courbe : tableau[1..100] de point ;

                N,i: entier ;

Debut

**(\* Saisie des points  de la courbe \*)**

Ecrire(‘Entrer le nombre de points de la courbe : ‘) ;

                Lire(N) ;

                Pour i <--  1 à N faire

                  Ecrire(‘Entrer l’abscisse du point N° ,’i’,’: ‘) ;

                              Lire(courbe[i].X) ;

                  Ecrire(‘Entrer l’ordonnée du point N° ,’i’,’: ‘) ;

                              Lire(courbe[i].Y) ;

                Fin pour i

**(\*traçage de la courbe point par point \*)**

Pour i <--  1 à N faire

                   Aller à (courbe[i].X, courbe[i].Y) ;

                               Ecrire(‘\*’) ;

                Fin pour i

Fin

***Exercice 4 :***

Dans  une **banque** un client est connu par :

-Nom

-Prénom

-Adresse

-Numéro de compte

-solde

a-      Ecrire l’**enregistrement** client.

b-      Ecrire un **programme** permettant :

D’entrer les informations de tous les clients de la **banque**,(on considère NC :nombre de client)

De faire la somme d’argents totale STA et la moyenne d’argent Moy par client ,

D’afficher les données et résultats à l’écran.

***Solution :***

Type

**client=enregistrement**

                               nom : chaine de 15 caractères ;

                               Pr : chaine de 20 caractères ;

                               Adr: chaine de 35 caractères  ;

                               Num : Entier long ;

                               Sold : réel ;

**FinEnregistrement**

Var

                T : Tableau[1..50] de client ;

                i,NC : entier ;

                STA,Moy : réels ;

Debut

                Ecrire(‘programme de gestion d’une banque’) ;

**(\*saisie de données \*)**

Ecrire(‘entrer le nombre de clients : ’) ;

Lire(NC) ;

Pour i <--  1 à NC faire

                Ecrire(‘\*\*\*\*\*\*\*\*\*\*Information du client numéro  ‘,i,’ \*\*\*\*\*\*\*\* ’) ;

                Ecrire(‘entrer le nom du client :’) ;

               Lire(T[i].nom) ;

               Ecrire(‘entrer le prénom  du client :’) ;

               Lire(T[i].Pr) ;

               Ecrire(‘entrer le numéro du compte:’) ;

               Lire(T[i].num) ;

               Ecrire(‘entrer l’adresse du client :’) ;

               Lire(T[i].Adr) ;

               Ecrire(‘entrer le solde  du client :’) ;

               Lire(T[i].Sold) ;

Fin pour i ;

**(\*calcul\*)**

STA <--  0 ;

Pour  i<-- 1 à NC faire

                STA <--  STA+T[i].Sold ;

Fin pour i

Moy <--  STA/NC ;

**(\*affichage des r ésultats\*)**

Ecrire (‘la somme totale d’’argent est :’,STA) ;

Ecrire(‘la moyenne d’argent est :’,Moy) ;

Pour  i<-- 1 à NC faire

       Avec T[i] faire

                   Ecrire(‘information du client numéro :’,i) ;

                   Ecrire(‘Nom  :’,nom) ;

                   Ecrire(‘Prenom :’,pr) ;

                   Ecrire(‘adresse:’,adr) ;

                   Ecrire(‘numéro de compte :’,num) ;

                   Ecrire(‘solde :’,Sold) ;

      FinAvec

Fin pour i

Fin

***Les sous programmes (les procédures et fonctions)***

***Exercice 1 :***

Ecrire un sous **programme** qui reçoit un nombre réel, comme paramètre, teste s’il est négatif, positif ou nul et affiche le résultat à l’écran.

Prévoir un **algorithme** appelant ce sous programme.

***Solution :***

Procedure  signe(a :réel)();

Debut

                Si a>0 alors

                               Ecrire (‘le nombre est positif’)

                Sinon

                          Si a<0 alors

                                               Ecrire (‘le nombre est negatif’)

                          Sinon

                                               Ecrire (‘le nombre est nul’) ;

                          Finsi

                Finsi

fin

***Algorithme determination signe ;***

Var

                X : réel ;

Debut

                Ecrire(‘Entrer un nombre réel ‘) ;

                Lire(x) ;

                Signe(X)() ;

fin

***Exercice 2 :***

Ecrire un sous **programme** permettant de calculer la surface d’un trapèze.

Prévoir un algorithme appelant ce sous **programme**.

N.B :

                           S= (Gb+Pb)\*h/2

                             Avec :

                                               Gb : grande base du trapèze

                                               Pb : petite base du trapèze

                                               h :  la hauteur du trapèze

***Solution :***

Fonction surface (Gb,Pb,h :réel)( ) :réel;

Var

                S : réel ;

Debut

                S <-- (Gb+Pb)\*h/2 ;

                Retour (S) ;

fin

Algorithme surface\_trapeze;

Var

                X,Y,Z,T : réel ;

Debut

                Ecrire(‘Entrer  la grande base : ‘) ;

                Lire(X) ;

                Ecrire(‘Entrer  la petite base : ‘) ;

                Lire(Y) ;

                Ecrire(‘Entrer  la hauteur: ‘) ;

                Lire(Z) ;

                T  <--  surface(X ,Y,Z)() ;

                Ecrire(‘la surface de ce trapèze est :’,T) ;

fin

***Exercice 3 :***

On désire effectuer des opérations arithmétiques usuelles sur les nombres réels.

Ecrire un **algorithme** contenant les sous programmes suivants :

-Une **procédure** saisie d’un nombre réel,

-Une **procédure** affichage d’un nombre réel,

-Une **procédure** somme de deux nombres réels,

-Une **procédure** différence de deux nombre réels,

-Une **procédure** rapport de deux nombres réels,

-Une **fonction** racine carrée d’un nombre réel,

-Une **fonction**  carrée d’un nombre réel,

-Une **procédure** menu de gestion de toutes ces opérations.

***Solution :***

-          **Procédure saisie :**

Procedure saisie ()( A :réel) ;

Debut

                Ecrire(‘entrer un nombre reel :’);

                Lire(A) ;

fin

-          **Procédure affichge :**

Procedure affichage (A :réel)( ) ;

Debut

                Ecrire(‘le resultat est :’,A);

fin

-          **Procédure somme :**

Procedure somme (A ,B:réel)(S :réel ) ;

Debut

                S ß<--  A+B ;

fin

-          **Procédure différence :**

Procedure difference (A ,B:réel)( D :réel) ;

Debut

                D <--  A-B ;

fin

-          **Procédure produit :**

Procedure produit (A ,B:réel)( P :réel) ;

Debut

                P <--  A\*B ;

fin

-          **Procédure rapport:**

Procedure produit (A ,B:réel)( R :réel) ;

Debut

                R <--  A/B ;

fin

-          **Fonction  inverse:**

Fonction inverse (A :réel)( ) : réel;

Var

                I : réel ;

Debut

                I <-- 1/A;

                Retour (I) ;

fin

-          **Fonction racine carrée :**

Fonction racine (A :réel)( ) : réel;

Var

                R : réel ;

Debut

                R<-- sqrt(A);

                Retour (R) ;

fin

-          **Fonction  carrée :**

Fonction carree (A :réel)( ) : réel;

Var

                C : réel ;

Debut

                C<-- A\*A ;

                Retour (C) ;

fin

-          **Procédure menu  :**

Procedure menu()( ) ;

Var

                Choix : Caractère ;

Debut

            Ecrire(‘\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Menu \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*’);

Ecrire(‘ 1   \*\*\*\*\*\*\*\*\*\* \*\*Somme \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*’);

Ecrire(‘2 \*\*\*\*\*\*\*\*\*\*\*\*\*Différence\*\*\*\*\*\*\*\*\*\*\*\*\*\*’);

Ecrire(‘3 \*\*\*\*\*\*\*\*\*\*\*\*\* Produit\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*’);

Ecrire(‘4 \*\*\*\*\*\*\*\*\*\*\*\*\* Rapport \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*’);

Ecrire(‘5 \*\*\*\*\*\*\*\*\*\*\*\*\* Inverse \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*’);

Ecrire(‘6 \*\*\*\*\*\*\*\*\*\*\*\* Racine carrée \*\*\*\*\*\*\*\*\*\*\*\*’);

Ecrire(‘7 \*\*\*\*\*\*\*\*\*\*\*\* \* carrée \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*’);

Ecrire(‘8 \*\*\*\*\*\*\*\*\*\*\*\*\*\* Quitter \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*’);

Ecrire(‘\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*’);

            Ecrire(‘entre votre choix :’) ;

Lire(choix) ;

           Cas choix

‘1’ :        saisie()(X) ;

                saisie()(Y) ;

                somme(X,Y)(Z) ;

                affichage(Z)() ;

‘2’ :        saisie()(X) ;

                saisie()(Y) ;

                difference (X,Y)(Z) ;

                affichage(Z)() ;

‘3’ :        saisie()(X) ;

                saisie()(Y) ;

                produit (X,Y)(Z) ;

                affichage(Z)() ;

 ‘4’ :       saisie()(X) ;

                saisie()(Y) ;

                si Y=0 alors

                               Ecrire(‘calcul impossible !! ‘)

                Sinon

                               rapport(X,Y)(Z) ;

                               affichage(Z)() ;

                Finsi

‘5’ :        saisie()(X) ;

                si X=0 alors

                               Ecrire(‘calcul impossible !! ‘)

                Sinon

                               Z <--  inverse (X)() ;

                               affichage(Z)() ;

                Finsi

‘6’ :        saisie()(X) ;

                si X<0 alors

                               Ecrire(‘calcul impossible !! ‘)

                Sinon

                               Z <--  racine (X)() ;

                               affichage(Z)() ;

                Finsi

‘7’ :        saisie()(X) ;

                Z <--  carree(X)() ;

                affichage(Z)() ;

‘8’ :        Ecrire (‘A bien tôt ‘) ;

Sinon

Ecrire (‘votre choix est mauvais ‘) ;

      Fincas ;

Fin

-          **Algorithme  permettant d’appeler cette procédure menu de gestion de toutes les opérations :**

Algorithme calculs\_réels;

Var

                X,Y,Z : réel ;

Debut

                Tantque (3<4) **(\*boucle infinie\*)**

                               Menu()() ;

                Fintantque

fin

***Notion de récursivité***

***Exercice 1 :***

Ecrire une**fonctions récursive** qui calcule la factorielle d’un nombre entier N.

N.B :

                N ! = 1\*2\*3\*…….\*N

***Solution :***

Pour écrire la forme récursive d’une fonction, il faut chercher tout d’abord la récurrence mathématique, sinon la récursivité n’est pas utilisable.

Dans le cas du calcul de la factorielle, la récurrence mathématique peut se présenter comme suit :

                0 ! = 1

                N ! = N\*(N-1) !

Fonction factorielle (N : entier) () : entier;

Debut

                Si N=0 alors

                               Retour (1) ;

                Sinon

                                Retour(N\*factorielle(N-1)()) ;

                Finsi

fin

***Exercice 2 :***

Ecrire une **fonction récursive** qui calcule le Nième terme da la suite de Fibonacci définie comme suit :

F0 = 1

F1= 1

Fn =Fn-2+ Fn-1pour tout n >2

***Solution :***

Fonction fibonacci (N : entier) () : entier;

Debut

                Si N<= 2 alors

                               Retour (1) ;

                Sinon

                                Retour(fibonacci(N-2)() +fibonacci(N-1)());

                Finsi

fin

***Exercice 3 :***

|  |
| --- |
| Ecrire une **fonction récursive** qui calcule la somme de  N premiers nombres entiers naturels :  S=1+2+3+……+N |

***Solution :***

Pour écrire la forme**récursive** de la **fonction** somme, il faut chercher tout d’abord la récurrence mathématique.

                               S(0) = 0

                               S(N)=N+S (N-1)

Fonction somme (N : entier) () : entier;

Debut

                Si N=0 alors

                               Retour (N) ;

                Sinon

                                Retour( N+somme(N-1)());

                Finsi

fin

***Exercice 4 :***

Ecrire une**fonction récursive** qui calcule le Nième terme da la suite numérique  définie comme suit :

U0 = 2

U1= 2

U2= 2

Un =6\*Un-1+ 4\*Un-2- 5\*Un-3pour tout n >2

***Solution :***

Fonction suite (N : entier) () : entier;

Debut

                Si N< 3 alors

                               Retour (2) ;

                Sinon

                                Retour(6\*suite(N-1)() + 4\*suite(N-2)() – 5\*suite(N-3)()) ;

                Finsi

fin

***Exercice 5 :***

Ecrire une **fonction récursive** qui calcule les valeurs de polynôme d’Hermite Hn(x)  définie comme suit :

H0(x) = 1

H1(x) = 2\*x

Hn (x)=2\*x\*Hn-1(x)– 2(n-1)\*Hn-2(x)              pour tout n >1

***Solution :***

Fonction Hermite (N : entier,X : réel) () : entier;

Debut

                Si N= 0 alors

                               Retour (1) ;

                Sinon

                         Si N=1 alors

                                Retour(2\*X);

                         Sinon

                               Retour(2\*X\*hermite(N-1,X)() + 2\*(N-1)\*Hermite(N-2,X)()) ;

                        Finsi

                Finsi

fin