

# LIFAP1 – CC mi-parcours – Séquence 1

Contrôle Continu (Durée totale : 1h)

Mardi 12 novembre 2024

*Recommandations : Les documents, calculatrice, téléphone portable sont interdits. La qualité de l'écriture et de la présentation seront prises en compte dans la note finale. Vous veillerez à **respecter** les notations et les règles d'écriture des algorithmes vues en cours et en TD. Un soin tout particulier devra être apporté à l'écriture des entêtes des différents sous-programmes.*

NOM

.....

PRENOM

.....

Numéro Etudiant

.....

Groupe TD

.....

## Partie A – Algorithmique

En mathématiques, et notamment en combinatoire, les nombres de Delannoy dénombrent les chemins joignant deux points d'un réseau carré, en pas horizontaux, verticaux, et aussi diagonaux.

Le tableau ci-contre représente les premières valeurs des nombres de Delannoy. En ligne, l'abscisse  $n$  du point et en colonne son ordonnée  $m$ . On remarquera qu'il y a des 1 sur la première ligne et la première colonne du tableau et que les autres valeurs sont calculées en sommant les valeurs encadrées.

$n \backslash m$	0	1	2	3	4	5	6	7	8
0	1	1	1	1	1	1	1	1	1
1	1	3	5	7	9	11	13	15	17
2	1	5	13	25	41	61	85	113	145
3	1	7	25	63	129	231	377	575	833
4	1	9	41	129	321	681	1289	2241	3649
5	1	11	61	231	681	1683	3653	7183	13073
6	1	13	85	377	1289	3653	8989	19825	40081
7	1	15	113	575	2241	7183	19825	48639	108545
8	1	17	145	833	3649	13073	40081	108545	265729
9	1	19	181	1159	5641	22363	75517	224143	598417

Exemple  $129 = 41 + 63 + 25$

- 1- Ecrire l'algorithme d'une procédure `Saisie` qui demande à l'utilisateur et "retourne" une valeur entière comprise entre 3 et 10 inclus. La saisie sera recommencée tant que la valeur proposée n'appartient pas à l'intervalle  $[3 ; 10]$ .

Procédure saisie (val : entier)

Précondition : aucune

Donnée : aucune

Donnée / résultat : val

Description : saisit et retourne un entier entre 3 et 10

Variable locale : aucune

Début

    Faire

        Afficher( "Donnez une valeur entre 3 et 10 ")

        Saisir(val)

    Tant que (val<3 ou val>10)

Fin

- 2- Ecrire l'algorithme d'un sous programme `RemplitDelannoy` qui remplit un tableau 2D de taille  $MAX*MAX$  avec les premières valeurs des nombres de Delannoy jusqu'au rang  $n$  pour les lignes et jusqu'au rang  $m$  pour les colonnes,  $m$  et  $n$  étant passés en paramètres. On supposera qu'une constante  $MAX$  a été déclarée et vaut 100.

procédure `remplir_delannoy`(T tableau[ $MAX$ ][ $MAX$ ] d'entiers, TailleL entier, TailleC entier)

Précondition :  $0 < TailleL < MAX$ ,  $0 < TailleC < MAX$

Donnée : TailleL, TailleC

Donnée / résultat : T

Description : remplit le tableau T avec les valeurs de Delannoy

Variable locale :  $i, j$  : entiers

Début

Pour  $i$  allant de 0 à TailleL par pas de 1 faire

Pour  $j$  allant de 0 à TailleC par pas de faire

si ( $i=0$  ou  $j=0$ ) alors

$T[i][j] \leftarrow 1$

sinon

$T[i][j] \leftarrow T[i-1][j] + T[i-1][j-1] + T[i][j-1]$

Fin si

Fin pour

Fin pour

Fin

- 3- Ecrire l'algorithme d'un sous-programme `AfficheTab` qui affiche les  $n+1$  premières lignes et les  $m+1$  premières colonnes d'un tableau 2D de taille  $MAX*MAX$ ,  $m$  et  $n$  étant passés en paramètres.

procédure `affiche_tab`(T tableau[ $MAX$ ][ $MAX$ ] d'entiers, TailleL entier, TailleC entier)

Précondition :  $0 < TailleL < MAX$ ,  $0 < TailleC < MAX$

Donnée : TailleL, TailleC

Donnée / résultat : T

Description : affiche le tableau T

Variable locale :  $i, j$  : entiers

Début

Pour  $i$  allant de 0 à TailleL par pas de 1 faire

Pour  $j$  allant de 0 à TailleC par pas de faire

Afficher( $T[i][j]$ )

Fin pour

Afficher (saut de ligne)

Fin pour

Fin

- 4- Ecrire l'algorithme du programme principal qui fait choisir 2 entiers N et M compris entre 3 et 10, remplit le tableau 2D `Tab` avec les nombres de Delannoy et affiche le tableau obtenu. On utilisera tous les sous-programmes précédents et on fera attention de remplir et d'afficher uniquement les valeurs demandées.

```
Début
TL, TC, TabDelannoy tableau [MAX][MAX] d'entiers;
saisie(TL)
saisie(TC)
remplir_delannoy(TabDelannoy,TL,TC)
affiche_tab(TabDelannoy,TL,TC)
Fin
```

## Partie B – Langage C/C++

### Egalité entre sommes et produits

On souhaite écrire un programme permettant d'afficher la liste des couples de triplets d'entiers tels que la somme de leurs valeurs et le produit de leurs valeurs soient identiques.

Par exemple T1 = (1, 6, 6) et T2 = (2, 2, 9).

La somme des valeurs de T1 est 13 (= 1+6+6) et le produit 36 (= 1\*6\*6)

La somme des valeurs de T2 est 13 (= 2+2+9) et le produit 36 (= 2\*2\*9)

- 1- Ecrire en C++ un sous-programme `SommeProduit` permettant à partir de 3 entiers `a`, `b`, et `c` de calculer et "retourner" la somme **et** le produit de ces 3 entiers.

```
void som_prod (int a, int b, int c, int&s, int &p)
{
    s = a+b+c;
    p = a*b*c;
}
```

- 2- Ecrire en C++ un sous-programme `TripletsEgaux` qui à partir de 6 entiers passés en paramètres (2 triplets) retourne vrai si leurs sommes **et** leurs produits sont égaux. On utilisera le sous-programme précédent.

```
bool triplets_egaux (int a, int b, int c, int d, int e, int f)
{
    int s1,p1,s2,p2;
    som_prod(a,b,c,s1,p1);
    som_prod(d,e,f,s2,p2);
    return (s1==s2 && p1==p2);
}
```

- 3- Ecrire en C++ le programme principal permettant d'afficher les couples de triplets dont les sommes et les produits sont égaux. On n'affichera que pour les entiers inférieurs strictement à 10.

```
int main (void)
{
    int i,j,k,l,m,n;
    for(i=0;i<10;i++)
    {
        for (j=0;j<10;j++)
        {
            for (k=0;k<10;k++)
            {
                for (l=0;l<10;l++)
                {
                    for (m=0;m<10;m++)
                    {
                        for (n=0;n<10;n++)
                        {
                            if (triplets_egaux(i,j,k,l,m,n))
                                cout<<" "<<i<<" " , "<<j<<" " , "<<k<<" " , "<<l<<" " , "<<m<<" " , "<<n<<endl;
                        }
                    }
                }
            }
        }
    }
}
```