

Situation 1 : Combinais C_n^p

Activité : Saisir deux entiers n et p (0<p≤n) puis calculer et afficher $C_n^p = \frac{n!}{p!(n-p)!}$

Exemple : p=3, n=5 → C=10

Savoir faire : Structure itérative à condition d'arrêt : Syntaxe

Algorithme	Python
Traitement	
Tant que condition faire	
Traitement	Traitement
Fin Tant que	while condition :
Traitement	Traitement
Traitement	
Répéter	
Traitement	
Jusqu'à condition(s)	

Savoir- faire : Le type booléen

Définition : Le type booléen prend comme valeurs : vrai (True) ou faux (False)

Notation algorithmique / Notation Python

Algorithme	Python
non	not
et	and
ou	or

Table de vérité

A	B	Non(A)	A ou B	A et B
V	V			
V	F			
F	V			
F	F			

Exercice 1 : Compléter le tableau suivant :

Instruction	Résultat	Type
A ← (2>7) et (-2<5)		
B ← non (10≠-2)		

Exercice 2 : Exécuter manuellement ces instructions :

x ← 8 répéter x ← x+2 y ← x*2 jusqu'à y > 25	p ← 0 Tant que p < 5 faire p ← p+2 Fin Tant que	p ← 0 Tant que p > 5 faire p ← p+5 Fin Tant que	x ← 5 répéter x ← x+2 k ← x*2 jusqu'à k < 30
x= y=	p=	p=	x= k=
Nombre d'itérations :	Nombre d'itérations :	Nombre d'itérations :	Nombre d'itérations :

Exercice 3 : Saisir un temps t > 0 en secondes puis afficher son équivalent en heures, minutes et secondes.

Exemple : t=4000s → le programme affiche : 1 h 6 mns 40 s

Algorithme	Python
div	//
mod	%

Exercice 4 : Saisir un entier n composé de 3 chiffres puis calculer et afficher la somme de ces chiffres. (2 méthodes) **Exemple :** n=123 → Somme=6

Exercice 5 : Compléter le tableau suivant :

Instruction	Résultat	Type du résultat
15 / 3		
15 div 3		
15 mod 3		
14 div 3		
14 mod 3		
14 mod 3		

Exercice 6 : Saisir un entier n composé de 4 chiffres puis l'inverser. (2 méthodes) **Exemple :** n=1234 → Inv=4321

Savoir-faire : Les fonctions prédéfinies sur le type chaîne de caractères

Algorithme	Python	Description
Long(S)	len(S)	Retourne le nombre de caractères de la chaîne S.
Pos (S1, S2)	S2.find(S1)	Retourne la première position de S1 dans S2. Si S1 n'existe pas dans S2, retourne -1
Convch(X)	str(X)	Convertit un nombre X en chaîne.

Estnum(S)	S.isnumeric()	Retourne Vrai si la chaine S est convertible en numérique, Faux dans le cas contraire.
Valeur(S)	int(S)	Retourne la conversion d'une chaine S en numérique si c'est possible.
Sous_Chaine (S, D, F)	S[d:f]	Retourne une partie de la chaine S à partir de la position D jusqu'à la position F exclue.
Effacer (S, D, F)	S[:D]+S[F:]	Efface des caractères de la chaine S à partir de la position D jusqu'à la position F exclue.
Majus(S)	S.upper()	Convertit la chaine S en majuscule.

Exercice 7 : Compléter le tableau suivant :

Instruction	Résultat	Type du résultat
A ← 1+2		
B ← "1"+"23"		
C ← non ("b" ≥ "Baccalauréat")		

Exercice 8 : Saisir une chaîne de caractères ch de taille impaire et strictement supérieure ou égale à 3, puis afficher la chaîne contenant sa longueur, son premier caractère, le caractère du milieu ainsi que le dernier caractère.

Exemple : ch="poste" → Le programme affiche : "5pse"

Exercice 9 (Bac pratique 2005) : Saisir une chaîne ch non vide et supprimer les parenthèses, si elles existent ainsi que leur contenu. NB : On suppose que la chaîne contient au maximum deux parenthèses correctement utilisées (une ouvrante et une fermante).

Exemple : Lire un entier N strictement positif (N>0) et l'afficher. → Le programme affiche : Lire un entier N strictement positif et l'afficher.

Savoir-faire : Structures itératives complètes : Syntaxe

Algorithme	Python
Pour compteur de Vi à Vf faire	for element in sequence :
Traitement	Traitement La variable element est dite variable d'itération, elle prend successivement les différentes valeurs de la sequence
Finpour	Exemples for i in range(10) : print(i) # affiche les entiers de 0 à 9
Vi : valeur initiale du compteur	for i in range(5,10) : print(i) # affiche les entiers de 5 à 9
Vf : valeur finale du compteur	for n in range(10,20,2): print(n) # affiche les entiers de 10 a 18 avec un pas =2 (10,12 ,14,16,18)
Le compteur doit être de type scalaire	for i in range (10,0,-1) : print (i) #affiche les entiers de 10 à 1 avec un pas = -1

Savoir- faire : Le type scalaire

Prédécesseur	Valeur	Successeur	Type
	10		
	5		
	"C"		
	"d"		
	5,2		
	7,05		
	"porte"		

Le type scalaire a un et un seul prédécesseur et un et un seul successeur. **Exemples :**

Exercice 10 : Saisir une chaîne numérique puis calculer la somme de ses chiffres.

Exemple : ch="123456" → S=21

Exercice 11 : Saisir une chaîne ch de taille>1 puis l'inverser et l'afficher.

Exemple : ch="123456" → Inv= ch="654321"

Exercice 12 : Saisir une chaîne ch de taille paire (supposée alphabétique) puis afficher le nombre de voyelles et la chaîne de consonnes. **Exemple :** ch="anniversaire" → Nbvoy=6, chcons=ch="nnvrsr"

Exercice 13 : de Saisir deux entiers x et y (x>0 et y≥0) puis calculer et afficher x^y.

Exemple : x=3, y=2 → Le programme affiche : Puissance(3,2)=9

Savoir- faire : Les fonctions : Syntaxe algorithmique

Fonction Nom_fonction (pf1 : type1, pf2 : type2, ... ,pfn : typen) : type_résultat

DEBUT

 Traitement

Retourner Résultat

FIN

Les fonctions : Appel algorithmique

Objet ← Nom_fonction (pe1, pe2, ..., pen)

NB : pf est un paramètre formel défini au niveau de la définition de la fonction. pe est un paramètre effectif utilisé lors de l'appel de la fonction.

Savoir- faire : Les procédures : Syntaxe algorithmique

Procédure Nom_procedure (pf1 : type1, pf2 : type2, ... ,pfn : typen)

DEBUT

 Traitement

FIN

Les procédures : Appel algorithmique

Nom_procedure (pe1, pe2, ..., pen)

NB : pf est un paramètre formel défini au niveau de la définition de la fonction. pe est un paramètre effectif utilisé lors de l'appel de la fonction.

Python

```
def saisir():
    n=int(input('n='))
    p=int(input('p='))
    while(not(0<p<=n)):
        n=int(input('n='))
        p=int(input('p='))
    return p,n

def fact(x):
    f=1
    for i in range(1,x+1):
        f=f*i
    return f

#Programme principal
p,n=saisir()
f1=fact(n)
f2=fact(p)
f3=fact(n-p)
c=f1/(f2*f3)
print('c=',c)
```

Exercice 14 : Saisir un nombre composé de 2 chiffres, calculer et afficher la somme des factorielles de ces 2 chiffres.

Exemple : n=15 → S=1 !+5 !=1+120=121

Exercice 15 : Saisir deux nombres n et m (n>0 et m>0) puis afficher s'ils sont amis ou non. Deux nombres sont dits amis si n= somme des diviseurs de m sauf lui-même et m=somme des diviseurs de n sauf lui-même.

Exemple :

- Diviseurs (220) : 1+2+4+5+10+11+20+22+44+55+110=284
- Diviseurs (284) : 1+2+4+71+142=220

Savoir-faire : Structure conditionnelle réduite simple : Syntaxe

Algorithme	Python
Si condition Alors Traitement 1 Fin si	if conditions(s) : Traitement 1

Exercice 16 : Le jeu consiste à tirer au hasard un entier X formé de deux chiffres.

- Si le chiffre de dizaine est égal au chiffre d'unité, on gagne 10 points.
- Si le nombre est divisible par 3, on gagne 2 points.
- Si le chiffre de dizaine est impair, on gagne 1 point.
- Si les 3 conditions sont vérifiées alors afficher X est un nombre magique

- Afficher le score final.

Exemple : si le nombre tiré est 99, on gagne 13 pts → 99 est un nombre magique

Ecrire un algorithme qui permet de calculer, afficher le score obtenu et afficher si le nombre est magique.

Exercice 17 : Saisir un nombre $n > 10$ puis affiche s'il est sublime ou non. Un nombre est dit sublime si le nombre de ses diviseurs et la somme de ses diviseurs sont parfaits.

Un nombre est dit parfait s'il égal à la somme de ses diviseurs sauf lui-même.

Exemple : 12 est un nombre sublime

Diviseurs : 1, 2, 3, 4, 6, 12

Nombre de diviseurs = 6 → parfait

Somme des diviseurs = 28 → parfait

Exercice 18 : Saisir $n1$ et $n2$ ($n1 > 0, n2 > 0$ et $n1 \neq n2$) puis afficher s'ils sont jumeaux.

Deux nombres $n1$ et $n2$ sont dits jumeaux si $n1$ et $n2$ sont premiers et $n2 = n1 + 2$

Exemple : 5 et 7 sont jumeaux

Exercice 19 : Saisir une chaîne ch de taille minimale 4, saisir aléatoirement un entier p ($0 < p < \text{long}(ch) - 1$), inverser la chaîne (caractère d'indice 0, caractère d'indice p), inverser la chaîne (caractère d'indice $p+1$, caractère d'indice $\text{long}(ch) - 1$) puis concaténer les deux chaînes inversées et afficher la chaîne résultat.

Exemple : $ch = \text{"abc12+59?"}$, $p = 3$, chaîne résultat : "1cba?95+2" .

La fonction Alea

Définition

Alea est une fonction qui permet de saisir une valeur comprise entre une valeur initiale V_i et une valeur finale V_f .

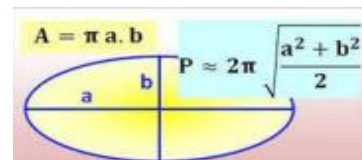
Exercice 11 : Lancer Thonny et remplir le tableau suivant :

Description	Algorithme	Python
.....	$X \leftarrow \text{aléa}()$ <code>X = random()</code> <code>print (X)</code>
Générer un nombre entier dans [val1, val2]	$X \leftarrow \text{Aléa}(\text{val1}, \text{val2})$ <code>X = (14, 30)</code> <code>print(X)</code>

Savoir-faire : Fonctions prédéfinies des types numériques

Algorithmique	Rôle
Arrondi (x)	Retourne l'entier le plus proche de la valeur de x.
RacineCarré (x)	Retourne la racine carrée d'un nombre x positif.
Aléa (vi, vf)	Retourne un entier aléatoire de l'intervalle [vi, vf].
Ent (x)	Retourne la partie entière de x.
Abs (x)	Retourne la valeur absolue de x

Exercice 20 : Saisir un réel b et un réel a une valeur aléatoire comprise entre [b+2, b+20] puis calculer et afficher la surface et le périmètre d'une éclipse.



Exercice 21 : saisir une chaîne de caractère ch représentant un code binaire de 8 bits. Le programme déterminera le code ASCII et le caractère qui correspondent à ce code.

Exemple : ch = "01011001" → le programme affichera : "89 : Y". Pour cet exemple, on détermine le code ASCII comme suit : Code binaire 0 1 0 1 1 0 0 1 Code ASCII $0*2^7 + 1*2^6 + 0*2^5 + 1*2^4 + 1*2^3 + 0*2^2 + 0*2^1 + 1*2^0$

Savoir-faire : Le type caractère

Les lettres alphabétiques majuscules ou minuscules ("A".."Z", "a".."z"), les chiffres ("0".."9"), les signes de ponctuation (".", "?", "...") et les symboles ("#", "&", " "...) // " " → le caractère espace

Savoir-faire : Les fonctions sur le type caractère

Algorithme & Python	Rôle	Exemples
A ← ord(c)	A contient le code ASCII du caractère c	print(ord("A"))=..... → ord("B")=..... print(ord("a"))=..... → ord("b")=.....
C ← chr(d)	C contient le caractère dont le code ASCII est d	print(chr(65))=..... → chr(66)=..... print(chr(97))=..... → chr(98)=.....

Exercice 22 : Saisir une chaîne de caractères ch alphabétique et afficher l'occurrence de chaque lettre de l'alphabet dans cette chaîne.

Exemple : ch="Anniversaire" → le programme affiche :

2 Lettre A	2 Lettre I	0 Lettre R
0 Lettre B	0 Lettre J	1 Lettre S
...
2 Lettre E	2 Lettre N	0 lettre X
0 Lettre F	0 Lettre O	0 Lettre Y
...	0 Lettre Z

Exercice 23 : Soit la suite U définie par :

$$U_0=1$$

$$U_1=3$$

$$U_n=2*U_{n-1}+3*U_{n-2} \text{ pour tout } n \geq 3$$

Saisir n ($3 \leq n \leq 20$) puis afficher les n premiers termes de la suite U.

Exemple : n=5 → Le programme affiche : 1, 3, 9, 21, 69

Situation 2 : Jeu Chilaformi

Règles du jeu

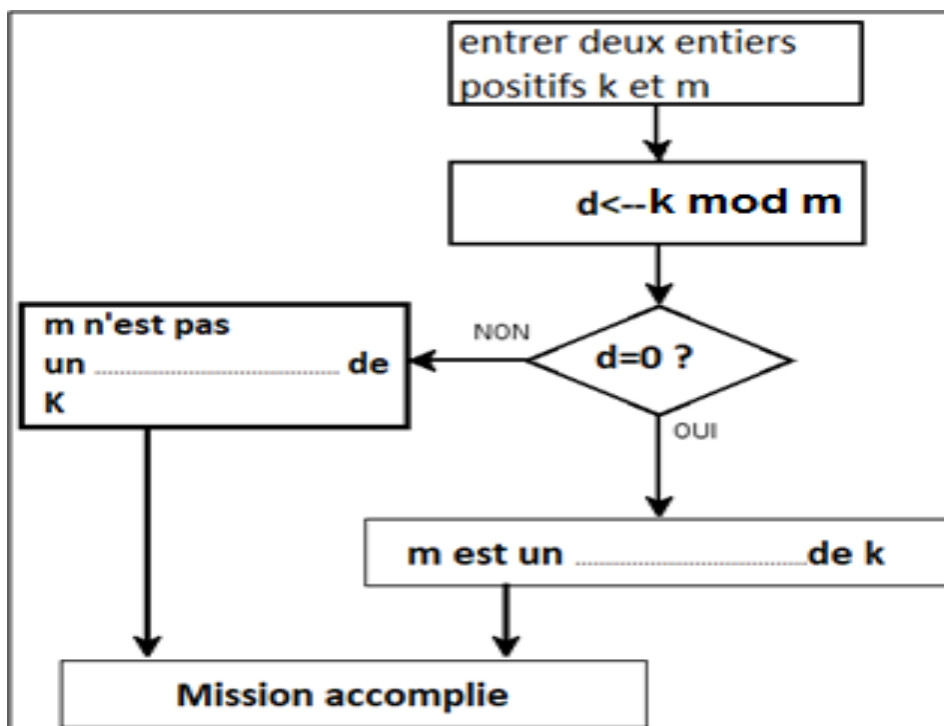
- L'ordinateur saisit aléatoirement un nombre V1 compris entre 0 et 5.
- L'utilisateur saisit un nombre V2 compris entre 0 et 5.
- Calculer la somme de V1 et V2.
- Si la somme est paire, on ajoute 1 au score S1 de l'ordinateur.
- Si la somme est impaire, on ajoute 2 au score S2 de l'utilisateur.
- Le jeu s'arrête quand l'un des scores (S1 ou S2) atteint 5.
- Afficher le joueur gagnant.

Savoir-faire : Structure conditionnelle simple : Syntaxe

Algorithme	Python
Si condition Alors Traitement 1 Sinon Traitement 2 Fin si	if conditions(s) : Traitement 1 else : Traitement 2

Exercice 24 : Un algorithme peut aussi être représenté sous forme graphique. On parle d'organigramme (ou d'ordinogramme).

Transformer cet organigramme en un algorithme.



Exercice 25 : Saisir un entier X puis afficher s'il est positif ou négatif. **Exemple : X=5 → 5 positif**

Exercice 26 : Saisir un entier X puis afficher s'il est positif ou négatif ou neutre.

Exemple : X=0 → 0 neutre

Savoir- faire : Structure conditionnelle généralisée : Syntaxe

Algorithme	Python
Si condition Alors Traitement1 Sinon si condition 2 Alors Traitement 2 Sinon si Alors Sinon Traitement n Fin si	if conditions 1 : Traitement 1 elif condition 2 : Traitement 2 else : Traitement n

Exercice 27 : saisir 3 réels a, b et c puis résoudre dans R l'équation $ax^2+bx+c=0$.

- ✓ Si $a \neq 0$, il s'agit de calculer $\Delta=b^2-4ac$
 - Si $\Delta=0$ puis $x1 = \frac{-b-\sqrt{\Delta}}{2a}$ et $x2 = \frac{-b+\sqrt{\Delta}}{2a}$
 - Si $\Delta=0$ alors le programme affiche une solution unique : $= \frac{-b}{2a}$
 - Si $\Delta<0$ alors le programme affiche « Pas de solutions réelles ».
- ✓ Sinon ($a=0$), l'équation devient de premier degré :
 - Si $b \neq 0$ alors le programme affiche la solution : $= \frac{-c}{b}$
 - Sinon ($b = 0$) :
 - Si $c \neq 0$ alors le programme affiche « Ensemble vide ».
 - Sinon ($c = 0$) alors le programme affiche « Ensemble R ».

Exercice 28 : Ecrire un algorithme qui permet de saisir deux dates valides d1 et d2 sous forme jj/mm/aaaa puis afficher la date la plus récente.

Exemple : d1=31/01/2017, d2=02/02/2018 → 02/02/2018 est plus récente que 31/01/2017

Exercice 29 : Déterminer et afficher le nombre total d'appuis sur les touches du clavier d'un téléphone portable pour saisir un mot donné de N lettres, supposées non accentuées ($4 \leq N \leq 9$).

Sur les touches (2, 3, 4, 5, 6, 7, 8 et 9) du clavier d'un téléphone portable, sont inscrites des lettres pour écrire des messages en plus des chiffres.

Par exemple, sur la touche 5 sont inscrites les lettres J, K et L.

- Pour taper la lettre J on appuie une seule fois.
- Pour taper la lettre K on appuie deux fois.
- Pour taper la lettre L on appuie trois fois.

Indication :

La figure suivante donne la répartition des lettres sur les touches du clavier d'un téléphone portable.

1	2 ABC	3 DEF
4 GHI	5 JKL	6 MNO
7 PQRS	8 TUV	9 WXYZ
* +	0 -	#

Savoir- faire : Structure conditionnelle à choix : Syntaxe

Au niveau de l’algorithme	Au niveau du Python
Selon Sélecteur Faire V1_1, V1_2,...,V1_n : Traitement 1 V2_1..V2_n : Traitement 2 [Sinon Traitement n] Fin Selon	<pre> a=int(input("donner un entier : ")) match a : case 1: print ('un') case 2: print ('deux ') case 3: print ('trois ') case 4 : print ('quatre') case _: print ('bonjour') </pre>

Exercice 30 : Saisir un caractère c puis afficher sa nature : consonne majuscule ou voyelle majuscule ou consonne minuscule ou voyelle majuscule ou point de ponctuation ou chiffre ou symbole.

Exemple : c="!" → Le programme affiche : **signe de ponctuation**

Exercice 31 : Saisir l’heure (comprise entre 0 et 23) et les minutes (comprises entre 0 et 59) puis afficher l’heure qu’il sera une minute plus tard.

Exemples

- Si l'utilisateur tape 21 puis 32, le programme affiche : **"Dans une minute, il sera 21 h 33 mns"**.
- Si l'utilisateur tape 21 puis 59, le programme affiche : **"Dans une minute, il sera 22 h 00 mns"**.
- Si l'utilisateur tape 23 puis 59, le programme affiche : **"Dans une minute, il sera 00 h 00 mns"**.

Exercice 32 : Saisir un mois m compris entre 1 et 12 et une année a composée de 4 chiffres puis afficher le nombre de jours du mois.

NB : Une année est dite bissextile si elle est divisible par 4.

Exemples

- m=5 → Nbjours=31
- m=4 → Nbjours=30
- m=2, a=2000 (année bissextile) → Nbjours=29
- m=2, a=1999 (année non bissextile) → Nbjours=28

Exercice 33 : Saisir une chaîne de taille>0 puis vérifier si elle est palindrome. Une chaîne est dite palindrome si elle est la même de droite à gauche et de gauche à droite. **Exemples :** "AZZA", "radar"

Exercice 34 : Tout nombre positif de deux chiffres ab, tel que a≠b, possède une liste appelée "liste vers 9". Le principe est le suivant : on calcule la différence entre ab et son

symétrique ba, le résultat trouvé subit le même traitement, on répète ce processus jusqu'à obtenir une différence = 9. L'ensemble constitué par le nombre initial et les résultats des différences est appelé "liste vers 9". *TAF* : Saisir un nombre X positif composé de deux chiffres obligatoirement différents, générer sa "liste vers 9" et l'afficher.

Exemple : $X = 18$, $81-18 = 63$, $63-36 = 27$, $72-27 = 45$, $54-45 = 9$ La liste vers 9 est de 18 est : 63 27 45 9

Exercice 35 : (Bac pratique 2005)

Saisir deux entiers naturels et afficher s'ils sont premiers entre eux ou non.

a et b sont dits premiers entre eux, si $PGCD(a,b) = 1$.

Exemple :

Pour $a = 7$ et $b = 4$, $PGCD(7,4) = 1$ donc 7 et 4 sont premiers entre eux.

Indication :

Sachant que $PGCD(a, b) = PGCD(b, r)$, avec $r = a \bmod b$.

Tant que le reste r est non nul, on remplace a par b et b par r. Le dernier reste r non nul est alors le PGCD des deux nombres

Exemple

a	b	Reste (a mod b)
323	247	76
247	76	19
76	19	0

Donc $PGCD(323,247) = 19$

Sur les touches (2, 3, 4, 5, 6, 7, 8 et 9) du clavier d'un téléphone portable, sont inscrites des lettres pour écrire des messages en plus des chiffres.

Par exemple, sur la touche 5 sont inscrites les lettres J, K et L.

- Pour taper la lettre J on appuie **une seule fois**.
- Pour taper la lettre K on appuie **deux fois**.
- Pour taper la lettre L on appuie **trois fois**.

Exercice 36 PGCD (Méthode de différence) : Ecrire un algorithme qui permet de saisir 2 entiers a et b strictement positifs puis calculer et afficher leur PGCD en utilisant la méthode de différence.

Exemple : $a=12, b=8$

$PGCD(12, 8) = ?$

$12 > 8 \rightarrow PGCD(12, 8) = PGCD(12-8, 8) = PGCD(4, 8)$

$4 < 8 \rightarrow PGCD(4, 8) = PGCD(4, 8-4) = PGCD(4, 4)$

$4=4$ Donc $PGCD(12, 8)=4$

Exercice 37 : Saisir deux entiers $A > 0$ et $B > 0$, calculer et afficher PPCM(A,B).

Exemple : $a=8, c=12 \rightarrow PPCM(8,12)=24$

Exercice 38 : Saisir $n > 1$ puis le décomposer en facteurs premiers.

Exemple : $n=30 \rightarrow$ Le programme affiche : 2,3,5

Exercice 39 (Bac pratique 2007) :

Soit l'algorithme du programme principal suivant :

```

Début Liste9
Ecrire("Donner un entier de deux chiffres distincts ")
Lire (N)
Répéter
    N ← Absolue (N - Echanger(N))
    Ecrire(N)
Jusqu'à N = 9
Fin Liste9

```

N.B : la fonction *Echanger* permet de permuter les chiffres d'un nombre *N* donné formé de deux chiffres distincts.

L'algorithme ci-dessus est celui d'un programme permettant de déterminer et d'afficher la liste des nombres générés, à partir d'un entier *N* donné formé de deux chiffres distincts, selon le principe de l'exemple suivant :

Exemple : Si *N* = 19

1. Permuter les chiffres de *N*, on obtient 91.
2. Calculer la valeur absolue de la différence entre 91 et 19 pour générer le nombre 72 ; ($72 = 91 - 19$)
3. Répéter les deux étapes précédentes avec le nombre 72 pour générer le nombre 45 ; ($45 = 72 - 27$)
4. Refaire le même travail avec chacun des nombres générés jusqu'à obtenir un dernier nombre égal à 9.

- 1- Transformer la séquence 1) en une procédure Lecture et y ajouter les contrôles nécessaires pour que *N* soit composé de 2 chiffres distincts et en tenir compte dans le programme principal.
- 2- Ecrire la fonction *Echanger* permettant de permuter les 2 chiffres de *N*.
- 3- Ajouter dans le programme principal une instruction permettant d'afficher à chaque fois le nombre généré.

Situation 3 : Jeu de dominos (Bac théorique 2022 : Exercice 3)

Le jeu de dominos est un jeu chinois qui comporte 28 dominos. Un domino est formé de 2 parties. Chaque partie contient de 0 à 6 points.

Une suite de dominos est dite **valide** lorsque la 2^{ème} partie d'un domino est identique à la 1^{ère} partie du domino voisin.



Figure 1- Suite valide de dominos

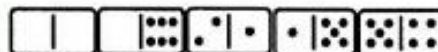
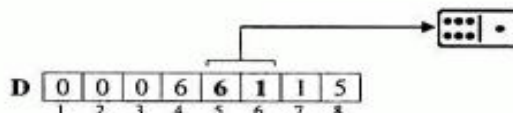


Figure 2- Suite invalide de dominos

La suite de dominos de la **Figure 2** est **invalide** car la 2^{ème} partie du 2^{ème} domino (6 points) est différente de la 1^{ère} partie du domino voisin (2 points).

Afin d'automatiser ce jeu, on se propose d'utiliser un tableau **D** d'entiers. Chaque case contient une valeur comprise entre 0 et 6. Ainsi, le tableau correspondant à la **Figure 1** est représenté comme suit :



Les paires (D[1], D[2]), (D[3], D[4]), (D[5], D[6]) et (D[7], D[8]) représentent une suite de 4 dominos

Questions

- 1) Présenter, sous forme d'un tableau **D**, la suite de dominos de la **Figure 2**.
- 2) Ecrire un algorithme nommé **Suite_Dominos** qui permet de :
 - Saisir la taille **n** du tableau **D**, avec **n** un entier pair et $4 \leq n \leq 56$.
 - Remplir le tableau **D** par **n** entiers compris entre 0 et 6.
 - Vérifier si la suite de dominos représentée par le tableau **D** est **valide** ou **invalide** puis afficher un message adéquat.

Savoir- faire : La structure tableau à une dimension

En algorithmme

Déclaration : Méthode 1		Déclaration : Méthode 2	
TDO		Tableau de Déclaration des Nouveaux Types (TDNT)	
Objet	Type / Nature	Type	
Nom_tableau	Tableau de n type_élément	Nom_type=Tableau de n type_élément	
		TDO	
Objet	Type / Nature		
Nom_tableau	Nom_type		

En python

On utilisera la bibliothèque **numpy** pour implémenter les tableaux.

- La déclaration d'un tableau se fait en deux étapes :
 - **Importation** des modules nécessaires de la bibliothèque **numpy**

Importation
<pre>from numpy import array ou from numpy import * ou import numpy as alias</pre>

- **Déclaration** du tableau

Tableau à une dimension

Déclaration
<pre>T = array ([Type_élément] * N) ou bien T = array ([valeur_initiale] * N)</pre>

Exemples de déclarations de tableaux en Python

Déclaration	Explication
T = array ([5] * 10)	Déclarer un tableau T de 10 entiers et initialiser ses éléments par « 5 ».
T = array ([float ()] * 10)	Déclarer un tableau T de 10 réels et initialiser ses éléments par «0.0 ».
T = array ([str] * 10)	Déclarer un tableau T de 10 chaînes de caractères.
T = array ([str()] * 10)	Déclarer un tableau T de 10 caractères et initialiser ses éléments par le caractère vide.
T = array ([''] * 10 , dtype = 'U20')	Déclarer un tableau T de 10 éléments initialisés par une chaîne vide. Chaque élément peut contenir 20 caractères au maximum.

T= array([int()]*50)	Déclarer un tableau T de 50 entiers
----------------------	-------------------------------------

Exercice 40 : Saisir un entier n pair, remplir un tableau T par n entiers positifs, calculer et afficher la somme, le maximum, le minimum des éléments de T.

Exemple : n=6

T	15	125	4	0	-2	-13
	0	1	2	3	4	5

S=129, max=125, min=-13

Exercice 41 : Saisir un entier n>0, remplir T1 par n chaînes numériques non vides et de taille maximale 7, remplir un tableau T2 telle que : T2[i] = Somme des chiffres de T1[i] puis afficher T2.

Exemple : n=4

T1	"157"	"11"	"4"	"1078"		T2	13	2	4	16
----	-------	------	-----	--------	--	----	----	---	---	----

Exercice 42 : Saisir n (2<n<10), de remplir aléatoirement T par n lettres minuscules, de diviser T en 2 tableaux (T1 : tableau de voyelles, T2 : tableau de consonnes) et d'afficher T1 et T2.

Exemple : n=5

T	"a"	"B"	"C"	"d"	"E"
---	-----	-----	-----	-----	-----

T1

"a"	"E"
-----	-----

 T2

"B"	"C"	"d"
-----	-----	-----

Exercice 43 : Saisir $n > 0$, remplir T par n lettres alphabétiques et afficher le nombre de voyelles et la chaîne des consonnes dans ce tableau T.

Exemple :

T

"a"	"B"	"C"	"d"	"E"
-----	-----	-----	-----	-----

Nbvoy=2, chcons="BCd"

Exercice 44 : Saisir $5 \leq n \leq 20$, remplir T par n entiers composés de 2 chiffres, saisir p (compris entre 0 et n-1), saisir X composé de 2 chiffres, insérer X dans le tableau T à la position p et afficher T.

Exemple : n=5, p=3, X=12

T

25	65	10	77	45
0	1	2	3	4

T après insertion de X à la position p

T

25	65	10	<u>12</u>	77	45
0	1	2	3	4	5

Exercice 45 : (Bac pratique 2006)

Ecrire un algorithme qui permet de :

- de remplir un tableau T de n chaînes de caractères ($2 < n < 20$). Chaque chaîne doit avoir un nombre de caractères supérieur ou égal à son indice dans le tableau.
- d'afficher pour chaque élément T[i] du tableau, les i premiers caractères de la chaîne.

Exemple : Soit T un tableau de 6 chaînes de caractères.

T

Bit	Modem	Ecran	Souris	Processeur	Mémoire
1	2	3	4	5	6

Le programme affichera :

B
Mo
Ecr
Sour
Proce
Mémoir

Exercice 46 : (Bac pratique 2006)

On se propose d'écrire un **algorithme** permettant de remplir deux tableaux T1 et T2 de N entiers à deux chiffres chacun ($2 \leq N \leq 15$) puis de former un tableau T tel que un élément T[i] est le résultat de la fusion des deux éléments T1[i] et T2[i] selon le principe suivant :

- Insérer le chiffre des dizaines du plus petit nombre parmi T1[i] et T2[i], entre les deux chiffres du plus grand nombre parmi T1[i] et T2[i].
- Mettre le chiffre des unités du plus petit nombre parmi T1[i] et T2[i], à droite du nombre obtenu.

Exemples :

- Pour T1[i] = 52 et T2[i] = 36 , T[i] sera égal à 5326
- Pour T1[i] = 13 et T2[i] = 47 , T[i] sera égal à 4173

Exercice 47 : Saisir n ($5 \leq n \leq 30$), remplir aléatoirement le tableau T par n entiers compris entre 0 et 100, saisir p compris entre 1 et n-1, vérifier si T[p] est égal à la somme d'un certain nombre d'éléments consécutifs de T qui le précèdent immédiatement. Dans ce cas, le programme affiche ces éléments, sinon, il affiche "**Condition non vérifiée**".

Exemple : n=7, p=5

T	3	<u>2</u>	<u>4</u>	<u>6</u>	<u>12</u>	18	20
	0	1	2	3	4	5	6

Le programme affiche : 6,4,2

Exercice 48 : Multiplication russe

Saisir deux entiers A et B ($2 \leq A \leq 100$ et $2 \leq B \leq 100$) puis calculer et afficher le produit de A et B en utilisant la méthode russe. Il s'agit d'utiliser seulement la division, la multiplication par 2 et l'addition selon le principe suivant :

- Le premier nombre est divisé par 2 (division entière) et le deuxième nombre est multiplié par 2.
- Ce processus est répété jusqu'à avoir 1 comme valeur du premier nombre.
- Les deux nombres de départ ainsi que les résultats de calcul seront mis dans 2 tableaux DIVIS et MULT : DIVIS contient les différentes valeurs du premier nombre et MULT contient les différentes valeurs du deuxième nombre.
- Le résultat du produit recherché est la somme des nombres du tableau MULT qui sont en face des nombres impairs du tableau DIVIS.

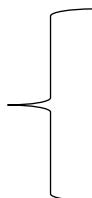
Exemple : A=11, B=13

DIVIS	11	5	2	1
	0	1	2	3

MULT	13	26	52	104
	0	1	2	3

$11 * 13 = 13 + 26 + 104 = 143$

Exercice 49 : Soit la suite U définie par :



$$U_0=1$$

$$U_1=3$$

$$U_n=2*U_{n-1}+3*U_{n-2} \text{ pour tout } n \geq 3$$

Saisir n ($3 \leq n \leq 20$) puis afficher les n premiers termes de la suite U.

Exemple : n=5

Le programme affiche : 1, 3, 9, 21, 69

Exercice 50 : (Bac pratique 2015)

Pour sécuriser l'envoi des messages, deux chercheurs cryptent leurs messages en utilisant le principe suivant :

1. Saisir le message à crypter **msg**, sachant qu'il est composé uniquement par des lettres,
2. Remplir un tableau **T** par les ordres alphabétiques des lettres de **msg** de façon à ce que **T[i]** lui corresponde de **msg[i]** (Sachant que "A" et "a" sont d'ordre 1, "B" et "b" sont d'ordre 2, ...),
3. Remplacer chaque **T[i]** par $(T[i])^e \bmod (p*q)$ avec **p**, **q** et **e** trois constantes ayant pour valeurs respectivement **17**, **19** et **5**.

Le tableau **T** ainsi obtenu représente le code de la chaîne **msg**.

Exemple :

Pour la chaîne **msg**="Bonjour", **T** sera rempli initialement comme suit :

2	15	14	10	15	21	18
1	2	3	4	5	6	7

En effet "B" est d'ordre alphabétique 2, "o" est d'ordre alphabétique 15, ...

Après avoir codé en remplaçant chaque **T[i]** par $(T[i])^e \bmod (p*q)$ on obtient :

32	2	29	193	2	89	18
1	2	3	4	5	6	7

En effet :
 T[1] est remplacé par $(T[1])^e \bmod (p*q) = 2^5 \bmod (17*19) = 32$
 T[2] est remplacé par $(T[2])^e \bmod (p*q) = 15^5 \bmod (17*19) = 2$
 Etc.

Saisir une chaîne non vide formée uniquement par des lettres, la crypter selon le principe décrit ci-dessus et afficher le tableau de code obtenu.

Situation 4 : Algorithme de tri

Trier un tableau / une chaîne = Ordonner ses éléments par ordre croissant ou décroissant. Il existe plusieurs principes de tri.

Activité : Soit A un vecteur de N cases ($3 \leq N \leq 20$), contenant des entiers de l'intervalle [65..90]. On se propose de trier le vecteur A dans l'ordre croissant et de l'afficher. **N.B.** Le remplissage du vecteur A se fait par la fonction prédéfinie Alea.

- Décomposer le problème en modules.
- Écrire l'algorithme de chacun des modules envisagés dans le programme principal.
- Déclarer les objets globaux du programme principal et les objets locaux relatifs à chaque module.

Tri par sélection

Principe

- 1) Rechercher la position de l'élément de plus petite valeur dans la partie non triée du tableau.
NB : 1^{ère} itération : la recherche se fait sur tout le tableau.
- 2) Permuter cet élément avec celui de la première position (s'ils sont différents).
- 3) Refaire 1- et 2- (N-1 fois) sur la partie non triée du tableau

Exemple : Soit N=5

T

1	5	2	3	4
0	1	2	3	4

T

0	1	2	3	4

T

0	1	2	3	4

T

0	1	2	3	4

T

0	1	2	3	4

Situation 5 : Recherche séquentielle

Activité : Saisir un entier $n > 1$, remplir un tableau T par n entiers strictement positifs, saisir X strictement positif puis vérifier et afficher si X existe dans T.

Exemple : Soit N=5

T

1	5	2	3	4
0	1	2	3	4

X=2 → 2 existe dans T

$X=7 \rightarrow 7$ n'existe pas dans T

Exercice 55 : Saisir n pair, remplir un tableau T par n cartes d'identités sachant que chaque carte d'identité est composée de 8 chiffres et uniques puis afficher T.