

Les sous-programmes

Prof : Mme Mkhinini Raja

1. Saisir un caractère Majuscule.

Procédure **saisie_majus** (@ c : caractère)

Début

Répéter

Ecrire ("Donner un caractère")

Lire(c)

Jusqu'à $c \in ["A".."Z"]$

Fin

2. Saisir une chaîne de caractère non vide et de longueur maximale égale à 20.

Procédure **saisie_ch** (@ ch : chaîne)

Début

Répéter

Ecrire ("Donner une chaîne")

Lire(ch)

Jusqu'à $\text{long}(ch) \in [1..20]$

Fin

3. Saisir un entier compris entre 10 et 100

Procédure **saisie_n** (@ n : entier)

Début

Répéter

Ecrire ("Donner un nombre")

Lire(n)

Jusqu'à $n \in [10..100]$

Fin

4. Remplir un tableau T par N entiers

Procédure **RemplirT** (@ t : tab ; n : entier)

Début

Pour i de 0 à n-1 faire

Lire(T[i])

Fin Pour

Fin

5. Remplir un tableau T par N entiers aléatoires de 2 chiffres

Procédure **Tab_alea** (@ t : tab ; n : entier)

Début

Pour i de 0 à n-1 faire

$T[i] \leftarrow \text{alea}(10,99)$

Fin Pour

Fin

6. Retourner une chaîne aléatoire de 10 caractères

Fonction **Chaine_aela** () : chaîne

Début

$ch \leftarrow ""$

Pour i de 1 à 10 faire

$ch \leftarrow ch + \text{chr}(\text{alea}(0,256))$

Fin Pour

Retourner(ch)

Fin

7. Remplir un tableau par des caractères aléatoires

Procédure **Tab_alea** (@ t : tab ; n : entier)

Début

Pour i de 0 à n-1 faire

$T[i] \leftarrow \text{chr}(\text{alea}(0,256))$

FinPour

Retourner(ch)

Fin

8. Retourner une chaîne aléatoire de 10 lettres majuscules

Fonction **Chaine-min** () : chaîne

Début

$ch \leftarrow ""$

Pour i de 1 à 10 faire

$ch \leftarrow ch + \text{chr}(\text{alea}(65,90))$

Fin Pour

Retourner(ch)

Fin

9. Retourner une chaîne aléatoire de 10 lettres minuscules

Fonction **Chaine-maj** () : chaîne

Début

$Ch \leftarrow ""$

Pour i de 1 à 10 faire

$ch \leftarrow ch + \text{chr}(\text{alea}(97,122))$

Fin Pour

Retourner(ch)

Fin

10. Convertir une chaîne majuscule en minuscule

Fonction **minus**(ch: chaîne): chaîne

Début

$Chm \leftarrow ""$

Pour i de 0 à $\text{long}(ch)-1$ faire

$chm \leftarrow chm + \text{chr}(\text{ord}(ch[i])+32)$

Fin Pour

Retourner(chm)

Fin

11. Vérifier si chaîne donnée est alphabétique

```

Fonction Verif (ch : chaîne) : booléen
Début
  Test ← vrai
  i ← 0
  Répéter
    Si (ch[i]) ∈ ["A".."Z", "a".."z"] Alors
      i ← i+1
    Sinon
      Test ← faux
  FinSi
  Jusqu'à (i=long(ch)) ou (test=faux)
  Retourner (test)
Fin

```

12. Remplir un tableau T par N entiers positifs croissant.

```

Procédure CroiTab(@ t : tab ; n : entier)
Début
  Ecrire ("Donner élément 1 du tableau")
  Lire(t[1])
  Pour i de 1 à n-1 faire
    Lire(t[i])
    jusqu'à (t[i]>t[i-1])
  FinPour
Fin

```

13. Afficher les éléments pairs d'un tableau T

```

Procédure Affiche (T : tab ; n : entier)
Début
  Pour i de 0 à n-1 faire
    Si T[i] mod 2 = 0 alors
      Ecrire (T[i])
  Fin Pour
Fin

```

14. Compter l'occurrence (nombre d'apparition) d'un entier dans un tableau.

```

Fonction Occ (t : tab ; n : entier, x : entier) : entier
  début
    nb ← 0
    Pour i de 0 à n-1 faire
      Si T[i]=x alors
        nb ← nb+1
      fin si
    Fin Pour
    Retourner (nb)
Fin

```

15. Compter l'occurrence (nombre d'apparition) d'un caractère dans une chaîne.

```

Fonction Occ (c : caractère, ch : chaîne) : entier
  Début
    nb ← 0
    Pour i de 0 à long(ch)-1 faire
      Si ch[i]=c alors
        nb ← nb+1
      Fin Pour
    Retourner (nb)
Fin

```

16. Déterminer le maximum d'un tableau.

```

Fonction maximum (t : tab ; n : entier) : entier
  Début
    Max ← t[0]
    Pour i de 1 à n-1 faire
      Si t[i]>max alors
        Max ← t[i]
      FinSi
    Fin Pour
    Retourner(max)
Fin

```

17. Déterminer le minimum dans T

```

Fonction minimum (t : tab ; n : entier) : entier
  Début
    min ← t[0]
    Pour i de 1 à n-1 faire
      Si t[i]<min alors
        min ← t[i]
      FinSi
    Fin Pour
    Retourner(min)
Fin

```

18. Inverser une chaîne de caractère.

```

Fonction Inverse (ch : chaîne) : chaîne
  Début
    Chx ← ""
    Pour i de long(ch)-1 à 0 (pas=-1) faire
      Chx ← Chx+ch[i]
    Fin Pour
    Retourner (Chx)
fin

```

19. Calculer la puissance de 2 entiers

```

Fonction puissance (a,b:entier):entier
  Début
    p ← 1
    Pour i de 1 à b faire
      p ← p*a
    Fin pour
    Retourner(p)
Fin

```

20. Calculer le factoriel d'un entier

```

Fonction factoriel (x: entier):entier
Début
  f ← 1
  Pour i de 1 à x faire
    f ← f * i
  Retourner(f)
Fin

```

21. Chercher un entier dans un tableau

```

Function Recherche (t:tab; n:entier,x:entier): booleen
Début
  i ← 0
  test ← faux
  répéter
    si t[i]=x alors
      test ← vrai
    sinon
      i ← i+1
  fin si
  jusqu'à (test = vrai) ou (i=n)
  retourner(test)
Fin

```

22. Générer une chaîne chn contenant que les chiffres à partir d'une chaîne ch

```

Fonction générer (ch :chaîne) :chaîne
Début
  Chn ← ""
  Pour i de 0 à long(ch)-1 faire
    Si ch[i] dans ["0".."9"] alors
      Chn ← Chn+ch[i]
    Fin si
  Fin pour
  Retourner (Chn)
Fin

```

23. Déterminer si un entier est premier ou pas

```

Fonction Premier (x :entier) : booleen
Début
  nb ← 0
  Pour i de 1 à x faire
    Si x mod i = 0 alors
      nb ← nb+1
    Fin Si
  Fin Pour
  Si nb=2 alors
    test ← vrai
  Sinon
    test ← Faux
  Fin si
  Retourner(test)
Fin

```

24. Afficher les entiers premiers qui ∈ [a..b]

```

Procédure afficher(a,b:entier)
Début
  Pour i de a à b faire
    Si premier(i)=vrai alors
      Ecrire(i,"/")
    Fin si
  Fin pour
Fin

```

25. Calculer le PGCD

```

Fonction PGCD(a,b:entier):entier
Début
  Tant que a ≠ b faire
    Si a > b alors
      a ← a-b
    sinon
      b ← b-a
    Fin Si
  Fin Tantque
  Retourner(a)
Fin

```

26 . Calculer le PPCM

```

Fonction PPCM(a,b:entier):entier
Début
  x ← a
  y ← b
  Tant que a ≠ b faire
    Si a < b alors
      a ← a+x
    Sinon
      b ← b+y
  Fin si
  Fin Tantque
Fin

```

27. Trier un tableau

```

Procédure tri_bulle @ T:tab ; n:entier)
Début
  Répéter
    Ok ← faux
    Pour i de 0 à n-2 faire
      si (t[i]>t[i+1]) alors
        Aux ← t[i]
        T[i] ← t[i+1]
        T[i+1] ← aux
      Ok ← vrai
    Fin si
  Fin Pour
  Jusqu'à (ok=faux)
fin

```

