

Fiche d'algorithmes et programmation

I - Les fonctions prédéfinies :

Chr(cod)	Retourne le caractère dont le code ASCII est cod
Ord(car)	Retourne le code ASCII du caractère car
Pos(ch1,ch2)	Retourne la 1ère position de la chaîne ch1 dans ch2
Convch(n)	Convertir une valeur numérique n en une chaîne
Valeur(ch)	Convertir une chaîne ch en une valeur numérique si c'est possible
Estnum(ch)	Retourne Vrai si ch est convertible en valeur numérique si non Faux
Sous_chaine(ch,d,f)	Retourne une partie de ch avec de la position d jusqu'à f (f exclue)
Effacer(ch,d,f)	Efface une partie de ch avec de la position d jusqu'à f (f exclue)

II - Des fonctions à apprendre :

Tri

```
procedure Tri(T:tab,n:entrier):  
debut  
  pour i de 0 à n-2 faire  
    pour j de i+1 à n-1 faire  
      si(T[i] > T[j]) alors  
        aux ← T[i]  
        T[i] ← T[j]  
        T[j] ← aux  
      fin si  
    fin pour  
  fin pour  
fin
```

Premier

```
fonction premier(n:entrier):booleen  
debut  
  test ← Vrai  
  pour i de 2 à n-1 faire  
    si(n mod i = 0) alors  
      test ← Faux  
    fin si  
  fin pour  
  retourner test  
fin
```

Factoriel

```
fonction fact(n:entier):entier  
debut  
  si(n = 0) alors  
    retourner 1  
  sinon  
    retourner n*fact(n-1)  
  fin si  
fin
```

Puissance

```
fonction puiss(a,n:entier):entier  
debut  
  si(b = 0) alors  
    retourner 1  
  sinon  
    retourner a*fact(a,n-1)  
  fin si  
fin
```

PGCD

```
fonction pgcd(a,b:entier):entier
debut
  si (b = 0) alors
    retourner a
  sinon
    retourner pgcd(b,a mod b)
  fin si
fin
```

PPCM

```
fonction ppcm(a,b:entier):entier
debut
  si (a mod b = 0) alors
    retourner a
  sinon
     $r \leftarrow a \bmod b$ 
  fin si
  retourner (a div r)*ppcm(b,r)
fin
```

Convert base 10 à base b

```
fonction conv_base10_b(x,b:entier):chaine
debut
  ch  $\leftarrow$  ""
  Tant que (x  $\neq$  0) faire
     $r \leftarrow x \bmod b$ 
    si (r < 10) alors
      c  $\leftarrow$  Convch(r)
    sinon
      c  $\leftarrow$  chr(r + 55)
    fin si
    ch  $\leftarrow$  c+ch
     $x \leftarrow x \div b$ 
  fin Tant que
  retourner ch
fin
```

Convert base b à base 10

```
fonction conv_b_b10(ch:chaine,b:entier):entier
debut
  s  $\leftarrow$  0
  pour i de 0 à long(ch)-1 faire
    si (ord("0") <= ord(ch[i]) <= ord("9")) alors
      k  $\leftarrow$  Valeur(ch[i])
    sinon
      k = ord(ch[i]) - 55
    fin si
    s  $\leftarrow$  s+k*puiss(b,long(ch)-i-1)
  fin pour
  retourner s
fin
```

Transfert de fichier à tableau

```
procedure Transfert_fich_tab(f:fich):
debut
  ouvrir("nom_fichier","f","rb")
  i  $\leftarrow$  0
  Tant que (non(Fin_fichier(f))) faire
    lire(f,e)
    T[i]  $\leftarrow$  e
    i  $\leftarrow$  i+1
  fin Tant que
  Fermer(f)
  i  $\leftarrow$  i - 1
fin
```

Transfert de tableau à fichier

```
procedure Trans_tab_fich(T:tab,i:entier,@f:fich):
debut
  ouvrir("nom_fichier.dat","f","wb")
  pour j de 0 à i-1 faire
    ecrire(f,T[j])
  fin pour
  Fermer(f)
fin
```

Suite

$$U_0 = 1$$

$$U_n = 3U_{(n-1)} + 2$$

```
fonction suite(n:entier):entier
debut
  si (n = 0) alors
    retourner 1
  sinon
    retourner (3*suite(n-1) + 2)
fin
```

Calcule d'un constante

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

```
fonction constante_approx(x,eps:réel):entier
debut
  s1 ← 1
  s2 ← s1 + x
  i ← 2
  Répéter
    s1 ← s2
    s2 ← s1 + puiss(x,i) div fact(i)
  jusqu'a(abs(s1-s2) < eps)
  retourner s1
fin
```

Calcule d'air Rectangle

```
fonction calc_air_rect(a,b:réel,n;entier):réel
debut
  som ← 0
  h ← (b-a)/n
  x ← a + (h/2)
  pour i de 0 à n-1 faire
    som ← som + f(x)
    x ← x + h
  fin pour
  retourner som * h
fin
```

Calcule d'air Trapèze

```
fonction calc_air_trpz(a,b:réel,n;entier):réel
debut
  h ← (a+b)/n
  som ← (f(a) + f(a+h))/2
  x ← a
  pour i de 0 à n-1 faire
    som ← som + (f(x) + f(x+h))/2
  fin pour
  retourner som * h
fin
```