

Lycée secondaire Rue de Russie	<i>Devoir de contrôle</i> <i>N° 3</i>	Année scolaire : 2006 / 2007
Professeur : Mohamed TRABELSI		Durée : 1h - Théorie
Matière : Algorithmique		Classe : 3 <sup>ème</sup> SI

Nom & prénom : .....

Note : /20

Énoncé :

On vous demande de créer un programme nommé **Recherche1** qui saisi un tableau C de n caractères alphabétiques voyelles minuscules avec  $1 < n < 200$ .  
Afficher le nombre d'occurrence du premier élément dans C.

Exemple :

C	a	e	i	a	y	a
	1	2	3	4	5	6

Affichage : Nombre d'occurrence de a est : 3.

TAF :

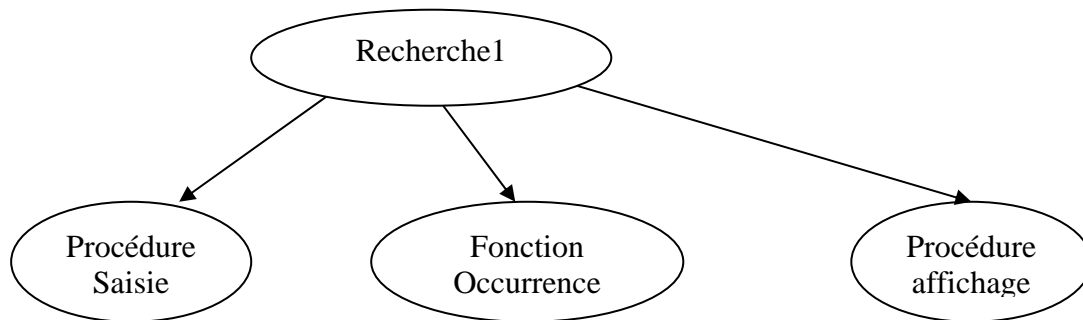
1. Tracer le diagramme de décomposition modulaire de ce problème.
2. Faire l'Analyse principale.
3. Créer un Type énuméré nommé : **alpha\_minus** composé de l'ensemble des caractères alphabétiques minuscules.
4. Créer un Type tableau nommé : **Tab\_char**, de type **alpha\_minus**.
5. On suppose que vous disposez de deux procédures déjà existantes :
  - Procédure **saisie** (var C : Tab\_char, var n : octet)
  - Procédure **affichage** (nbr\_occ : octet)
6. Faire l'analyse du module **occurrence ()** : précisez la nature et les paramètres de ce module. Utiliser le dans votre analyse.

Réponse :

Lycée secondaire Rue de Russie	<i>Correction</i> <i>Devoir de contrôle</i> <i>N° 3</i>	Année scolaire : 2006 / 2007
Professeur : Mohamed TRABELSI		Durée : 1h - Théorie
Matière : Algorithmique		Classe : 3 <sup>ème</sup> SI

### Solution

a. Décomposition modulaire



b. Analyse principale:

**Résultat :** Affichage (nbr\_occ)

**Traitements :**

nbr\_occ ← Occurrence (C, n)

Saisie (C, n)

### Algorithmme

- 0) Début **Recherche1**
- 1) saisie (C, n)
- 2) nbr\_occ ← Occurrence (C, n)
- 3) Affichage (nbr\_occ)
- 4) Fin **Recherche1**

Type
alpha_minus = (a, e, i, o, u, y)
Tab_char = tableau de 200 alpha_minus

objet	Type	Rôle
C	Tab_char	Tableau de caractères alpha_minus
n	Octet	Taille du tableau
nbr_occ	Octet	Nombre d'occurrence du premier élément de C

c. Analyse de la fonction occurrence :

Fonction **occurrence** (C : tab\_char, n : octet) : octet

**Résultat** : occurrence

**Traitement** :

Occurrence ← cpt,

avec cpt une variable qui compte le nombre d'occurrence de element.

element ← C[1] et cpt est initialisé par 1.

Une boucle Pour nous permettra de parcourir C et de faire le teste suivant :

Si element = C[i] alors cpt ← cpt + 1

Fin si

**Algorithme**

0) Fonction **occurrence** (C : tab\_char, n : octet) : octet

1) cpt ← 1

element ← C[1]

Pour i de 2 à n faire

Si element = C[i] alors cpt ← cpt + 1

Fin si

Fin pour

2) Occurrence ← cpt

3) Fin **occurrence**

T.D.O Locaux

<b>Objet</b>	<b>Type</b>	<b>Rôle</b>
cpt	Octet	Nombre d'occurrence
i	Octet	Compteur
element	alpha_minus	

#### d. Traduction Pascal :

```
Program Recherche1 ;
Uses wincrt;
Type
  Alpha_minus = (a, e, i, o, u, y);
  Tab_char = array [1..200] of alpha_minus;
var
  C : Tab_char;
  n : byte;
  nbr_occ : byte;

Procedure saisie(var C:tab_char; var n :byte);
Var i : byte;
Begin
  Write('Donner la taille du tableau :');
  Readln(n) ;

  For i :=1 to n do
  begin
    Write('Donner l"élément n°,i,') ;
    Readln(C[i]);
  end;
end;

function occurrence (C : tab_char; n : byte) : byte;
var
  cpt, i : byte;
  element : alpha_minus;
begin
  cpt := 1;
  element := C[1];
  for i:=2 to n do
    if element = C[i] then inc(cpt,1);
  occurrence := cpt;
end ;

procedure affichage (nbr_occ : byte);
begin
  write('le Nombre d"occurrence du premier élément de C est :', nbr_occ) ;
end;

begin
  saisie (C,n) ;
  nbr_occ := occurrence (C,n);
  affichage (nbr_occ) ;
end.
```