REPUBLIQUE TUNISIENNE MINISTERE DE L'EDUCATION ❖❖❖❖❖ **EXAMEN DU BACCALAUREAT** ❖❖❖❖❖ SESSION DE JUIN 2015	**EPREUVE PRATIQUE D'INFORMATIQUE** **SECTIONS** MATHEMATIQUES SCIENCES EXPERIMENTALES SCIENCES TECHNIQUES **DATE : 21/05/2015** **DUREE :** 1h **COEFFICIENT :** 0.5

Important :

1) Une solution modulaire au problème posé est exigée.

*2) Enregistrer au fur et à mesure votre programme dans le dossier **bac2015** se trouvant sur la racine du disque **C** en lui donnant comme nom votre **numéro d'inscription (6 chiffres).***

Pour sécuriser l'envoi des messages, deux chercheurs cryptent leurs messages en utilisant une clé de cryptage selon le principe suivant :

1. Saisir le message à crypter **msg**, sachant qu'il est composé par des lettres minuscules et des espaces,
2. Saisir une clé de cryptage **chcle** qui est une chaîne formée uniquement par des lettres minuscules et ayant la même longueur que le message à crypter,
3. Remplacer chaque lettre du message **msg** d'indice **i** par la lettre miniscule d'ordre alphabétique **k** sachant que:
   - k =ABS (ord(msg[i]) – ord(chcle[i])) + 1
   - L'espace ne sera pas crypté.

Exemple : soit le message suivant : "bonne reception" et soit la clé "homeofhappiness"

Message :	b	o	n	n	e		r	e	c	e	p	t	i	o	n
La clé de cryptage :	h	o	m	e	o	f	h	a	p	p	i	n	e	s	s
Message crypté :	g	a	b	j	k		k	e	n	l	h	g	e	e	f

En effet :

- La lettre "b" sera remplacé par la lettre d'ordre alphabétique k =ABS (ord("b") – ord("h")) + 1 qui est "g".
  En effet, k = ABS (66-72)+1 = 7 qui est l'ordre alphabétique de la lettre "g".
- La lettre "o" sera remplacé par la lettre d'ordre alphabétique k =ABS (ord("o") – ord("o")) + 1 qui est "a".
  En effet, k = ABS (79-79)+1 = 1 qui est l'ordre alphabétique de la lettre "a".
- etc.

Travail demandé :

Ecrire un programme Pascal qui permet de saisir un message **msg** et une clé de cryptage **chcle** en respectant les contraintes citées ci-dessus puis d'afficher le message crypté en utilisant le principe décrit précédemment.

Grille d'évaluation :

Questions	Nombre de points
Décomposition en modules	**2**
Appels des modules	**2**
Si exécution et tests réussis avec respect des contraintes **Sinon**	**16**
▪ Structures de données adéquates au problème posé	**3**
▪ Saisie de **msg** et de **chcle** avec respect des contraintes	**6 = (3+3)**
▪ Cryptage du message	**6**
▪ Affichage	**1**

```
Program    cryptage;
Uses    wincrt ;
Var     msg,chcle:string;
(************************************************)
Procedure    saisie_msg (var   msg : string);
Var   i:integer;
      verif:boolean;
begin
   repeat
         writeln('Message = ');
         readln(msg);
         i:=0;
         repeat
            i:=i+1;
            verif := msg[i]  in  ['a'..'z',' '] ;
         until  (verif = false) or (i=length(msg));
  until   verif;
end;
(************************************************)
Procedure   saisie_cle (msg:string ;  var  chcle:string);
var    i:integer;
       verif:boolean;
begin
   repeat
        writeln('Clé = ');
        readln(chcle);
        i:=0;
        repeat
            i:=i+1;
            verif := chcle[i]   in   ['a'..'z'];
        until     (verif = false) or (i=length(msg));
   until   verif     and   (length(chcle)=length(msg));
end;
(************************************************)
Function   crypter  (msg,chcle:string):string;
var     k,i:integer;
begin
  for   i:=1  to    length(msg)   do
     if  msg[i]<> ' '
              then  begin
                        k:= abs(ord(msg[i])-ord(chcle[i]))+1;
                        msg[i]:= chr(ord('a')+k-1);         (* ou   msg[i]:=chr(96+k); *)
                    end;
  crypter:=msg;
end;
(********************P.P**********************)
begin
    saisie_msg (msg);
    saisie_cle (msg,chcle);
    Writeln (crypter(msg,chcle));
end.
```

REPUBLIQUE TUNISIENNE MINISTERE DE L'EDUCATION ❖❖❖❖❖ **EXAMEN DU BACCALAUREAT** ❖❖❖❖❖ SESSION DE JUIN 2015 www.fennisalah.blogspot.com	**EPREUVE PRATIQUE D'INFORMATIQUE** **SECTIONS** MATHEMATIQUES SCIENCES EXPERIMENTALES SCIENCES TECHNIQUES **DATE : 21/05/2015** **DUREE :** 1h **COEFFICIENT :** 0.5

Important :

1) *Une solution modulaire au problème posé est exigée.*
2) *Enregistrer au fur et à mesure votre programme dans le dossier **bac2015** se trouvant sur la racine du disque **C** en lui donnant comme nom votre **numéro d'inscription (6 chiffres).***

On définit le **D**egré de **R**essemblance **DR** entre deux mots de même longueur par la formule suivante :

DR = (nombre de caractères en communs bien placés / longueur du mot) * 100

NB : Un caractère est dit bien placé lorsqu'il occupe la même position dans les deux mots.

Exemples :

- Pour mot1 = "**EXEMPLE**" et mot2 = "**EXAMENS**"

  Le degré de ressemblance DR = (3 / 7)*100 = 42.85

- Pour mot1 = "TRAITEMENTS" et mot2 = "INFORMATION"

  Le degré de ressemblance DR = (0/11)*100 = 00.00

Travail demandé :

Ecrire un programme Pascal qui permet de saisir une chaîne **Ch** non vide et composée de lettres majuscules, puis de remplir un tableau **T** par **N** ($5 \leq N \leq 10$) chaînes de caractères composées de lettres majuscules et de même longueur que **Ch** et d'afficher le degré de ressemblance entre **Ch** et les éléments de **T**.

Grille d'évaluation :

Questions	Nombre de points
Décomposition en modules	**2**
Appels des modules	**2**
Si exécution et tests réussis avec respect des contraintes	**16**
Sinon	
▪ Structures de données adéquates au problème posé	**3**
▪ Saisie de **N**, **Ch** et de **T** avec respect des contraintes	**5=(1+2+2)**
▪ Détermination des degrés de ressemblance	**6**
▪ Affichage	**2**

```
Program   Degre_ressemblance;
Uses    wincrt;
Type    tab = array[1..10] of string;
Var      t:tab;
         n:integer;
         ch:string;
(*********************************************************)
Procedure  saisies ( var ch:string ; var n:integer ; var t:tab);
Var     i:integer;
        Function   verif (st:string):boolean;
        Var    j:integer;
               test:boolean;
        begin
             j:=0;
             repeat
                  j:=j+1;
                  test:=st[j] in ['A'..'Z'];
             until (test=false) or (j=length(st));
             verif:=test;
        end;
begin
     repeat
          write('ch = ');  readln(ch);
     until   verif (ch);

     repeat
          write('N = ');  readln(n);
     until   n  in  [5..10];

     for   i:=1   to   n   do
          repeat
               readln(t[i]);
          until   verif (t[i])    and   (length(t[i])=length(ch));
end;
(*****************************************************)
Procedure  DR_mots (ch:string ; n:integer ; t:tab);
Var    i,j,nb:integer;
begin
    for   i:=1  to  n   do
          begin
               nb:=0;
               for   j:=1   to  length(ch)   do
                            if   ch[j]=t[i][j]   then   nb:=nb+1;
              writeln (t[i],' DR = ',(nb/length(ch))*100);
          end;
end;
(******************P.P*****************************)
begin
     Saisies (ch,n,t);
     DR_mots (ch,n,t);
end.
```

REPUBLIQUE TUNISIENNE
MINISTERE DE L'EDUCATION

❖❖❖❖❖

EXAMEN DU BACCALAUREAT

❖❖❖❖❖

SESSION DE JUIN 2015

EPREUVE PRATIQUE D'INFORMATIQUE

SECTIONS	MATHEMATIQUES SCIENCES EXPERIMENTALES SCIENCES TECHNIQUES
DATE : 21/05/2015	
DUREE : 1h	**COEFFICIENT : 0.5**

Important :

1) *Une solution modulaire au problème posé est exigée.*
2) *Enregistrer au fur et à mesure votre programme dans le dossier* ***bac2015*** *se trouvant sur la racine du disque* ***C*** *en lui donnant comme nom votre* ***numéro d'inscription (6 chiffres).***

Un hôtel souhaite attribuer des séjours gratuits à ses résidents à l'occasion de la fête de fin d'année en se basant sur leurs numéros de réservation qui sont des entiers de **4** chiffres.

Les résidents gagnants sont ceux qui possèdent plus de nombres premiers formés à partir de leurs numéros de réservation (le nombre lui-même, les nombres formés de trois chiffres adjacents, les nombres formés de deux chiffres adjacents et les nombres formés par un seul chiffre).

Exemple

Pour les numéros de réservation suivants :

3322	4774	3114	1012	3577	2291	1854	3149	4766	1579
1	2	3	4	5	6	7	8	9	10

Les numéros de réservation des résidents gagnants sont : 3577 et 1579 puisque :

- 3577 possède 5 nombres premiers qui sont 3, 5, 7, 7 et 577
- 1579 possède 5 nombres premiers qui sont 5, 7, 79, 157 et 1579

N.B. : Un nombre est dit premier s'il n'est divisible que par 1 et par lui-même. Par définition, 1 n'est pas premier.

Travail demandé

Ecrire un programme Pascal qui permet de remplir un tableau **T** par **N** ($10 \leq N \leq 100$) numéros de réservation, puis d'afficher la liste des résidents gagnants.

Grille d'évaluation :

Questions	Nombre de points
Décomposition en modules	2
Appels des modules	2
Si exécution et tests réussis avec respect des contraintes	16
Sinon ▪ Structures de données adéquates au problème posé ▪ Saisie de N et de T avec respect des contraintes ▪ Vérification de la propriété "premier" ▪ Détermination des résidents gagnants ▪ Affichage	3 4=(1+3) 2 5 2

```
Program residents_gagnants;
Type tab=array[1..100] of integer;
Var t:tab;
    n,k,pn:integer;
(**********************************)
Procedure saisie (var n:integer ; var t:tab);
Var i:integer;
begin
  repeat
    readln(n);
  until n in [10..100];
  for i:=1 to n do
    repeat
      readln(t[i]);
    until (1000<=t[i]) and (t[i]<=9999);
end;
(**********************************)
Function nombres (m:integer):integer;
Var i,j,x,err,nb : integer;
    ch,ch1:string;
  function premier (y:integer):boolean;
  var i,nb:integer;
  begin
    nb:=2;
    for i:=2 to (y div 2) do
      if (y mod i) =0 then nb:=nb+1;
    premier:=nb=2;
  end;
begin
  str(m,ch);
  nb:=0;
  for i:=4 downto 1 do
    for j:=1 to i do
      begin
        case i of
          4:ch1:=copy(ch,j,1);
          3:ch1:=copy(ch,1,j+1);
          2:ch1:=copy(ch,i,j+1);
          1:ch1:=copy(ch,3,j+1);
        end;
      val(ch1,x,err);
      if (x>1) and premier(x)
        then nb:=nb+1;
    end;
  nombres:=nb;
end;
(**********************************)
Function maxim (n:integer;t:tab):integer;
Var i,max,nb:integer;
begin
  max:=0;
  for i:=1 to n do
    begin
      nb:=nombres(t[i]);
      if nb>max then max:=nb;
    end;
  maxim:=max;
end;
(**************P.P******************)
begin
  saisie(n,t);
  pn :=maxim(n,t) ;
  for k:=1 to n do
    if nombres(t[k]) = pn
      then write (t[k],' ');
end.
```

RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION ◇◆◇◆ **EXAMEN DU BACCALAURÉAT** SESSION **2016**	**Épreuve pratique d'informatique**	
	Durée : 1h	**Coefficient : 0.5**
Sections : Maths, Sciences expérimentales et techniques	**Date : 19 mai 2016**	

Important :

1) *Une solution modulaire au problème est exigée.*
2) *Enregistrez au fur et à mesure votre programme dans le dossier **Bac2016** situé à la racine **C:** en lui donnant comme nom votre numéro d'inscription (**6 chiffres**).*

Un nombre est dit **oblong** s'il est le produit de deux entiers naturels consécutifs.

Exemples :

- 12 est un nombre oblong car 12 = 3 * 4
- 272 est un nombre oblong car 272 = 16 * 17
- 1640 est un nombre oblong car 1640 = 40 * 41

On se propose d'écrire un programme Pascal permettant de remplir un tableau T par N entiers positifs de quatre chiffres (avec $2 \leq N \leq 10$) et d'afficher tous les nombres ***oblongs*** du tableau T. Pour cela, on donne l'algorithme du programme principal suivant :

```
0) Début Oblong
1) Répéter
     Ecrire ("Donner la taille du tableau : ")
     Lire (N)
   Jusqu'à N Dans [2..10]
2) Pour i de 1 à N Faire
     Répéter
          Ecrire ("T[", i, "] = ")
          Lire (T[i])
     Jusqu'à (T[i] ≥ 1000) et (T[i] ≤ 9999)
   Fin Pour
3) Proc Afficher (T, N)
4) Fin Oblong
```

Travail demandé :

a. Traduire l'algorithme **Oblong** en un programme Pascal et ajouter les déclarations nécessaires.
b. Transformer la séquence n°1 en un module et apporter les modifications nécessaires dans le programme principal.
c. Transformer la séquence n°2 en un module et apporter les modifications nécessaires dans le programme principal.
d. Développer le module **Afficher** qui permet d'afficher les nombres ***oblongs*** contenus dans un tableau T de N entiers positifs de quatre chiffres.

Exemple : Pour **N = 6** et le tableau **T** suivant :

1056	3061	4512	1260	2724	5835
1	2	3	4	5	6

Le programme affichera :

Les nombres oblongs sont :
1056
1260

Grille d'évaluation :

Questions	Nombre de points
a. Traduction de l'algorithme **Oblong** en Pascal + Ajout des déclarations nécessaires.	6 + 1
b. Transformation de la séquence n°1 en un module + Modifications nécessaires dans le programme principal.	3 + 1
c. Transformation de la séquence n°2 en un module + Modifications nécessaires dans le programme principal.	4 + 1
d. Développement du modulé **Afficher**.	4

RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION ◇◇◇◇ **EXAMEN DU BACCALAURÉAT** SESSION 2016	Épreuve pratique d'informatique	
	Durée : 1h	Coefficient : 0.5
Sections : Maths, Sciences expérimentales et techniques	Date : 19 mai 2016 8h → 9h Sujet 2	

Important :

1) *Une solution modulaire au problème est exigée.*
2) *Enregistrez au fur et à mesure votre programme dans le dossier **Bac2016** situé à la racine **C:** en lui donnant comme nom votre numéro d'inscription (**6 chiffres**).*

Un nombre est dit **pronique** s'il est le produit de deux entiers naturels consécutifs.

Exemples :

- 12 est un nombre pronique car 12 = 3 * 4
- 272 est un nombre pronique car 272 = 16 * 17

On se propose d'écrire un programme Pascal permettant de remplir un tableau **T** par **N** entiers positifs de trois chiffres (avec $2 \leq N \leq 10$) et d'afficher tous les nombres ***proniques*** du tableau **T**. Pour cela, on donne l'algorithme du programme principal suivant :

```
0) Début Pronique
1) Répéter
     Ecrire ("Donner la taille du tableau : ")
     Lire (N)
   Jusqu'à N Dans [2..10]
2) Pour i de 1 à N Faire
     Répéter
          Ecrire ("T[", i, "] = ")
          Lire (T[i])
     Jusqu'à (T[i] ≥ 100) et (T[i] ≤ 999)
   Fin Pour
3) Proc Afficher (T, N)
4) Fin Pronique
```

Travail demandé :

a. Traduire l'algorithme **Pronique** en un programme Pascal et ajouter les déclarations nécessaires.
b. Transformer la séquence n°1 en un module et apporter les modifications nécessaires dans le programme principal.
c. Transformer la séquence n°2 en un module et apporter les modifications nécessaires dans le programme principal.
d. Développer le module **Afficher** qui permet d'afficher les nombres ***proniques*** contenus dans un tableau **T** de **N** entiers positifs de trois chiffres.

Exemple : Pour N = 6 et le tableau **T** suivant :

132	306	451	122	272	583
1	2	3	4	5	6

Le programme affichera :

Les nombres proniques sont :
132
306
272

Grille d'évaluation :

Questions	Nombre de points
a. Traduction de l'algorithme **Pronique** en Pascal + Ajout des déclarations nécessaires.	6 + 1
b. Transformation de la séquence n°1 en un module + Modifications nécessaires dans le programme principal.	3 + 1
c. Transformation de la séquence n°2 en un module + Modifications nécessaires dans le programme principal.	4 + 1
d. Développement du module **Afficher**.	4

RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION ◇◇◇◇ **EXAMEN DU BACCALAURÉAT** SESSION 2016	**Épreuve pratique d'informatique**	
	Durée : 1h	**Coefficient : 0.5**
Sections : Maths, Sciences expérimentales et techniques	**Date : 19 mai 2016** 9h30 → 10h30	

Important :

1) *Une solution modulaire au problème est exigée.*
2) *Enregistrez au fur et à mesure votre programme dans le dossier **Bac2016** situé à la racine **C:** en lui donnant comme nom votre numéro d'inscription (**6 chiffres**).*

Un mot est dit **mono-vocalisme en une voyelle donnée** s'il inclut une seule voyelle, qui est celle donnée, avec une ou plusieurs occurrences, sans distinction entre majuscule et minuscule.

Exemples :

- "**Cas**" est un mono-vocalisme en "**a**" car il inclut une seule occurrence (une seule fois) d'une seule voyelle ("**a**").
- "**Cesse**" est un mono-vocalisme en "**e**" car il inclut deux occurrences d'une seule voyelle ("**e**").
- "**Case**" n'est pas un mono-vocalisme en "**a**" car il inclut une autre voyelle différente de "**a**".

On se propose d'écrire un programme Pascal permettant d'afficher les mots mono-vocalismes en une voyelle donnée, dans un tableau **T** de **N** chaînes de caractères formées uniquement par des lettres (avec $1 \leq N \leq 20$). Pour cela, on donne l'algorithme du programme principal suivant :

```
0) Début MonoEn_V
1) Répéter
       Lire (N)
   Jusqu'à N Dans [1..20]
2) Pour i de 1 à N Faire
       Répéter
           Ecrire ("Donner une chaîne de caractères : ")
           Lire (T[i])
           j ← 0
           Répéter
               j ← j + 1
           Jusqu'à Non (Majus (T[i][j]) Dans ["A".."Z"]) ou (j = Long (T[i]))
       Jusqu'à (Majus (T[i][j]) Dans ["A".."Z"]) et (Long (T[i]) Dans [3..15])
   Fin Pour
3) Proc Saisir_V (V)
4) Proc Afficher (N, T, V)
5) Fin MonoEn_V
```

Travail demandé :

a. Traduire l'algorithme **MonoEn_V** en un programme Pascal et ajouter les déclarations nécessaires.
b. Transformer la séquence n°2 en un module et apporter les modifications nécessaires dans le programme principal.
c. Développer le module **Saisir_V** qui permet de saisir une voyelle **V**.
d. Développer le module **Afficher** qui permet de déterminer et d'afficher les mots mono-vocalismes en une voyelle donnée d'un tableau **T** de **N** chaînes de caractères.

Exemple : Pour **N = 6**, **V = "a"** et le contenu du tableau **T** suivant :

T	autre	trente	cours	HASARD	Son	bon
	1	2	3	4	5	6

Le programme affichera :

Les mots mono-vocalismes en "a" sont :
HASARD

Grille d'évaluation :

Questions	Nombre de points
a. Traduction de l'algorithme **MonoEn_V** en Pascal + Ajout des déclarations nécessaires.	6 + 1
b. Transformation de la séquence n°2 en un module + Modifications nécessaires dans le programme principal.	4 + 1
c. Développement du module **Saisir_V**.	2,5
d. Développement du module **Afficher**.	5,5

RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION ◇◇◇◇ **EXAMEN DU BACCALAURÉAT** SESSION 2016	**Épreuve pratique d'informatique**	
	Durée : 1h	**Coefficient : 0.5**
Sections : Maths, Sciences expérimentales et techniques	**Date : 19 mai 2016** 11R a'12h Sujet1	

Important :

1) *Une solution modulaire au problème est exigée.*
2) *Enregistrez au fur et à mesure votre programme dans le dossier* ***Bac2016*** *situé à la racine* ***C:*** *en lui donnant comme nom votre numéro d'inscription* ***(6 chiffres)****.*

Le terme "**CAPTCHA**" désigne un code permettant de différencier de manière automatisée un utilisateur humain d'un robot (*voir figure1*).

Pour générer automatiquement un code "CAPTCHA", on suit les étapes suivantes :

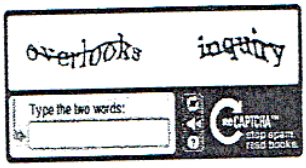

Figure1

1) On remplit d'une manière aléatoire, par des 0 ou des 1, un tableau **T** de 26 cases indicées de "A" à "Z".
2) On génère une chaîne de caractères **CH** par la concaténation des indices des cases du tableau **T** contenant la valeur 1.
3) On ajoute à la fin de la chaîne **CH**, le caractère dont le code ASCII est égal à : 70 + le nombre de voyelles contenues dans la chaîne **CH**.
   La chaîne obtenue représente le code CAPTCHA.

Exemple :

1) Pour le tableau **T** suivant :

0	1	0	0	1	1	1	0	1	0	0	1	0	0	1	0	0	0	0	1	0	1	0	1	0	1
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

2) La chaîne **CH** formée à partir des indices du tableau **T** est : "BEFGILOTVXZ".
3) On ajoute à la fin de la chaîne **CH**, le caractère "I" dont le code ASCII est égal à 73, car
   - le nombre de voyelles contenues dans la chaîne **CH** est égal à 3
   - 70 + 3 = 73 (qui est le code ASCII de la lettre "I").

D'où, le code "CAPTCHA" obtenu est la chaîne : "BEFGILOTVXZI"

On se propose d'écrire un programme Pascal permettant de générer automatiquement **N** codes "CAPTCHA" de la manière présentée ci-dessus (avec 1 < **N** < 11). Pour cela, on donne l'algorithme du programme principal suivant :

```
0) Début CAPTCHA
1) Répéter
       Ecrire ("Donner le nombre de codes à générer : ")
       Lire(N)
   Jusqu'à N Dans [2..10]
2) Pour i de 1 à N Faire
       Proc Remplir (T)
       Ecrire ("Le code CAPTCHA n° ", i, " est : ", FN GenererCap(T))
   Fin Pour
3) Fin CAPTCHA
```

Travail demandé :

a. Traduire l'algorithme **CAPTCHA** en un programme Pascal et ajouter les déclarations nécessaires.
b. Transformer la séquence n°1 en un module et apporter les modifications nécessaires dans le programme principal.
c. Développer le module **Remplir** qui permet de remplir d'une manière aléatoire, par des 0 ou des 1, un tableau **T** de 26 cases indicées de "A" à "Z".
d. Développer le module **GenererCap** qui permet de générer le code CAPTCHA comme indiqué précédemment.

Grille d'évaluation :

Questions	Nombre de points
a. Traduction de l'algorithme **CAPTCHA** en Pascal + Ajout des déclarations nécessaires.	5 + 1
b. Transformation de la séquence n°1 en un module + Modifications nécessaires dans le programme principal.	3 + 1
c. Développement du module **Remplir**.	4,5
d. Développement du module **GenererCap**.	5,5

RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION ◇◆◇◆ **EXAMEN DU BACCALAURÉAT** SESSION **2016**	**Épreuve pratique d'informatique**	
	Durée : 1h	**Coefficient : 0.5**
Sections : Maths, Sciences expérimentales et techniques	**Date : 19 mai 2016**	

Important :

1) *Une solution modulaire au problème est exigée.*
2) *Enregistrez au fur et à mesure votre programme dans le dossier **Bac2016** situé à la racine **C:** en lui donnant comme nom votre numéro d'inscription (**6 chiffres**).*

Un entier **N** est dit **unitairement parfait** s'il est égal à la somme de ses **diviseurs unitaires** sauf lui-même.

On appelle **diviseur unitaire** d'un entier **N**, tout entier **D** qui vérifie les conditions suivantes :

- **D** est un diviseur de **N**.
- **D** et (**N** Div **D**) sont premiers entre eux.

NB : Deux nombres sont dits **premiers entre eux** si leur plus grand commun diviseur (PGCD) est égal à 1.

Exemple 1 : Pour **N** = 36,

N n'est pas un entier unitairement parfait car il n'est pas égal à la somme de ses diviseurs unitaires :

Les diviseurs de 36	1	2	3	4	6	9	12	18
36 DIV diviseur	36	18	12	9	6	4	3	2
Test de primalité entre eux	**oui**	non	non	**oui**	non	**oui**	non	non
Les diviseurs unitaires de 36	1			4		9		
La somme des diviseurs unitaires de 36	**14** ($\neq 36$)							

Exemple 2 : Pour **N** = 60

N est un entier unitairement parfait car il est égal à la somme de ses diviseurs unitaires :

Les diviseurs de 60	1	2	3	4	5	6	10	12	15	20	30
60 DIV diviseur	60	30	20	15	12	10	6	5	4	3	2
Test de primalité entre eux	**oui**	non	**oui**	**oui**	**oui**	non	non	**oui**	**oui**	**oui**	non
Les diviseurs unitaires de 60	1		3	4	5			12	15	20	
La somme des diviseurs unitaires de 60	**60**										

On se propose d'écrire un programme Pascal permettant de déterminer et d'afficher tous les nombres ***unitairement parfaits*** de l'intervalle [**a**,**b**] (avec $2 \leq a < b \leq 100$). Pour cela, on donne l'algorithme du programme principal suivant :

0) Début UnitParf
1) Répéter
  Ecrire ("a = "), Lire (a)
  Ecrire ("b = "), Lire (b)
Jusqu'à (2 ≤ a) et (a < b) et (b ≤ 100)
2) Proc Afficher (a, b)
3) Fin UnitParf

Travail demandé :

a. Traduire l'algorithme **UnitParf** en un programme Pascal et ajouter les déclarations nécessaires.

b. Transformer la séquence n°1 en un module et apporter les modifications nécessaires dans le programme principal.

c. Développer le module **Afficher** qui permet d'afficher tous les nombres ***unitairement parfaits*** de l'intervalle **[a,b]**.

N.B : On pourra utiliser la fonction Test_Primalité ci-desssous, qui vérifie si deux entiers **p** et **k** sont premiers entre eux :

```
Function Test_Primalite (p, k : Byte) : Boolean;
Begin
    While (p <> k) Do
        If p > k Then p := p-k Else k := k-p;
    Test_Primalite := (p=1);
End;
```

Grille d'évaluation :

Questions	Nombre de points
a. Traduction de l'algorithme **UnitParf** en Pascal + Ajout des déclarations nécessaires.	4,5 + 1
b. Transformation de la séquence n°1 en un module + Modifications nécessaires dans le programme principal.	4 + 1
c. Développement du module **Afficher**.	9,5

RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION ◇◆◇◆ **EXAMEN DU BACCALAURÉAT** SESSION **2016**	**Épreuve pratique d'informatique**	
	Durée : 1h	**Coefficient : 0.5**
Sections : Maths, Sciences expérimentales et techniques	**Date : 19 mai 2016** du 13h30 à 14h30 Sujet 3	

Important *:*

1) *Une solution modulaire au problème est exigée.*
2) *Enregistrez au fur et à mesure votre programme dans le dossier* ***Bac2016*** *situé à la racine* ***C:*** *en lui donnant comme nom votre numéro d'inscription* ***(6 chiffres)****.*

Etant donné un entier **N** qui vérifie la propriété suivante :
"Le **produit** des **diviseurs** de **N** sauf lui-même est égal à une **puissance** de **N** avec un **exposant** strictement supérieur à 0".

Exemples :

- **N** = 6 vérifie cette propriété car le **produit** de ses diviseurs sauf lui-même est égal à **6** (1 * 2 * 3 = 6) qui est une **puissance** de **6**, avec un **exposant** égal à **1** (car $6 = 6^1$).
- **N** = 12 vérifie cette propriété car le **produit** de ses diviseurs sauf lui-même est égal à 144 (1 * 2 * 3 * 4 * 6 = 144) qui est une **puissance** de **12**, avec un **exposant** égal à **2** (car $144 = 12^2$).
- **N** = 30 vérifie cette propriété car le **produit** de ses diviseurs sauf lui-même est égal à 27000 (1 * 2 * 3 * 5 * 6 * 10 * 15 = 27000) qui est une **puissance** de **30**, avec un **exposant** égal à **3** (car $27000 = 30^3$).
- **N** = 9 ne vérifie pas cette propriété car le **produit** de ses diviseurs sauf lui-même est égal à **3** (1 * 3 = 3) qui n'est pas une **puissance** de **9**.
- **N** = 11 ne vérifie pas cette propriété car le **produit** de ses diviseurs sauf lui-même est égal à **1** qui est une **puissance** de **11**, avec un **exposant** égal à **0**.

On se propose d'écrire un programme Pascal permettant de déterminer et d'afficher tous les nombres de l'intervalle [**a**,**b**] (avec $2 \leq a < b \leq 100$) vérifiant la propriété donnée ci-dessus. Pour cela, on donne l'algorithme du programme principal suivant :

```
0) Début PuissDiv
1) Répéter
       Ecrire ("a = ")
       Lire (a)
       Ecrire ("b = ")
       Lire (b)
   Jusqu'à (2 ≤ a) et (a < b) et (b ≤ 100)
2) Proc Afficher (a, b)
3) Fin PuissDiv
```

Travail demandé :

a. Traduire l'algorithme **PuissDiv** en un programme Pascal et ajouter les déclarations nécessaires.
b. Transformer la séquence n°1 en un module et apporter les modifications nécessaires dans le programme principal.
c. Développer le module **Afficher** qui permet d'afficher tous les nombres de l'intervalle [**a**,**b**] et vérifiant la propriété donnée ci-dessus.

Grille d'évaluation :

Questions	Nombre de points
a. Traduction de l'algorithme **PuissDiv** en Pascal + Ajout des déclarations nécessaires.	4,5 + 1
b. Transformation de la séquence n°1 en un module + Modifications nécessaires dans le programme principal.	4 + 1
c. Développement du module **Afficher**.	9,5

```
Program  Oblong;
Uses   wincrt;
Type   tab=array[1..10] of integer;
Var    t:tab;
       n:integer;
(*****************************************)
Procedure  saisie(var n:integer);
begin
    repeat
          write('Donner la taille du tableau');
          readln(n);
    until n in [2..10];
end;
(*********************************************)
Procedure  remplir(n:integer ; var t:tab);
var  i:integer;
begin
    for i:=1 to n do
          repeat
                write('T[',i,'] = ');
                readln(T[i]);
          until  (1000<=T[i]) and (T[i]<=9999);
end;
(*********************************************)
Procedure  afficher (n:integer ; t:tab);
var  k,a:integer;
begin
   writeln('Les nombres oblongs sont :');
   for k:=1 to n do
         begin
            a:=trunc(sqrt(T[k]));
            if  t[k]=a*(a+1) then  writeln(t[k]);
         end;
end;
(****************P.P*********************)
begin
 saisie(n);
 remplir(n,t);
 afficher(n,t);
end.
```

```
Program  Pronique;
Uses   wincrt;
Type   tab=array[1..10] of integer;
Var    t:tab;
       n:integer;
(*****************************************)
Procedure  saisie(var n:integer);
begin
    repeat
         writeln('Donner la taille du tableau');
         readln(n);
    until n in [2..10];
end;
(*********************************************)
Procedure  remplir(n:integer ; var t:tab);
var  i:integer;
begin
    for i:=1 to n do
          repeat
               write('T[',i,'] = ');
               readln(T[i]);
          until  (100<=T[i]) and (T[i]<=999);
end;
(*********************************************)
Procedure  afficher (n:integer ; t:tab);
Var    i,k:integer;
begin
   writeln('Les nombres proniques sont :');
   for  k:=1  to  n  do
         begin
            i:=9;
            repeat
                 i:=i+1;
            until  i*(i+1) >= t[k];
            if  (t[k] = i*(i+1))  then  writeln(t[k]);
        end;
end;
(****************P.P************************)
begin
  saisie(n);
  remplir(n,t);
  afficher(n,t);
end.
```

```
Program  MonoEn_V;
Uses   wincrt;
Type   tab=array[1..20] of string;
Var    t:tab;
       n:integer;
       v:char;
(*************************************************)
Procedure  remplir(n:integer ; var t:tab);
var  i,j:integer;
begin
    for i:=1 to n do
      repeat
         write('T[',i,'] = ');
         readln(T[i]);
         j:=0;
         repeat
           j:=j+1;
         until not(upcase(t[i][j]) in ['A'..'Z'])or(j=length(t[i]));
      until (upcase(t[i][j]) in ['A'..'Z'])and(length(t[i]) in [3..15]);
end;
(*************************************************)
Procedure  saisie_v (var v:char);
begin
  repeat
    writeln('saisir une voyelle');
    readln(v);
  until upcase(v)in['A','E','O','I','U','Y'];
end;
(*****************************************************)
Procedure  afficher (n:integer ; t:tab ; v:char);
var    i,k,x,y:integer;
begin
   writeln('Les mots mono-vocalismes en ',v,' sont :');
   for k:=1 to n do
     begin
        x:=0;y:=0;
        for i:=1 to length(t[k]) do
              if upcase(v)=upcase(t[k,i])
                    then x:=x+1
                    else  if upcase(t[k,i])in['A','E','O','I','U','Y']
                                  then  y:=y+1;
        if (x<>0)and(y=0) then writeln(t[k]);
      end;
end;
(****************P.P************************)
begin
  repeat
      writeln('Donner la taille du tableau ');
      readln(n);
  until n in [1..20];
  remplir(n,t);
  saisie_v(v);
  afficher(n,t,v);
end.
```

```
Program CAPTCHA;
Uses    wincrt;
TYPE   Tab=Array['A'..'Z'] of 0..1;
VAR    T:tab;
       N,i:integer;
(*****************************************)
Procedure Saisie_N (VAR N:integer);
Begin
   Repeat
      Write('Donner le nombre de codes à générer : ');
      Readln(n);
   until n in [2..10];
End;
(*****************************************)
Procedure Remplir (VAR T:tab);
Var i:char;
Begin
   Randomize;
   For i:='A'  to 'Z'  do T[i]:=Random(2);
End;
(*******************************************************)
Function GenererCap (T:tab):String;
Var   i:char;
      ch:String;
      nbv:integer;
Begin
  ch:='';
  nbv:=0;
  for i:='A' to 'Z' do
       if  T[i]=1
           then Begin
                   Ch:=ch+i;
                   If  i IN ['A','E','I','O','U','Y']
                               then nbv:=nbv+1;
                End;
   ch:=ch + CHR(70+nbv);
   GenererCap:=ch;
End;
(***************P.P*********************************)
BEGIN
  Saisie_N(N);
  for i:=1 to N do
      Begin
         Remplir(T);
         Writeln('Le code CAPTCHA n° ', i, ' est : ', GenererCap(T));
      End;
END.
```

```
Program puissdiv;
Uses wincrt;
Var   a,b:integer;
(*****************************************)
Procedure saisie(var a,b:integer);
begin
    repeat
      write('a= '); readln(a);
      write('b= '); readln(b);
   until (2<=a)and(a<b)and(b<=100);
end;
(*********************************************)
Procedure afficher (a,b:integer);
Var n,i,prod:integer;
begin
   for n:=a to b do
       begin
          prod:=1;
          for i:=2 to n div 2  do if  (n mod i = 0) then  prod:=prod*i;
          if (prod mod n = 0) then writeln(n);
       end;
end;
(****************P.P************************)
begin
    saisie(a,b);
    afficher(a,b);
end.
```

```
Program  Unitparf;
Uses   wincrt;
Var    a,b:integer;
(*****************************************)
Procedure  saisie(var a,b:integer);
begin
    repeat
        write('a= '); readln(a);
        write('b= ');   readln(b);
   until  (2<=a) and (a<b) and (b<=100);
end;
(*********************************************)
Procedure  afficher (a,b:integer);
Var   n,i,som:integer;
  Function  primalite(p,k:integer):boolean;
  begin
      while p<>k do
          if  p>k  then  p:=p-k  else  k:=k-p;
      primalite:=p=1;
 end;
begin
   for  n:=a  to  b  do
      begin
         som:=0;
         for  i:=1  to  n div 2  do
             if (n mod i = 0) and primalite (i,n div i)
                    then som:=som+i;
         if  som = n
             then  writeln(n,' est unitaire parfait');
     end;
end;
(****************P.P************************)
begin
  saisie(a,b);
  afficher(a,b);
end.
```