Exercice 1 :

Pour chacun des cas suivants donner l'algorithme et le code Pascal d'un sous programme qui permet de :

1) Saisir un caractère Majuscule.

```
Def Proc saisie (var c : caractère)
Répéter
    Ecrire (''Donner un caractère alphabétique'')
    Lire(c)
    Jusqu'à majus(c) dans [''A''..''Z'']
Fin saisie
```

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

```
Def Proc saisie (var ch : chaine)
Répéter
    Ecrire (''Donner une chaine non vide de longueur max 20'')
    Lire(ch)
    Jusqu'à long(ch) dans [1..20]
Fin saisie
```

3) Vérifier est-ce qu'une chaîne de caractère donnée est alphabétique ou non.

```
Def FN Verif (ch : chaine) : booléen
Répéter
    Si majus(ch[i]) dans [''A''..''Z''] Alors
      Test ← vrai
    Sinon
      Test ← faux
    FinSi
    i← i+1
    Jusqu'à i>long(ch) ou test=faux
Verif ← test
Fin Verif
```

4) Remplir un tableau T par N entiers positifs croissant.

```
Def Proc CroiTab(var t : tab, n : entier)
Ecrire (''Donner élément 1 du tableau'')
Lire(t[1])
Pour i de 2 a n faire
     Répéter
      Ecrire (''Donner T['',i,''] supérieur à '', t[i-1])
      Lire(t[i])
     Jusqu'à (t[i]>t[i-1])
    FinPour
Fin CroiTab
```

5) Remplir un tableau T par N caractères Majuscules aléatoires

```
Def Proc Aleatoire(var t : tab, var n : entier)
Repeter
    Ecrire (''Donner 0<n≤100'')
    Lire (n)
    Jusqu'à n dans [1..100]
Pour i de 1 à n faire
    T[i]←alea(255)
    FinPour
Fin Aleatoire
```

6) Afficher un tableau T de N éléments.

```
Def Proc Affiche (t : tab, n :entier)
Pour i de 1 a n faire
    Ecrire (t[i],'' | '' )
```

```
    Fin Pour
  Fin Affiche
```

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

```
Def FN Occ(c : caractère, ch :chaine) :entier
n ← 0
Pour i de 1 à n faire
    Si ch[i]=c alors
      n← n+1
    Fin Pour
Occ← n
 Fin Occ
```

8) Vérifier la présence d'un caractère dans une chaîne.

```
Def FN Present (c : caractère, ch : chaine) : booléen
Test ← faux
i ← 1
Repeter
     Si (ch[i]=c) alors
       Test ← vrai
     FinSi
     i ← i+1
    jusqu'à i>long(ch) ou test=vrai
    present← test
 Fin Present
```

9) Déterminer le maximum d'un tableau.

```
Def FN max (t : tab) : entier
M←t[1]
Pour i de 2 à n faire
    Si t[i]>m alors
      M ← t[i]
    FinSi
    Fin Pour
Max ←m
Fin Max
```

10) Inverser une chaîne de caractère.

```
Def Proc Inverse (var ch : chaine)
Pour i de 1 à n div 2 faire
    X ← Ch[i]
    Ch[i] ← ch[n-i+1]
    Ch[n-i+1] ← x
    Fin Pour
Fin Inverse
```

Exercice 2 :

Soit la fonction Traitement suivante écrite en Pascal :

```
FUNCTION Traitement (d,f :integer;T:tab): Integer;
VAR   indmin, i : integer ;
BEGIN
Indmin: =d;
For i: = d+1 to f do
  IF T[i] < T[indmin] THEN
        Begin
            Indmin: = i;
        End;
Traitement: = indmin;
```

```
END;
```

1. Déterminer et compléter le type de cette fonction ainsi que la partie déclaration des variables locales.
2. Quelle est la valeur renvoyée par la fonction Traitement si **d=2, f=5** et le tableau contient les éléments suivants :

T	**-10**	**5**	**0**	**-6**	**10**	**13**
i	**1**	**2**	**3**	**4**	**5**	**6**

3. Quel est le rôle de cette fonction ?

➔ ***Détermine le minimum d'une suite d'éléments successifs dans un tableau T entre d et f***

Exercice 3 :

Soit l'Algorithme suivant :

```
0) DEF FN Traitement (T : Tab ; N : Entier) : Entier
1) NB ← 0
2) Pour i de 1 à N faire
        Si Non (majus (T [i]) dans ["A", "E", "O", "I", "U", "Y"]) alors
            NB ← NB + 1
        Fin si
   Fin Pour
3) Traitement ← NB
4) Fin Traitement
```

1. Préciser le rôle de la fonction Traitement suite à l'exécution suivante :

T	**"T"**	**"e"**	**"c"**	**"h"**	**"n"**	**"o"**	**"l"**	**"o"**	**"g"**	**"i"**	**"e"**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**

Détermine de nombre de consonne dans un tableau de caractères

2. Transformer la fonction Traitement en une procédure

```
Def Proc Traitement (T : Tab ; N : Entier ; var nb: Entier)
NB ←0
   Pour i de 1 à N Faire
        Si Non (majus (T [i]) dans ["A", "E", "O", "I", "U", "Y"]) alors
            NB ←NB + 1
        Fin si
   Fin Pour
Fin Traitement
```

Exercice 4 :

Soit la fonction booléenne **VERIF** suivante :

```
FUNCTION VERIF (Ch: string): Boolean;
Var
test:boolean;
Vc:integer;
Begin
Test: = False;
Vc: = 0;
Repeat
    Vc: = Vc+1;
    If Not (Upcase (Ch [Vc] )in ['A'..'Z']) Then
            Begin
                Test: = True;
            End;
Until (Test) or (Vc = Length (Ch));
Verif:=test ;
End;
```

1. Compléter les pointillés par les données manquantes.
2. Quel est le rôle de cette fonction ?

Vérifie est ce que une chaine contient des caractères qui ne sont pas des lettres

Exercice 5 :

Soit l'algorithme de la fonction suivante :

```
0) DEF FN Traitement (ch : chaine) : entier
1) P ← pos (' ', ch)
2) Nb← 0
3) Tant que (p<>0) faire
        Efface (ch, p, 1)
        Nb ← nb+1
        P ← pos (' ', ch)
   Fin tant que
4) Traitement ← nb+1
5) Fin Traitement
```

Cette fonction permet de calculer le nombre des mots dans une phrase ch.

On suppose que :

- La phrase **ch** ne commence pas par un espace
- La phrase **ch** ne se termine pas par un espace
- Entre deux mots de la phrase **ch** il y a un espace unique

a. Compléter les points par le code convenable.

b. Transformer la fonction Traitement en une procédure Traitement.

```
0) Def Proc Traitement (ch : chaine ; nb : entier)
1) P ← pos (' ', ch)
2) Nb ← 0
3) Tant que (p<>0) faire
        Efface (ch, p, 1)
        nb ← nb+1
        p ← pos (' ', ch)
        Fin Tant que
4) nb ← nb+1
5) Fin Traitement
```

Exercice 6 :

```
function inconnu (ch:string;c:char):boolean;
var
    i:integer;
    tr:boolean;
begin
    tr:=(1=0) ; i:=0;
    repeat
        i:=i+1;
        tr:=(ch[i]=c);
    until (i=length(ch)) or (tr);
inconnu:=TR;
end;
```

1) compléter les pointilles par les données marquantes.

2) déterminer le résultat retourné par la fonction pour chacun des cas suivants :

- **a.** inconnu ('algorithme', 'g') > ***inconnu = vrai***
- **b.** inconnu ('pascal' ,'H') > ***inconnu = faux***
- **c.** inconnu ('1H5','5') > ***inconnu = vrai***

3) donner le rôle de la fonction inconnu

la fonction inconnu permet de chercher un caractère c dans une chaine ch et renvoie s'il existe ou non

4) convertir la fonction inconnue en une procédure

```
Procedure inconnu (ch:string;c:char,var tr:boolean);
var
i:integer;
tr:boolean;
begin
```

```
        tr:=(1=0) ;
        i:=0;
        repeat
                i:=i+1;
                tr:=(ch[i]=c);
        until (i=length(ch)) or (tr);
end;
```

5) recopier et compléter le tableau suivant sachant que l'appel se fait en utilisant une variable X:

Appel de la fonction inconnu	Appel de la procédure inconnu
X :=inconnu(ch,c) ;	*Inconnu(ch,c,x) ;*

Exercice 7 :

Ecrire un programme permettant d'afficher tous les couples d'entiers (m,n) vérifiant la propriété suivante :
M ∈[1,100] et n∈[2,15]
Et m figure dans l'écriture du produit m.n
Exemples :

1- Si m = 20 et n= 6
Alors le produit m.n = 120 contient le nombre 20 dans ce cas le couple (20,6) sera affiché.
2- Si m = 20 et n = 12
Alors le produit m.n = 20 * 12 = 240 ne contient pas le nombre 20.

```
program Ex7  ;
  uses WinCrt;
  var
  m,n,i,j:byte;
  procedure saisie(var m,n:byte);
  begin
  repeat
  writeln('Donner m dans [1..100]');
  readln(m);
  until m in [1..100];
  repeat
  writeln('Donner n dans [2..15]');
  readln(n);
  until n in [2..15];
  end;
  procedure couple(m,n:byte);
  var
  chm, chp:string;
  p:byte;
  begin
  str(m,chm) ;
  str(n*m,chp);
  p:=pos(chm,chp);
  if p<>0 then
  writeln(' Le couple ',m,' , ',n,' vérifie les propriétés')
  else
  writeln('Le couple ne vérifie pas les propriétés');
  end;
begin
 saisie(m,n);
 couple(m,n);
end.
```

Exercice 8 :

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 lettre J, K et L.
Pour taper la lettre J on appuie une seule fois.

Pour taper la lettre K on tape deux fois.
Pour taper la lettre L on appuie trois fois.
Ecrire un programme pascal permettant de déterminer et d'afficher le nombre total d'appuies sur les touches du clavier d'un téléphone portable pour saisir un mot donné de N lettres, supposées non accentuées. (4<=N<=9).
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 -	#

```
program EX8;
uses  WinCrt;
var
nbr,n:byte;
ch:string;
procedure rempli(var n:byte;var
ch:string);
var
i:byte;
test:boolean;
begin
   repeat
      writeln('Donner une chaine sans
caracteres accentués') ;
      readln(ch);
      n:=length(ch);
      i:=0;
      test:=true;
      repeat
      i:=i+1;
       if not(upcase(ch[i]) in ['A'..'Z'])
then
          test:=false;
      until (test=false) or (i=n);
   until (n in [4..9]) and test;
   end;
procedure appui (ch:string;var
nbr:byte;n:byte);
var
i:byte;
begin
nbr:=0;
for i:=1 to n do
begin
 case upcase(ch[i]) of
'A','D','G','J','M','P','T','W':nbr:=nbr+1;
 'B','E','H','K','N','Q','U','X':nbr:=nbr+2;
 'C','F','I','L','O','R','V','Y':nbr:=nbr+3;
 'S','Z':nbr:=nbr+4;
 end;
 end;
 end;
begin
```

```
rempli(n,ch);
appui(ch,nbr,n);
writeln(nbr);
end.
```

Exercice 9 :

Soit l'algorithme suivant :

```
0) DEF FN Somme (p : entier) : entier
1) S← 0
   Pour i de 1 à (p div 2) faire
      Si (p mod i = 0) Alors
            S← S+i
      FinSi
   Fin pour
2) Somme ←S
3) Fin Somme
```

1. Exécuter cet algorithme pour p=6 puis donner son rôle.

Pour p=6 la somme est égale à 6 la fonction permet de calculer la somme des diviseurs

2. Transformer la fonction Somme en une procédure.

```
0) DEF Proc Somme (p :entier ; var s :entier)
1) S ← 0
2) Pour i de 1 à (p div 2) faire
   Si (p mod i=0) alors
      S ← s+i
   FinSi
   FinPour
3) Fin Somme
```

3. Ecrire un algorithme d'une procédure Saisie, permettant de saisir deux entiers positifs non nuls.

```
0) DEF Proc saisie (var x,y : entier)
1) Répéter
   Ecrire (''Donner deux entiers x et y positifs et non nuls'')
   Lire (x,y)
   Jusqu'à (x≥0) ET (y≥0)
2) Fin Saisie
```

4. En utilisant les deux modules **Somme** et **saisie**, écrire un algorithme d'un programme intitulé Amis, qui permet de saisir deux entiers positifs non nuls et d'afficher s'ils sont amis ou non. Sachant que deux entiers m et n sont dis amis, si la somme des diviseurs de n est égal à m et inversement.

```
0) Debut Amis
1) Proc Saisie(m,n)
2) Sn ← somme(n)
3) Sm ← somme(m)
4) Si (sm=n) ET (sn=m) alors
   Ecrire (m, '' et '',n,'' sont deux nombres amis'')
   Sinon
   Ecrire (m, '' et '',n,'' ne sont pas deux nombres amis'')
   FinSI
5) FIN Amis
```

Exercice 10 :

T étant un tableau de dimension **N_MAX** (N_MAX = 100), remplir par **N** caractères alphabétiques **(N ≤N_MAX).**
On vous demande d'écrire un programme nommé **ECLATEMENT** qui permet de:

- ✧ Lire un entier **N** strictement positif **(0 < N ≤ N_MAX).**
- ✧ Remplir le tableau **T.**
- ✧ Éclater **T** en deux tableaux **Tc** (contenant **J** élément consonnes de **T**) et **Tv** (contenant **K** élément voyelles de **T**).
- ✧ Afficher les deux tableaux **Tv** et **Tc.**

```
program Eclatement;
uses WinCrt;
const
nmax=100;
type
tab=array[1..nmax] of integer;
var
t,tn,tp:tab;
n,nn,np:byte;
procedure saisie (var n:byte);
begin
repeat
writeln('Donner n la taille du tableau');
readln(n);
until (n in [1..nmax]);
end;
procedure rempli(var t:tab;n:byte);
var
i:byte;
begin
 for i:=1 to n do
 begin
  writeln('Donner élément T[',i,']=');
  readln(t[i]);
 end;
end;
procedure eclat (t:tab;n:byte;var tn,tp:tab;var nn,np:byte);
var
i:byte;
begin
nn:=0;
np:=0;
for i:=1 to n do
  if t[i]<0 then
  begin
  nn:=nn+1;
  tn[nn]:=t[i];
  end
  else
  begin
  np:=np+1;
  tp[np]:=t[i];
  end;
end;
procedure affiche (t:tab; n :byte);
var
i:byte;
begin
for i:=1 to n do
write (t[i],' | ');
writeln;
end;
begin
 saisie(n);
 rempli(t,n);
 eclat(t,n,tn,tp,nn,np);
 affiche (t,n);
 affiche(tn,nn);
 affiche(tp,np) ;
end.
```

Exercice 11 :

Un nombre est dit parfait s'il est égal à la somme de ces diviseurs (''sauf lui même'').
Exemple : 6=3+2+1.
Ecrire une analyse et un algorithme du programme intitulé **parfait** qui permet de saisir un entier strictement positif puis il vérifie s'il est parfait.

```
program nb_parfaits;
uses wincrt;
var x:longint;
procedure saisie(var x:longint);
begin
repeat
writeln('Effectuer le test du nombre parfait sur ');
readln(x);
until (x>0);
end;
procedure parfait(x:longint);
var
s,d:longint;
begin
  s:=0;
    for d:=1 to (x div 2) do
      if x mod d=0 then
          s:=s+d;
    if s=x then
          writeln(x,' Nombre parfait')
    else
         writeln(x,' Nombre non parait');
end;
begin
saisie(x);
parfait(x);
end.
```

Exercice 12 :

La « **multiplication Russe** » est une méthode particulière permettant la multiplication de deux entiers A et B en utilisant seulement la multiplication par 2, la division par 2 et l'addition.
Exemple : pour A =17 et B = 19, le produit de A par B se fait comme suit :

A	**B**
17	**19**

Le premier nombre est divisé par 2 (division entière) et le deuxième est multiplié par 2 : on aura

8	**38**

Le processus se répète jusqu'à avoir dans la première colonne 1 :

17	**19**
8	**38**
4	**76**
2	**152**
1	**304**

Le résultat est la somme des nombres de la deuxième colonne qui sont en face des nombres impairs de la première colonne (donc les nombres de la deuxième colonne qui sont en face des nombres pairs de la première colonne seront ignorés).

→	17	**19**	
	8	38	Ignoré
	4	76	Ignoré
	2	152	Ignoré
→	1	**304**	
17 * 19 =		**19 + 304**	**= 323**

❖ On veut écrire un programme qui lit deux entiers positifs non nuls et inférieur à 100, calcule et affiche le produit de deux entiers par la méthode Russe.

```
Program EX12;
uses wincrt;
var
a,b:integer;
r:longint;

procedure saisie (var a,b : integer);
begin
repeat
writeln('Donner A entre 1 et 99');
readln(a);
writeln('Donner B entre 1 et 99');
readln(b);

until (a in [1..99]) and (b in [1..99]);
end;

procedure mult (a,b:integer; var r :longint);
var
x,y:longint;
begin
if a mod 2=1 then
r:=b
else
r:=0;
writeln(a,' ',b,' ',r);
repeat
  a:=a div 2;
  b:=b*2;
  if a mod 2=1 then
   r:=r+b;
   writeln(a,' ',b,' ',r);
until (a=1);

end;

begin
 saisie(a,b);
 mult(a,b,r);
 writeln(a, ' * ', b ,' = ',r);
end.
```

Exercice 13 :

Soit le programme intitulé **info** qui permet de :

- ✓ Saisir la taille **N** d'un tableau **T,** avec (**1<N<15**).
- ✓ Remplir un tableau **T** par **N** chaînes des caractères tel que la taille de chacune est dans [**3**..**20**].
- ✓ Chercher et afficher tous les chaînes Totalogramme contenue dans **T**.

« Une chaîne de caractères est dite Totalogramme si elle commence et se termine par la même lettre. » (Sans distinction entre majuscule et minuscule)

Exemple : Pour **N=6** :

T	Samir	système	temporairement	Bonjour	ses	elle
	1	**2**	**3**	**4**	**5**	**6**

Les mots totalogramme sont : temporairement, ses, elle

```
program Ex13;
uses wincrt;
```

```
type
 tab=array[1..14] of string[20];
var
n,i:byte;
t:tab;

procedure saisie (var n : byte);
begin
repeat
writeln('Donner N la taille du tableau entre 2 et 14');
readln(n);
until (n in [2..14]);
end;

procedure remplir (n:byte; var t:tab);
begin
for i:=1 to n do
repeat
writeln('Donner la chaine de la case ', i);
readln(t[i]);
until (length(t[i]) in [3..20]) ;
end;

procedure totalo (n:byte; t:tab);
begin
writeln('Les chaines totalogrammes sont:');
 for i:=1 to n do
 if t[i][1]=t[i][length(t[i])] then
  write(t[i],' | ');
end;

begin
 saisie(n);
 remplir(n,t);
 totalo(n,t);
end.
```

Exercice 14 :

Un texte est dit tautogramme si tous les mots commencent par la même lettre.

Exemple :

« Mazarin, ministre malade, méditait même moribond malicieusement mille maltôtes »

On suppose que la phrase est écrite correctement. (Pas d'espace en double)

Ecrire un programme qui permet de vérifier si une chaîne de caractère saisie au clavier est tautogramme ou non.

```
program Ex14;
uses WinCrt;
var
ch:string;
procedure saisie (var ch:string);
begin
repeat
writeln('Donner une chaine');
readln(ch);
until(pos('  ',ch)=0);
end;
function tauto (ch:string):boolean;
var
 c:char;
 p:byte;
begin
```

```
c:=ch[1];
repeat
  p:=pos(' ',ch);
  delete(ch,1,p);
until (c<>ch[1]) or (p=0);
if c<>ch[1] then
tauto:=false
else
tauto:=true;
end;
begin
 saisie(ch);
 if tauto(ch) then
 writeln( ch , ' est un tautogramme')
 else
 writeln(ch, ' n''est pas un tautogramme');
end.
```

Exercice 15 :

Ecrire un programme qui permet de dire si deux chaînes ch1 et ch2 (non vides) sont anagrammes ou non.
Deux chaînes sont dites anagrammes si elles sont formées par les mêmes caractères.
Exemple :
« Chien » et « Chine » sont anagrammes.

```
program Ex15;
   uses WinCrt;
   var
  ch1,ch2:string;
   procedure saisie(var ch1,ch2: string);
   begin
   repeat
   writeln('Donner ch1 non vide');
   readln(ch1);
   writeln('Donner ch2 non vide');
   readln(ch2);
   until (length(ch1)<>0) and (length(ch2)<>0) and (length(ch1)=length(ch2));
   end;
   function anag(ch1, ch2:string):boolean;
   var
   i,p:byte;
   test:boolean;
   begin
   test:=true;
   i:=1;
   repeat
     p:=pos(ch1[i],ch2);
   i:=i+1;
   until (i>length(ch1)) or (p=0);
   if p=0 then
   test :=false;
   anag:=test;
   end;
begin
  saisie(ch1,ch2);
  if anag(ch1, ch2) then
    writeln(ch1,' et ', ch2,' sont anagrammes')
  else
  writeln(ch1,' et ', ch2,' ne sont pas anagrammes')
end.
```

Exercice 16 :

Ecrire un programme qui permet de remplir un tableau T par N caractères alphabétique aléatoires (2<=N<=10) puis déterminer et afficher le nombre d'occurrence d'un caractère alphabétique donnée dans le tableau T.

Exemple :

T	A	f	r	A	W	c

- ✧ Si **Car ="r"** le programme affiche « le caractère **r** existe dans le tableau ».
- ✧ Si **Car="a"** le programme affiche « le caractère **a** n'existe pas dans le tableau».

```
program Ex16;
  uses WinCrt;
  type
  tab=array[1..10] of char;
  var
  m,n,i,j:byte;
  t:tab;
  c:char;
  procedure saisie(var n:byte);
  begin
  repeat
  writeln('Donner n dans [2..10]');
  readln(n);
  until n in [2..10];
  end;
  procedure rempli(var t:tab; n:byte);
    var i:byte;
  begin
  for i:=1 to n do
  repeat
  writeln('Donner le caracètre n° ', i);
  readln(t[i]);
  until upcase(t[i]) in ['A'..'Z'];
  end;
  function occ(t:tab;n:byte;c:char):byte;
  var
  i,oc:byte;
  begin
  oc:=0;
  for i:=1 to n do
  if t[i]=c then
   oc:=oc+1;
  occ:=oc;
  end;
begin
 saisie(n);
 rempli(t,n);
 writeln('Donner le caracètre recherché');
 readln(c) ;
 if occ(t,n,c)<>0 then
   writeln('le caractère ',c,' existe dans le tableau et apparait ',occ(t,n,c),' fois')
 else
 writeln('le caractère ',c,' n''existe pas dans le tableau');
end.
```