

- *Aucun document autorisé*
- *Durée : 1h30*
- *Veillez à bien indiquer vos noms, prénoms et à noircir les cases correspondant à votre numéro dans la fiche de réponse à rendre*
- *Barème :*
 - *réponse correcte = 1 point*
 - *réponse partiellement correcte = (nb de bonnes réponses cochées / nb bonnes réponses) point*
 - *au moins 1 réponse fausse cochée = -0,25 point*
 - *pas de réponse = 0 point*

1 Questions générales

Question no1

En quoi consiste la phase de spécification dans un cycle de développement logiciel ?

- **A** : à écrire le pseudo-code des fonctions et procédures
- **B** : à définir précisément le cahier des charges du programme à réaliser
- **C** : à écrire une analyse descendante

Question no2

En quoi consiste le processus de compilation ?

- **A** : à écrire un programme en Pascal
- **B** : à écrire un programme en pseudo-code
- **C** : à créer un programme exécutable à partir d'un code source

Question no3

La fonction ci-dessous doit renvoyer vrai si le nombre entier donné en paramètre est un nombre premier. Par quelle instruction faut-il remplacer la ligne ********* pour que cet algorithme soit correctement écrit ?

fonction estPremier (x : Entier) : Booleen

Déclaration i : Entier ; res : Booleen

debut

 res ← Vrai

pour i ← 2 à \sqrt{x} **faire**

si (x mod i) = 0 **alors**

```

    fin
finpour
retourner res
fin
— A : retourner res
— B : res ← Vrai
— C : res ← Faux

```

Question no4

Lesquelles de ces lignes sont correctes pour déclarer des variables en Pascal ?

- **A** : Integer a, b;
- **B** : var a, b : Integer;
- **C** : Déclaration : a : Entier;
- **D** : var a : Integer; b : Real;

Question no5

Est-ce que le programme suivant

```

program carre ;

var x : Integer ;

begin
  write( 'Entrez une valeur : ');
  readln(x);
  x := x*x;
  writeln( 'Son carre est : ',x);
end.

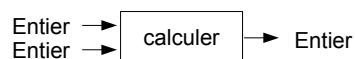
```

-
- **A** : fonctionne correctement
 - **B** : ne compile pas
 - **C** : provoque toujours une erreur à l'exécution

2 Fonctions et procédures

Question no6

Soit un sous-programme représenté par le schéma suivant :



Cocher les signatures pouvant lui correspondre.

- **A** : procedure calculer(E x,y :Entier ; S z :Entier)
- **B** : fonction calculer(a,b :Entier) : Entier

- C : procedure calculer(E/S x, y :Entier)

Question no7

- Quels éléments appartiennent à la signature d'une fonction ?
- A : son nom
 - B : son corps d'instruction
 - C : ses paramètres d'entrée
 - D : son type de retour

Question no8

- Qu'est ce qu'un paramètre d'entrée/sortie ?
- A : une variable dont la valeur sera modifiée par la procédure appelée
 - B : une valeur affichée à l'écran
 - C : une variable dont la valeur est nécessaire à la procédure appelée

Question no9

- Qu'affiche l'écran après l'exécution du programme suivant ?

```
fonction incr (b : Entier) : Entier
```

```
debut
```

```
    retourner b+1
```

```
fin
```

```
    Déclaration a,b : Entier
```

```
debut
```

```
    a ← 3
```

```
    b ← 10
```

```
    a ← incr(a)
```

```
    écrire(a)
```

```
fin
```

- A : 2
- B : 4
- C : 11
- D : b+1

Question no10

- Laquelle/Lesquelles de ces signatures est/sont bien écrite(s) ?
- A : fonction nomFonction (E a, b : Entier) : Entier
 - B : procedure nomProcedure (E a, b : Entier ; S c : Entier)
 - C : procedure nomProcedure (E a : Entier ; S b : Entier) : Entier
 - D : fonction nomFonction (a, b : Entier) : Entier

Question no11

- Étant donnée une fonction de signature $f(x, y : \text{Entier}) : \text{Entier}$, cocher ce qui pourrait correspondre à un programme principal correct.

- **A :**
Déclaration: x,y : Entier
debut
 lire(x,y)
 ecrire(f(x,y))
fin
- **B :**
Déclaration: a : Entier
debut
 lire(a)
 ecrire(f(a,a))
fin
- **C :**
Déclaration: e : Entier
debut
 e ← f(x,y)
 ecrire(e)
fin

Question no12

Qu'affiche l'écran après l'exécution du programme suivant ?

```

fonction carre (x : Entier) : Entier
debut
    retourner x * x
fin
fonction racine (x : Entier) : Entier
debut
    retourner  $\sqrt{x}$ 
fin
debut
    ecrire(racine(carre(4)))
fin

```

- **A :** 2
- **B :** 4
- **C :** 16
- **D :** racine(carre(4))

Question no13

Étant données les 2 signatures de sous-programmes et les déclarations de variables ci-dessous, lesquels de ces appels de procédure ou fonction sont valides ?

```

procedure permutation(var x, y : Integer);
fonction addition(x,y : Integer): Integer;
var a, b, c : Integer;

```

- **A :** permutation(a, b);
- **B :** permutation(a, 5);

- **C** : $c := \text{addition}(a, b)$;
- **D** : $c := \text{addition}(a, 5)$;

Question no14

Que fait cette fonction ?

```
function mystere(x: integer; y: integer): real;
var i : integer; p: real;
begin
  p := 1;
  for i := 1 to y do
    p := p * x;
  mystere := p;
end;
```

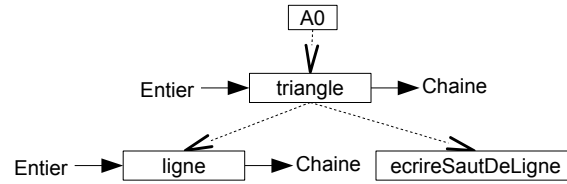
- **A** : Elle renvoie la valeur de x^y
- **B** : Elle renvoie la valeur de y^x
- **C** : Elle renvoie la valeur de $x * y$
- **D** : Elle ne compile pas

Question no15

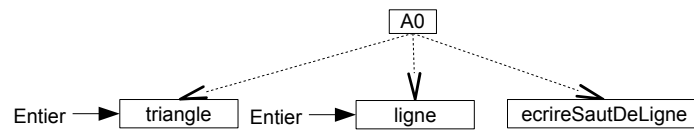
Quelle analyse descendante correspond au pseudo-code ci-dessous ?

```
procédure ligne (E x : Entier)
  Déclaration i : Entier
debut
  pour i ← 1 à x faire
    écrire('*')
  finpour
fin
procédure triangle (E x : Entier)
  Déclaration i : Entier
debut
  pour i ← 1 à x faire
    ligne(i)
    écrireSautDeLigne()
  finpour
fin
Déclaration lEntier : Entier
debut
  lire(lEntier)
  triangle(lEntier)
fin
```

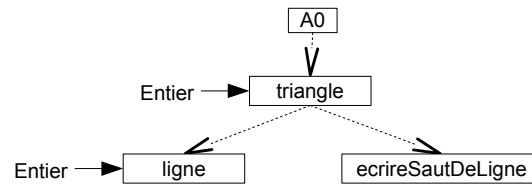
— A :



— B :



— C :



Question no16

Qu'affiche l'écran après l'exécution du programme suivant ?

procédure modifie (E/S unEntier : Entier)

Déclaration i : Entier

debut

pour i ← -10 à 1 pas de -2 faire

unEntier ← unEntier * 2

finpour

fin

Déclaration x : Entier

debut

x ← 1

modifie(x)

ecrire(x)

fin

-
- **A** : modifie
 - **B** : 1
 - **C** : 1024
 - **D** : 32

Question no17

Une variable locale à un sous-programme est :

- **A** : un paramètre effectif donné en entrée du sous-programme
- **B** : une variable utilisable par tous les sous-programmes d'un même fichier
- **C** : une variable utilisable seulement au sein d'un sous-programme

Question no18

Étant donnée une procédure de signature `calcul(E e:Entier; S s:Entier)`, cocher ce qui pourrait correspondre à une définition correcte par rapport au rôle attribué à chaque paramètre.

- **A** :
 debut
 `s ← e+e`
 fin
- **B** :
 debut
 `e ← s+s`
 fin
- **C** :
 debut
 `s ← e+s`
 fin

Question no19

Est-ce que le programme suivant

```
program cherche;  
  
function difference(annee : Integer): Integer;  
begin  
  difference := 2014 - annee;  
end;  
  
var year : Real;  
begin  
  write('Entrez_votre_date_de_naissance_:');  
  readln(year);  
  writeln('Vous_avez_autour_de_', difference(year), '_ans');  
end.
```

-
- **A** : fonctionne correctement
 - **B** : ne compile pas
 - **C** : provoque toujours une erreur à l'exécution

Question no20

Qu'affiche l'écran après l'exécution du programme suivant ?

procédure modifie (**E** a : **Entier** ; **S** b : **Entier**)

debut

a ← a * b

b ← b + 1

fin

Déclaration i, x, y : **Entier**

debut

x ← 1

y ← 1

pour i ← 1 à 5 **faire**

 modifie(x,y)

finpour

ecrire(x)

fin

- **A** : x
- **B** : 120
- **C** : 1

3 Tableaux

Question no21

Qu'est ce qu'un dépassement de tableau ?

- **A** : la déclaration d'un tableau de taille trop importante
- **B** : l'accès à un indice hors des indices définis lors de la déclaration du tableau
- **C** : la modification de la taille d'un tableau

Question no22

Parmi ceux-ci, quels types de données peuvent être utilisés pour les indices d'un tableau ?

- **A** : Réel
- **B** : Énumération
- **C** : Caractère
- **D** : Chaîne de caractère

Question no23

A quoi sert un tableau ?

- **A** : à stocker dans une variable plusieurs valeurs d'un même type
- **B** : à définir un nouveau type de donnée par une énumération de plusieurs valeurs
- **C** : à stocker dans une variable plusieurs valeurs de types différents
- **D** : à définir un nouveau type de donnée par un intervalle de valeurs possibles

Question no24

Que devient le tableau *tab* indicé de 1 à 3 s'il contient à l'origine ['c' 'b' 'a'] et si l'on fait :

```
tab[3] := tab[2];
```

```
tab[2] := tab[1];
```

```
tab[1] := tab[3];
```

- **A** : ['c' 'b' 'a']
- **B** : ['b' 'c' 'b']
- **C** : ['a' 'b' 'c']
- **D** : ['a' 'a' 'a']

Question no25

Quel code Pascal est valide ?

- **A** : Const MAX=100.0; Type MonTab=array[1..MAX] of Real;
- **B** : Const MAX=100; Type MonTab=array[10..MAX] of Real;
- **C** : Var max; Type MonTab=array[1..max] of Real;
- **D** : Type MonTab=array[100..1] of Real;

Question no26

Pour insérer une nouvelle valeur en début de tableau, il faut

- **A** : Seulement écrire la valeur à la première case du tableau
- **B** : Décaler chaque valeur du tableau dans la case qui suit, puis écrire la nouvelle valeur à la première case du tableau
- **C** : Tester que le tableau n'est pas plein
- **D** : Modifier la taille réelle du tableau pour y ajouter une case et la remplir avec la nouvelle valeur

Question no27

Pour supprimer la dernière valeur d'un tableau, il faut

- **A** : Remplir la dernière case avec la valeur 0
- **B** : Réduire la taille utilisée de 1
- **C** : Décaler chaque valeur du tableau dans la case qui précède, puis écrire 0 à la dernière case du tableau
- **D** : Tester que le tableau n'est pas vide

Question no28

On définit les paramètres suivants :

```
Const MIN = 1;
```

```
MAX = 8.7;
```

```
var minimum, maximum : Integer;
```

```
...
```

```
minimum := 1; maximum := 10;
```

Lesquels de ces types sont justes ?

- **A** : Type Tabl = array [1..MAX] of Integer ;
- **B** : Type Tabl = array [MIN..MAX] of Real ;
- **C** : Type Tabl = array [MIN..9] of Real ;
- **D** : Type Tabl = array [minimum..maximum] of Integer ;

Question n29

Considérons le programme suivant :

```
program essai;
```

```
var notes : array ['a'..'c'] of integer;
```

```
begin
```

```
  notes[1] := 'a';
```

```
  notes[2] := 'b';
```

```
  notes[3] := 'c';
```

```
end.
```

L'exécution de ce code, donne :

- **A** : Un tableau notes contenant [1 2 3]
- **B** : Un tableau notes contenant ['a' 'b' 'c']
- **C** : Ce programme ne compile pas

Question no30

Que voit-on d'affiché sur la console si l'on exécute le programme suivant :

```
program tabl;
```

```
var t : array [1..4] of integer ;
```

```
  i : Integer ;
```

```
begin
```

```
  for i := 1 to 5 do
```

```
    t[i] := i*2 ;
```

```
  for i := 5 downto 1 do
```

```
    write(t[i]) ;
```

```
end.
```

- **A** : Un texte imprévisible ou un message d'erreur
- **B** : 246810
- **C** : 8642
- **D** : 108642

Question no31

Etant donné les déclarations suivantes :

Constante MAX = 10

Type Suite = tableau[1..MAX] de Reel

procedure saisir(S s : Suite)

Cocher la/les définition(s) correcte(s) de la procédure saisir.

— A :

```
Déclaration: i : Entier
debut
  i ← 0
  tant que i < MAX faire
    i ← i+1
    lire(s[i])
  fintantque
fin
```

— B :

```
Déclaration: i : Entier
debut
  pour i ←-0 à MAX faire
    lire(s[i])
  finpour
fin
```

— C :

```
Déclaration: i : Entier
debut
  i ← 1
  tant que i ≤ MAX faire
    lire(s[i])
    i ← i+1
  fintantque
fin
```

Question no32

Pour représenter une durée exprimée en nombre d'heures, de minutes et de secondes, on définit les types suivants :

Type Unite = (heure,minute,seconde);

Duree = array[heure..seconde] of Integer;

function conversion(t : Duree) : Integer;

Cocher les définitions correctes de la fonction conversion.

— A : begin

```
conversion := Duree[1]*3600 + Duree[2]*60 + Duree[3]
end;
```

— B : begin

```
conversion := heure[t]*3600 + minute[t]*60 + seconde[t];
end;
```

— C : begin

```
conversion := t[heure]*3600 + t[minute]*60 + t[seconde];
end;
```

Question no33

Que fait la fonction suivante ?

```
Constante DW = 100
Type Zed = Tableau[1..DW] de Reel
fonction foo (a :Zed ; b :Naturel) : Reel
    Déclaration  i : Naturel
                 c : Reel

debut
    c ← 0
    pour i ←1 à b faire
        c ← c + (a[i]/b)
    finpour
    retourner c
fin
```

- A : Elle affiche les 100 éléments du tableau a.
- B : Elle affiche les 100 éléments du tableau Zed.
- C : Elle calcule la moyenne des b premiers éléments du tableau a.
- D : Elle calcule la moyenne des b premiers éléments du tableau Zed.

Question no34

Que fait la procédure suivante ?

```
Constante DW = 100
Type Zed = Tableau[1..DW] de Reel
procédure bar (E v : Reel, E/S t : Zed, n : Naturel)
    Déclaration  p : Naturel

debut
    si n < DW alors
        p ← n + 1
        tant que (p > 1) et (t[p - 1] > v) faire
            t[p] ← t[p - 1]
            p ← p - 1
        fintantque
        t[p] ← v
        n ← n + 1
    finsi
fin
```

- A : Elle ajoute la valeur v à la première position du tableau t.
- B : Elle ajoute la valeur v à la bonne position dans un tableau trié t.
- C : Elle supprime la valeur v du tableau t.
- D : Elle modifie la valeur située à l'indice v du tableau t en lui affectant la valeur suivante.

Question no35

Cocher les réponses vraies.

- **A** : dans un tableau `tab` trié par ordre croissant on a toujours $\text{tab}[i+1] \geq \text{tab}[i]$
- **B** : la recherche dichotomique dans un tableau trié a une complexité logarithmique
- **C** : un tableau trié occupe deux fois moins de place en mémoire qu'un tableau qui ne l'est pas
- **D** : la recherche dichotomique dans un tableau trié a une complexité linéaire

4 Tableaux à plusieurs dimensions

Question no36

Le nombre de dimensions d'un tableau

- **A** : peut être variable.
- **B** : est fixé à la déclaration du tableau.
- **C** : est limité à 2 maximum.

Question no37

Quel sera le contenu du tableau `tab` à la fin du programme suivant :

```
program tab2D;  
  
var tab : array [1..4,1..4] of Integer;  
    i, j : Integer;  
begin  
    for i := 1 to 4 do  
        for j := 1 to 4 do  
            tab[i][j] := i + j;  
        end;  
    end;  
end.
```

- **A** :
[1 2 3 4]
[2 2 3 4]
[3 3 3 4]
[4 4 4 4]
- **B** :
[1 2 3 4]
[2 3 4 5]
[3 4 5 6]
[4 5 6 7]
- **C** :
[2 3 4 5]
[3 4 5 6]
[4 5 6 7]
[5 6 7 8]

Question no38

Est-ce que le programme suivant

```
program pointage;  
  
Type Jours = (lundi , mardi , mercredi , jeudi , vendredi);  
  
var horaires : array[lundi..vendredi , 1..4] of Real;  
    i          : Integer; j : Jours;  
begin  
    for i := 1 to 4 do  
        begin  
            writeln('Entrez les temps de travail des jours de la semaine', i);  
            for j := lundi to vendredi do  
                readln(horaires[i][j]);  
            end;  
        end;  
    end.
```

- A : fonctionne correctement
- B : ne compile pas
- C : provoque parfois une erreur à l'exécution

Question no39

Cocher les déclarations valides en Pascal pour un tableau à plusieurs dimensions.

- A : var t : array['a'..'z', 1..20, -5..5] of Real;
- B : var n : Char; t : array['a'..n, 'a'..n] of Real;
- C : var t : array[1..10] of array ['A'..'L'] of array [1..20] of Real;
- D : var t : array[1..10, 1..20];

Question no40

On souhaite représenter dans un unique tableau la température moyenne (supposée entre -50,0 et +50,0 degrés) observée pour chaque année comprise entre 1950 et 2000, dans chaque département de France métropolitaine (désignés par leur numéro entre 1 et 95, la Corse étant représentée globalement par le numéro 20), à trois moments de la journée (matin, midi et soir). Pour représenter ce moment, le type suivant a été défini :

```
Type Moment = (matin, midi, soir);
```

Quel tableau parmi les propositions ci-dessous permet de stocker cette information

- A : var tab : array[-50..50, 1950..2000, 1..95] of Moment;
- B : var tab : array[1950..2000, 1..95] of Real;
- C : var tab : array[1..95, 1950..2000, matin..soir] of Real;
- D : var tab : array[-50..50, 1950..2000, 1..95, matin..soir] of Integer;