

- 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

- L'instruction *lire* en pseudo-code correspond à :
- **A** : un paramètre d'entrée d'une procédure
 - **B** : un paramètre de sortie d'une procédure
 - **C** : une instruction pour attendre une saisie au clavier

Question no2

Quelles sont les étapes, dans l'ordre, d'un processus classique de développement informatique ?

- **A** : Spécification, Conception Globale, Conception Détaillée, Implémentation
- **B** : Écriture de pseudo-code, Analyse Descendante, Implémentation
- **C** : Implémentation, Tests, Correction, Conception Globale

Question no3

A quoi correspond le niveau A0 d'une analyse descendante ?

- **A** : au programme principal
- **B** : à l'unité contenant toutes les procédures
- **C** : à l'ensemble des procédures et fonctions du programme

Question no4

La procédure ci-dessous doit réaliser une permutation des valeurs des paramètres a et b . Par quelle instruction faut-il remplacer la ligne `*****` pour que cet algorithme soit correctement écrit ?

procédure permutation (**E** a,b : **Naturel**)

Déclaration *****

debut

 c ← a

 a ← b

 b ← c

fin

— **A** : -

— **B** : c : **Naturel**

— **C** : a, b, c : **Naturel**

Question no5

Laquelle de ces lignes est correcte pour déclarer des variables en Pseudo-code ?

— **A** : Integer a, b;

— **B** : var a, b : Integer;

— **C** : Déclaration : a : Entier;

— **D** : var a : Integer; b : Real;

Question no6

A quoi sert l'instruction *var* en Pascal ?

— **A** : à déclarer des variables

— **B** : à déclarer des paramètres d'entrée

— **C** : à déclarer des paramètres de sortie

Question no7

Est-ce que le programme suivant

```
program pair ;
```

```
begin
```

```
  write('Entrez une valeur : ');
```

```
  readln(x);
```

```
  if ((x mod 2) = 0) then
```

```
    writeln('Ce nombre est pair')
```

```
  else
```

```
    writeln('Ce nombre est impair');
```

```
end.
```

— **A** : fonctionne correctement

— **B** : ne compile pas

— **C** : provoque toujours une erreur à l'exécution

Question no8

La fonction ci-dessous doit renvoyer la somme de tous les nombres pairs jusqu'au paramètre d'entrée x . Par quelle instruction faut-il remplacer la ligne ********* pour que cet algorithme soit correctement écrit ?

fonction somme (x : **Naturel**) : **Naturel**

Déclaration i : **Entier**; res : **Naturel**

debut

 res \leftarrow 0

pour i \leftarrow 2 à x **pas de 2 faire**

finpour

retourner res

fin

- **A** : retourner res
- **B** : res \leftarrow res + 2 * i
- **C** : res \leftarrow res + i

Question no9

Quand une variable est déclarée localement dans un sous-programme, elle est visible :

- **A** : dans tous les sous-programmes du fichier source du programme
- **B** : dans le sous-programme où elle est déclarée ainsi que dans les autres sous-programmes qui y sont appelés
- **C** : uniquement dans le sous-programme où elle est déclarée

2 Fonctions et procédures

Question no10

Qu'est ce qu'un paramètre d'entrée ?

- **A** : une variable dont la valeur sera modifiée par la procédure appelée
- **B** : une valeur affichée à l'écran
- **C** : une valeur nécessaire à la procédure appelée

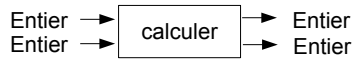
Question no11

Lesquelles de ces signatures sont bien écrites ?

- **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 no12

Soit un sous-programme représenté par le schéma suivant :



- Cocher les signatures pouvant lui correspondre.
- **A** : procédure calculer(E x,y :Entier ; S w,z :Entier)
 - **B** : fonction calculer(a,b :Entier) : Entier, Entier
 - **C** : procédure calculer(E/S x, y :Entier)

Question no13

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

```

fonction incr (x : Entier) : Entier
debut
  retourner x*2
fin

  Déclaration x,y : Entier

debut
  y ← 2
  x ← 10
  y ← incr(y)
  ecrire(y)
fin
  
```

- **A** : 2
- **B** : 4
- **C** : 20
- **D** : x*2

Question no14

Étant donnée une procédure de signature `proc(E x : Entier; S y : Entier)`, cocher les propositions qui pourraient être un programme principal correct.

- **A** :
 Déclaration: x,y : Entier
debut
 lire(x)
 y ← proc(x)
 ecrire(y)
fin
- **B** :
 Déclaration: x,y : Entier
debut
 lire(x)
 proc(x,y)
 ecrire(y)
fin

— **C** :

```

Déclaration: x,y : Entier
debut
  lire(y)
  proc(y,x)
  ecrire(x)
fin

```

Question no15

Que produit l'exécution du programme suivant ?

```

fonction carre (x : Entier) : Entier
debut
  retourner x * x
fin
procédure additionner (E a,b : Entier ; S c : Entier)
debut
  c ← a + b
fin
Déclaration v : Entier
debut
  additionner(2,carre(2),v)
  ecrire(v)
fin

```

- **A** : l'écriture de ce programme ne compilera pas
- **B** : il affiche v
- **C** : il affiche 6
- **D** : il affiche 2

Question no16

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);
function addition(x,y : Integer): Integer;
var a, b, c : Integer;

```

- **A** : permutation(a, b);
- **B** : permutation(a, 5);
- **C** : c := addition(a, b);
- **D** : c := addition(a, 5);

Question no17

Que fait cette fonction ?

```
function mystere(x: integer): real;  
var i : integer; p: real;  
begin  
    p := 1;  
    for i := 1 to 10 do  
        p := x * x;  
        mystere := p;  
end;
```

- **A** : Elle renvoie la valeur de x^{10}
- **B** : Elle renvoie la valeur de x^2
- **C** : Elle renvoie la valeur de $x * 10$
- **D** : Elle ne compile pas

Question no18

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

```
fonction concatene (x : Entier ; y : Caractere) : Chaine de caracteres
```

```
    Déclaration i : Entier ; res : Chaine de caracteres
```

```
debut
```

```
    res ← ""
```

```
    pour i ← 1 à x faire
```

```
        res ← res + y
```

```
    finpour
```

```
    retourner res
```

```
fin
```

```
procédure genererChaine (E a : Caractere)
```

```
    Déclaration str : Chaine de caracteres
```

```
debut
```

```
    str ← concatene(10, a)
```

```
    ecrire(str)
```

```
fin
```

```
    Déclaration leCar : Caractere
```

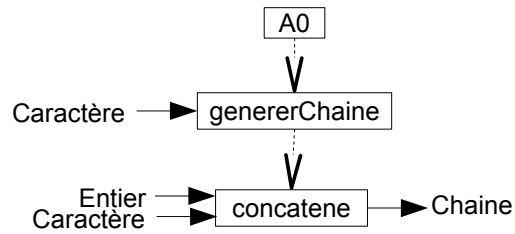
```
debut
```

```
    lire(leCar)
```

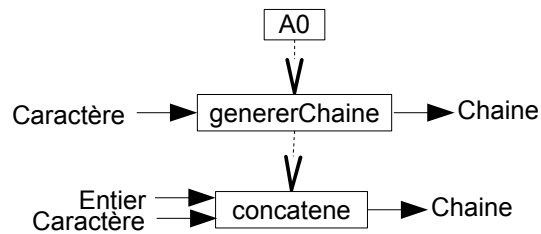
```
    genererChaine(leCar)
```

```
fin
```

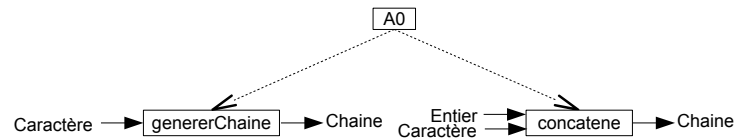
— A :



— B :



— C :



Question no19

Qu'est ce qui est affiché à l'écran par le programme de la question précédente quand l'utilisateur saisit le caractère A lors de l'affectation de la variable leCar ?

- A : ce programme ne fonctionne pas
- B : 10A
- C : AAAAAAAAAA
- D : Aucune de ces réponses

Question no20

Que produit l'exécution du programme suivant ?

procédure calculer (E a,b : Entier ; S c : Entier)

Déclaration i : Entier
debut
 c ← 1
 pour i ← 1 à b **faire**
 c ← c * a
 finpour
fin
Déclaration x : Entier
debut
 x ← calculer(2,4)
 ecrire(x)
fin

- **A** : l'écriture de ce programme ne compilera pas
- **B** : il affiche 8
- **C** : il affiche 16

Question no21

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

procédure modifie (E/S unEntier : Entier ; E x : Entier)

Déclaration i : Entier
debut
 pour i ← 10 à 1 **pas de** x **faire**
 unEntier ← unEntier * 2
 finpour
fin
Déclaration x : Entier
debut
 x ← 1
 modifie(x,-2)
 ecrire(x)
fin

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

3 Tableaux

Question no22

A quoi correspond le type de donnée tableau ?

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

Question no23

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 no24

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 no25

Lesquelles de ces affirmations sont correctes ?

- **A** : La taille réelle et le type de donnée d'un tableau sont fixés à la déclaration
- **B** : Un tableau peut contenir des valeurs de types de donnée différents
- **C** : Un tableau peut contenir des valeurs différentes d'un même type de donnée
- **D** : Un tableau définit un ensemble de valeurs d'un type de donnée

Question no26

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 no27

Pour supprimer une valeur en début de tableau, quelles sont les opérations à réaliser ?

- **A** : Affecter 0 à la première case du tableau
- **B** : Décaler chaque valeur du tableau dans la case qui précède
- **C** : Réduire la taille utilisée du tableau de 1
- **D** : Détruire la première case du tableau

Question no28

Quelle est la complexité d'un algorithme qui parcourt tous les éléments d'un tableau à une dimension

- **A** : constante $O(1)$
- **B** : linéaire $O(n)$
- **C** : quadratique $O(n^2)$
- **D** : exponentielle $O(e^n)$

Question n29

Considérons le programme suivant :

```
program essai ;  
  
var notes : array [1..3] 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 tab1 ;  
var t : array [1..5] 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

Qu'affiche le programme suivant :

```
procédure remplir (E x : Entier ; S tab : Tableau[1..10] de Entier )
```

```
  Déclaration i : Entier
```

```
debut
```

```
  pour i ← 1 à 10 faire
```

```
    tab[i] ← x+i
```

```
  finpour
```

```
fin
```

```
  Déclaration i : Entier ; t : Tableau[1..10] de Entier
```

```
debut
```

```
  remplir(5,t)
```

```
  pour i ← 1 à 10 faire
```

```
    écrire(t[i])
```

```
  finpour
```

```
fin
```

- A : L'écriture de ce programme ne compilera pas
- B : 12345678910
- C : 6789101112131415

Question no32

Pour représenter une masse exprimée en 3 valeurs entières correspondant aux tonnes, kilogrammes et grammes, nous définissons les types suivants :

```
Type Unite = (tonne,kilo,gramme);
```

```
Type Masse = array[tonne..gramme] of Integer;
```

```
function conversion(m : Masse) : Real;
```

Cocher la définition correcte de la fonction conversion.

- A : begin
 conversion := Masse[1]*1000 + Masse[2] + Masse[3]/1000
end;
- B : begin
 conversion := tonne[1]*1000 + kilo[2] + gramme[3]/1000
end;
- C : begin
 conversion := m[tonne]*1000 + m[kilo] + m[gramme]/1000
end;

Question no33

Que fait la procédure suivante ?

```
Constante DW = 100
```

Type Zed = **Tableau**[1..DW] **de Reel**
procédure uneProc (**E/S** t : Zed, a : **Naturel**)

Déclaration i : **Naturel**

debut

si a >= 1 **alors**

pour i ← 1 à a-1 **faire**

t[i] ← t[i+1]

finpour

a ← a - 1

finsi

fin

- **A** : Elle affecte la valeur **a** à toutes les cases du tableau **t**.
- **B** : Elle ajoute la valeur **a** à la fin d'un tableau **t**.
- **C** : Elle supprime le premier élément d'un tableau **t** dont la taille utilisée est **a**.
- **D** : Elle ajoute une nouvelle valeur à un tableau **t** dont la taille utilisée est **a**.

Question no34

Que fait le programme suivant ?

Type Tab1D = **Tableau**[1..5] **de Caractere**
procédure remplir (**E/S** t : Tab1D)

Déclaration c : **Caractere**; i : **Entier**

debut

i ← 1

pour c ← 'a' à 'j' **faire**

t[i] ← c

i ← i+1

finpour

fin

Déclaration i : **Entier**; t : Tab1D

debut

remplir(t)

pour i ← 1 à 10 **faire**

ecrire(t[i])

finpour

fin

- **A** : il affiche toujours 'abcdefghij'
- **B** : il affiche toujours '12345678910'
- **C** : l'écriture de ce programme ne compilera pas
- **D** : il provoque parfois une erreur à l'exécution

Question no35

Que fait la procédure suivante ?

procédure maProc (**E** x : Reel, **E/S** tab : Tableau[1..100] de Reel , y : Naturel)

Déclaration z : Naturel

debut

si y < 100 **alors**

z ← y + 1

tant que (z > 1) et (tab[z - 1] > x) **faire**

tab[z] ← tab[z - 1]

z ← z - 1

fin tant que

tab[z] ← x

y ← y + 1

finsi

fin

- **A** : Elle ajoute la valeur x à la première position du tableau **tab**.
- **B** : Elle ajoute la valeur x à la bonne position dans un tableau trié **tab**.
- **C** : Elle supprime la valeur x du tableau **tab**.
- **D** : Elle modifie la valeur située à l'indice x du tableau **tab** en lui affectant la valeur suivante.

4 Fichiers

Question no36

Quelle séquence d'instructions doit-on utiliser pour écrire dans un fichier en Pascal

- **A** : assign, reset, write, close
- **B** : assign, append, write, close
- **C** : assign, write, close
- **D** : write, reset, close

Question no37

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

```
program fichier ;  
  
var fic : Text; var chaine : string;  
begin  
  assign ( fic , 'monFichier.txt' );  
  while not (eof ( fic )) do  
    begin  
      readln ( fic , chaine );  
      writeln ( chaine );  
    end  
end
```

```

        end;
    close( fic );
end.

```

- **A** : Ce programme ne compile pas
- **B** : Il provoque une erreur à l'exécution
- **C** : Il affiche le texte contenu dans le fichier monFichier.txt

Question no38

Que fait le programme suivant

```

program fichier;

var fic : Text; var chaine : string;
begin
    assign( fic , 'monFichier.txt' );
    rewrite( fic );
    write( fic , 'hello_' );
    close( fic );
    assign( fic , 'monFichier.txt' );
    rewrite( fic );
    write( fic , 'world' );
    close( fic );
end.

```

- **A** : Ce programme ne compile pas
- **B** : Il provoque une erreur à l'exécution
- **C** : Il modifie le fichier monFichier.txt qui contiendra uniquement le texte 'world'
- **D** : Il modifie le fichier monFichier.txt qui contiendra uniquement le texte 'hello world'

Question no39

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

```

program fichier;

var fic : Text; var chaine : string;
begin
    assign( fic , 'monFichier.txt' );
    rewrite( fic );
    writeln( fic , 'bonjour' );
    writeln( fic , 'le' );
    writeln( fic , 'monde' );
    close( fic );
    assign( fic , 'monFichier.txt' );
    reset( fic );
    while not( eof( fic ) ) do

```

```

        readln(fic , chaine);
    writeln(chaine);
    close(fic);
end.

```

- **A** : Ce programme ne compile pas
- **B** : Il affiche 'bonjour le monde'
- **C** : Il affiche 'monde'
- **D** : Il affiche 'bonjour'

Question no40

Que fait le programme suivant

```

program fichier;

var fic : Text; var tab : array[1..100] of string;
    i, j : Integer; str : string;
begin
    readln(str);
    assign(fic , 'monFichier.txt ');
    reset(fic);
    i := 1;
    repeat
        readln(fic , tab[i]);
        i := i+1;
    until (eof(fic)) or (i = 100) or (str < tab[i-1]);
    writeln(i);
    if (i <> 100) then
        begin
            if (str < tab[i-1]) then
                begin
                    tab[i] := tab[i-1];
                    tab[i-1] := str;
                end
            else tab[i] := str;
                i := i + 1;
            end;
        while (not eof(fic)) and (i <= 100) do
            begin
                readln(fic , tab[i]);
                i := i+1;
            end;
        close(fic);
        assign(fic , 'monFichier.txt ');
        rewrite(fic);
        for j := 1 to i-1 do
            writeln(fic , tab[j]);
        close(fic);
    end.

```

- **A** : Il insère un texte saisi au clavier à la bonne position dans un fichier dont les lignes sont classées par ordre alphabétique
- **B** : Il provoque une erreur à l'exécution
- **C** : Ce programme ne compile pas
- **D** : Il modifie le fichier monFichier.txt pour inverser l'ordre dans lequel les lignes sont écrites