

L'usage de la calculatrice est interdit

Examen final– Structure des Ordinateurs et Applications

Questions de cours : [4 points]

1. Définir les termes suivants : algorithme, organigramme, programme, compiler et Exécuter. (1 points)
2. Quelle est la différence entre une **boucle FOR** et une **boucle WHILE** ? (1 points)
3. Quand utilise-t-on **REPEAT...UNTIL** plutôt que **WHILE** ? (1 points)
4. Qu'est-ce qu'une **structure conditionnelle** ? Donnez un exemple. (1 points)

Exercice 01 : [4 points]

1. Effectuer les conversions suivantes : $(110)_4 = (?)_{16} = (?)_8$, $(85)_{10} = (?)_2$ (1 points)
2. Traduire les expressions suivantes en langage Pascal (on rappelle que : $a^b = e^{b \ln(a)}$) (1 points)

$$E1 = \frac{\sqrt{|a-b|+c^2} - \ln(x+2)}{e^{y \ln(z)} + 1}, \quad E2 = \frac{4^2 + a^{3b} + \sqrt{e^x + d}}{\sqrt{y}}$$

3. Évaluer l'expression suivante en respectant l'ordre de priorité des opérateurs : (1 points)

$$\text{Exp1} = a + b * (e - f) / (c - d / 2)$$

Avec : $a = 4, b = 6, c = 3, d = 2, e = 5, f = 1$

4. Donner l'expression arithmétique correspondante à l'expression suivante écrite en Pascal : (1 points)

$$\text{Exp2} := \exp(\text{sqrt}(x + 1)) / (\text{abs}(a - b) + \ln(c))$$

Exercice 02 : [8 points]

Soit l'algorithme suivant :

Algorithme Exo2 ;

Variables

S : réel ;

J, F, N : entier ;

Début

Lire(N) ;

S ← 0 ;

F ← 1 ;

Pour J ← 1 à N faire

F ← F × J ;

S ← S + 1 / F ;

Fin Pour

Écrire("La somme S =", S:4:3) ;

Fin.

Questions :

1. Traduire l'algorithme en programme **PASCAL**. (2 points)
2. Dérouler l'algorithme pour **N = 4**. (2 points)
3. Dédurre l'expression mathématique générale de la somme S. (1.5 point)
4. Modifier le programme de sorte qu'il affiche, à chaque itération de J, la valeur de F. (1.5 point)
5. Réécrire l'algorithme en utilisant une boucle **tant que**. (1 point)

Exercice 03 : [4 points]

Une compagnie aérienne souhaite automatiser la réservation de ses vols. Écrire un programme qui saisit le nombre de places disponibles, le nombre de billets à réserver et le prix d'un billet, en distinguant les **places premium et standard**. Le programme doit vérifier si la réservation est possible (la réservation est possible seulement si le nombre de billets demandés est inférieur ou égal au nombre de places disponibles), puis calculer et afficher le **nombre total de billets réservés** ainsi que le **montant total à payer**.

Bonne chance

Corrigé de l'Examen Structure des Ordinateurs et Applications

L'usage de la calculatrice est interdit

Questions de cours : [4 points]

1. Définir les termes suivants : algorithme, organigramme, programme, compiler et Exécuter. (1 points)

Algorithme	Organigramme	Programme	Compiler et Exécuter
0.25	0.25	0.25	0.25
Une suite d'instructions permettant de résoudre un problème étape par étape qui sont ordonnées logiquement.	Une représentation graphique d'un algorithme avec des symboles et des flèches.	Un ensemble d'instructions écrites dans un langage de programmation pour être exécuté par un ordinateur.	Compiler : Transformer un programme écrit en langage humain (ex. Pascal) en code exécutable par l'ordinateur. Exécuter : Lancer le programme pour effectuer les calculs et produire les résultats.

2. Quelle est la différence entre une boucle FOR et une boucle WHILE ? (1 points)

Boucle For	Boucle While
0.25	0.25
<ul style="list-style-type: none"> Le nombre <u>d'itérations est fixé dès le départ.</u> On utilise un compteur qui <u>s'incrémente ou se décrémente automatiquement à chaque passage.</u> 	<ul style="list-style-type: none"> Le nombre <u>d'itérations est indéterminé à l'avance.</u> La boucle <u>continue tant qu'une condition est vraie.</u>
0.25	0.25

3. Quand utilise-t-on REPEAT...UNTIL plutôt que WHILE ? (1 points)

- WHILE vérifie la condition **avant** d'exécuter le bloc → la boucle peut ne jamais s'exécuter si la condition est fausse.
- REPEAT...UNTIL exécute d'abord le bloc, puis teste la condition → la boucle s'exécute toujours **au moins une fois**.

4. Qu'est-ce qu'une structure conditionnelle ? Donnez un exemple. (1 points)

Une structure conditionnelle est un ensemble d'instructions qui s'exécutent seulement si une condition donnée est vraie. Elle sert à contrôler le déroulement du programme selon des critères précis.

Exemple correct :

```
Program exmp ;
var
  T: integer;
begin
  writeln('Entrez la temperature T : ');
  readln(T);
  if T > 30 then
    writeln('Il fait chaud.')
  else
    writeln('Il ne fait pas trop chaud.');
```

Corrigé de l'exercice 01 : [4 points]

Effectuer les conversions suivantes : $(110)_4 = (?)_{16} = (?)_8$, $(85)_{10} = (?)_2$ (1 points)

A) Méthode 1 : Passage par la base 2 :

1. **Vers la base 16 : (méthode = 0,125 Résultat correct = 0,25)**

- Convertir chaque chiffre de base 4 en 2 bits : $(1)_4 = (01)_2$, $(1)_4 = (01)_2$, $(0)_4 = (00)_2$ on aura: $(110)_4 = (01\ 01\ 00)_2$
- Grouper les bits par 4 (en partant de la droite) : $(01\ 01\ 00)_2 \rightarrow \backslash 0001\ \backslash 0100$; $(0001)_2 = (1)_{16}$, $(0100)_2 = (4)_{16}$
Donc : $(110)_4 = (14)_{16}$

2. **Vers la base 8 : (méthode = 0,125 Résultat correct = 0,25)**

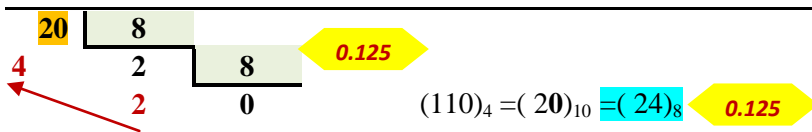
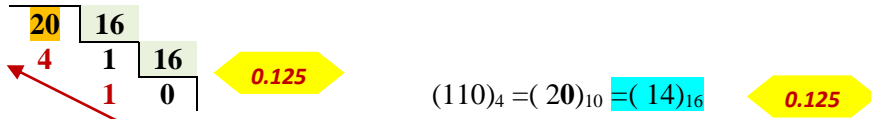
- Grouper les bits par 3 (en partant de la droite) : $(01\ 01\ 00)_2 \rightarrow \backslash 010\ \backslash 100$; $(010)_2 = (2)_8$, $(100)_2 = (4)_8$
Donc : $(110)_4 = (14)_{16} = (24)_8$

B) Méthode 2 : Passage par la base 10 :

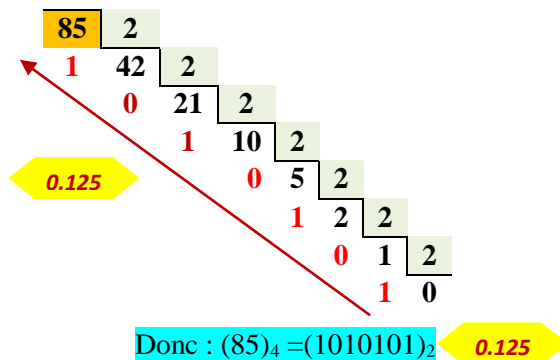
$$(110)_4 = (0 \cdot 4^0 + 1 \cdot 4^1 + 1 \cdot 4^2)_{10} = (20)_{10}$$

$$= (0 + 4 + 16)_{10}$$

$$= (20)_{10} \quad \text{0.25}$$



- Conversion de $(85)_{10} = (?)_2$



Donc : $(85)_{10} = (1010101)_2$ 0.125

C) Traduire les expressions suivantes en langage Pascal (on rappelle que $a^b = e^{b \ln(a)}$) (1 points)

$$E1 = \frac{\sqrt{|a-b| + c^2} - \ln(x+2)}{e^{y \ln(z)} + 1}, \quad E2 = \frac{4^2 + a^{3b} + \sqrt{e^x + d}}{\sqrt{y}}$$

$a^b = \exp(b \cdot \ln(a))$; $\text{sqrt}(x)$ = racine carrée ; $\text{abs}(x)$ = valeur absolue ; $\ln(x)$ = logarithme naturel $\exp(x) = e^x$

$E1 := (\text{sqrt}(\text{abs}(a - b) + \text{sqr}(c)) - \ln(x + 2)) / (\exp(y * \ln(z)) + 1);$ 0.5

$E2 := (\text{sqr}(4) + \text{Exp}(3 * b * \ln(a)) + \text{sqrt}(\exp(x) + d)) / \text{sqrt}(y);$ 0.5

D) Évaluer l'expression suivante en respectant l'ordre de priorité des opérateurs : (1 points)

$$\text{Exp1} = a + b * \frac{(e-f)}{(c-\frac{d}{2})}, \text{ Avec : } a = 4, b = 6, c = 3, d = 2, e = 5, f = 1$$

$\text{Exp1} := a + b * (e - f) / (c - d / 2);$ 0.125

$\text{Exp1} := 4 + 6 * (5 - 1) / (3 - 2 / 2);$ 0.125

$\text{Exp1} := 4 + 6 * 4 / (3 - 2 / 2);$ 0.125

$\text{Exp1} := 4 + 6 * 4 / (3 - 1);$ 0.125

$\text{Exp1} := 4 + 6 * 4 / 2;$ 0.125

$\text{Exp1} := 4 + 24 / 2;$ 0.125

$\text{Exp1} := 4 + 12;$ 0.125

$\text{Exp1} := 16;$ 0.125

E) Donner l'expression arithmétique correspondante à l'expression suivante écrite en Pascal : (1 points)

$$Exp2 := \exp(\text{sqrt}(x + 1)) / (\text{abs}(a - b) + \ln(c))$$

$$Exp2 = \frac{e^{\sqrt{x+1}}}{|a-b| + \ln(c)} \quad \text{01}$$

Corrigé de l'exercice 02 : [8 points]

1. Traduction de l'algorithme en programme **PASCAL** (2 points)

```

program Exo2; 0.125
var 0.125
  S : real; 0.125
  J, F, N : integer; 0.125
begin 0.125
  read(N); 0.125
  S := 0; 0.125
  F := 1; 0.125

  for J := 1 to N do 0.125
  begin
    F := F * J; 0.125
    S := S + 1 / F; 0.125
  end;
  write('La somme S = ', S:4:3); 0.125
end. 0.125

```

Note: A yellow callout '0.375' points to the loop body.

2. Déroulement de l'algorithme pour N = 4 (2 points)

Instructions	Variables				Affichage
	S	J	F	N	
Lire(N);				4	0.125
S ← 0;	0				0.125
F ← 1;		1			0.125
for J := 1		1			0.125
F := F * J; F := 1 * 1;					
F := 1;			1		0.125
S := S + 1 / F; S := 0 + 1 / 1;					
S := 1;	1				0.125
for J := 2		2			0.125
F := F * J; F := 1 * 2;					
F := 2;			2		0.125
S := S + 1 / F; S := 1 + 1 / 2;					
S := 1.5;	1.5				0.125
for J := 3		3			0.125
F := F * J; F := 2 * 3;					
F := 6;			6		0.125
S := S + 1 / F; S := 1.5 + 1 / 6;					
S := 1.6667;	1.6667				0.125
for J := 4		4			0.125
F := F * J; F := 6 * 4;					
F := 24;			24		0.125
S := S + 1 / F; S := 1.6667 + 1 / 24;					
S := 1.7084;	1.7084				0.125
Écrire("La somme S =", S:4:3);					0.125
					La somme S =1.7084

3. Expression mathématique générale de la somme S (1.5 point)

La somme calculée est : $S = \sum_{J=1}^N \frac{1}{J!}$ 1.5

<p>4. Modifier le programme de sorte qu'il affiche, à chaque itération de J, la valeur de F. (1.5 point)</p>	<p>5. Réécriture de l'algorithme en utilisant une boucle While (1 point)</p>
<pre> program Exo2; var S : real; J, F, N : integer; begin read(N); S := 0; F := 1; for J:= 1 to N do begin F := F * J; S := S + 1 / F; writeln('Iteration ', J, ' : F = ', F); end; write('La somme S = ', S:4:3); end. </pre>	<pre> Algorithme Exo2; Variables S : réel ; J, F, N : entier ; Début Lire(N) ; S ← 0 ; F ← 1 ; J ← 1 ; Tant que J ≤ N faire F ← F × J ; S ← S + 1 / F ; J ← J + 1 ; Fin Tant que ; Écrire("La somme S = ", S:4:3) ; Fin. </pre>

Corrigé de l'exercice 03: (04 points)

Programme en Pascal

```

program Exo_03;
var
  Nplaces: integer; // Nombre total de places disponibles
  Bstd: integer; // Nombre de billets standard
  Bpre: integer; // Nombre de billets premium
  Pstd: real; // Prix d'un billet standard
  Ppre: real; // Prix d'un billet premium
  Btotal: integer; // Nombre total de billets réservés
  Montant: real; // Montant total à payer
begin
  // Saisie des données
  writeln(' Entrez le nombre total de places disponibles : ');
  readln(Nplaces);
  writeln(' Entrez le nombre de billets standard à réserver : ');
  readln(Bstd);
  writeln(' Entrez le prix d un billet standard : ');
  readln(Pstd);
  writeln(' Entrez le nombre de billets premium à réserver : ');
  readln(Bpre);
  writeln(' Entrez le prix d un billet premium : ');
  readln(Ppre);

  Btotal := Bstd + Bpre; // Calcul du nombre total de billets

  if Btotal > Nplaces then // Vérification de la disponibilité
    writeln('Reservation impossible : pas assez de places disponibles.')
  else
    begin
      Montant := (Bstd * Pstd) + (Bpre * Ppre); // Calcul du montant total à payer
      // Affichage des résultats
      writeln('Reservation possible. ');
      writeln('Nombre total de billets réservés : ', Btotal);
      writeln('Montant total à payer : ', Montant:6:2, ' DA');
    end;
end.

```

0,125 { Variables correctes : **0,125 * 7**

0,125 * 10 {

0.25 {

0.25 {

0.25 {

0.25 {

0.125 * 3 {

0.125 {