

ÉPREUVE REGROUPEE

Discipline: INFORMATIQUE

Cours: 1 IN DP 08 / 09

Date: 11/12/2025

Durée: 105 minutes

Maître: JC Barros

Matériel autorisé:

- Aucun (aide-mémoire en fin d'épreuve)

Consignes de travail:

- Ecriture lisible et propre demandée
- Mettre en évidence les résultats
- 11 pages dont 3 d'aide-mémoire à la fin

Nom et prénom:

Total: 62 points + 6 bonus

Ex 1.

/ 3 pts

Expliquez brièvement l'importance de l'un des trois sujets suivants du point de vue de l'histoire de l'informatique.

John von Neumann **OU** Alan Turing **OU** Loi de Moore

Explication:

Ex 2.

/ 2 pts

Un smartphone est-il un ordinateur ? Justifiez votre réponse.

Ex 3.

/ 5 pts

Répondre aux QCM suivants. Une seule case est correcte!

Si plusieurs cases sont cochées dans un QCM, ce QCM sera nul.

a.

(1 pts)

Quel composant permet de stocker des données de manière volatile ?

- ☐ La RAM (mémoire vive)
- ☐ La ROM (mémoire morte)
- ☐ Le disque dur
- ☐ L'alimentation électrique
- ☐ La carte graphique

b.

(1 pts)

Comment nomme-t-on un dispositif matériel qui permet à l'utilisateur de saisir des données dans un système informatique?

- ☐ Une télécommande
- ☐ Un périphérique d'entrée
- ☐ Une carte SD
- ☐ Un périphérique de stockage
- ☐ Un périphérique de sortie

c.

(1 pts)

Quel acronyme désigne le processeur ?

- ☐ MHz
- ☐ OS
- ☐ CPU
- ☐ HDD
- ☐ I/O

d.

(1 pts)

Quel est l'élément qui contient les instructions que doit suivre le processeur au démarrage de la machine?

- ☐ La carte graphique
- ☐ Le système d'exploitation
- ☐ Le BIOS
- ☐ Le North Bridge
- ☐ Le South Bridge

e.

(1 pts)

A quoi correspond un octet en informatique ?

- ☐ à la fréquence de l'horloge
- ☐ à 8 bits
- ☐ à la base 8
- ☐ à un bug
- ☐ à la mémoire cache du processeur

Ex 4.

/ 4 pts

Durant le festival de Cannes, les jurés doivent visionner 6 films par jour pendant 10 jours. L'organisateur souhaite enregistrer tous les films sur une clé USB pour les jurys. Sachant que chaque enregistrement fait environ 700 Mo, combien de Go au minimum doit avoir la clé USB ?

Montrer le détail de vos calculs pour justifier.

Ex 5.**/ 6 pts**

Voici des informations sur deux ordinateurs :

Ordinateur A

Processeur : Intel Core i5-8350U 1.7 GHz

Mémoire : 16 Go RAM

Stockage : 256 Go SSD

Carte graphique : Intel UHD Graphics 620

Taille d'écran : 15.6 pouces

Écran : Full-HD (1920 x 1080)

Système d'exploitation : Windows 11

Ordinateur B

Processeur : Intel Core i5-7200U 2.50 GHz

Mémoire : 8 Go RAM

Stockage : 500 Go SSD

Carte graphique : Intel HD 620

Taille d'écran : 14 pouces

Écran : QHD (2560 x 1440)

Système d'exploitation : Windows 10

a.

(2 pts)

Alice cherche un ordinateur qui soit capable de traiter une très grande quantité de calculs mathématiques.

Entourez l'ordinateur qui convient le mieux : **A / B**

Justifiez avec les informations pertinentes ci-dessus :

b.

(2 pts)

Bob a besoin de lancer beaucoup d'applications en même temps pour son travail.

Entourez l'ordinateur qui convient le mieux : **A / B**

Justifiez avec les informations pertinentes ci-dessus :

c.

(2 pts)

Carole est cinéophile. Elle a besoin d'enregistrer une grande quantité de films sur son ordinateur.

Entourez l'ordinateur qui convient le mieux : **A / B**

Justifiez avec les informations pertinentes ci-dessus :

Ex 6.**/ 4 pts**

a. A quoi sert un système d'exploitation?

(2pts)

b. Donnez deux exemples de systèmes d'exploitation.

(2pts)

Ex 7.**/ 14 pts (dont 6 BONUS)**

a. Convertissez de décimal ₍₁₀₎ en binaire ₍₂₎ :

(2 pts)

$58_{(10)} = \dots\dots\dots_{(2)}$

b. Convertissez de binaire ₍₂₎ en décimal ₍₁₀₎ :

(2 pts)

$1010110_{(2)} = \dots\dots\dots_{(10)}$

c. Coder le texte suivant en binaire à l'aide de la table ASCII

(2 pts)

SALUT !

d.

(2 pts)

Une photographie est enregistrée dans deux fichiers: l'un en noir et blanc et l'autre en couleur. Lequel occupera plus de place en mémoire ? Expliquez pourquoi ?

e. BONUS (e. f. et g.)

(2 pts)

Combien d'octets mesure le fichier de la photographie en noir et blanc si elle est constituée de 100 x 80 pixels ?

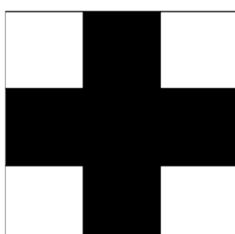
f. Combien d'octets mesure le fichier de la photographie en couleur ?

(2 pts)

On a le même nombre de pixels et le format RVB avec 8 bits pour chaque couleur.

g. Donnez une **représentation numérique** de l'image ci-dessous (9 pixels).

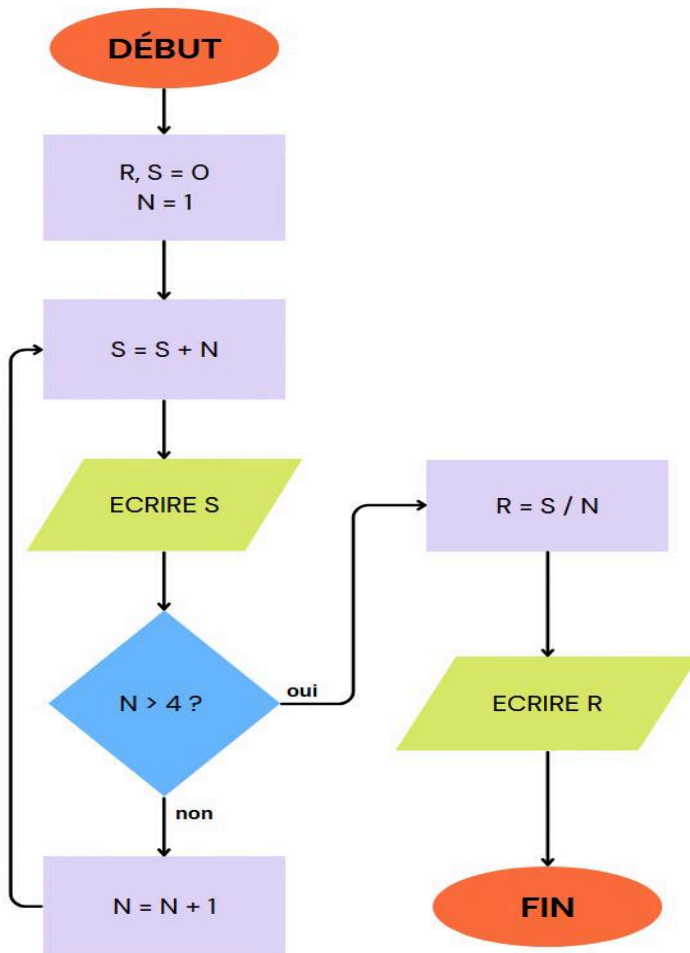
(2 pts)



Ex 8.

/ 7 pts

Voici un algorithme sous forme de logigramme:



- a. Pour justifier les réponses de cet exercice, déroulez l'algorithme étape par étape et créez le **tableau des valeurs de chaque variable** ci-dessus. (2 pts)
- b. Quel est le premier nombre écrit par cet algorithme ? (1 pts)
- c. Quel est le dernier nombre écrit par cet algorithme ? (1 pts)
- d. Entourez la boucle dans cet algorithme. (1 pts)
- e. Combien de fois se répète-t-elle ? (1 pts)
- f. A quoi correspond la valeur de R à la fin ? (1 pts)

Ex 9.**(8 pts)**

Pour cet exercice, on considère le programme suivant:

```
1 for i in range(2):  
2     print("Quoi?")  
3     print("Feur...")  
4 print("Fin!")
```

- a. A quoi servent les deux points (:) à la fin de la ligne 1 ainsi que les indentations (espaces vides à gauche) des lignes 2 et 3? (2 pts)
- b. Quelles valeurs sont affectées à i pendant l'exécution du programme ? (2 pts)
- c. Qu'est-ce qu'affiche ce programme exactement ? (2 pts)
- d. Qu'est-ce qu'afficherait ce programme si la ligne 3 n'était pas indentée ? (1 pts)
- e. Et si la ligne 2 n'était pas indentée non plus ? (1 pts)

Ex 10. Lecture de code**(4 pts)**

Pour cet exercice, on considérera le code suivant:

```
1 n = int(input('Choisis un nombre:'))  
2 for i in range(n):  
3     print('OK')
```

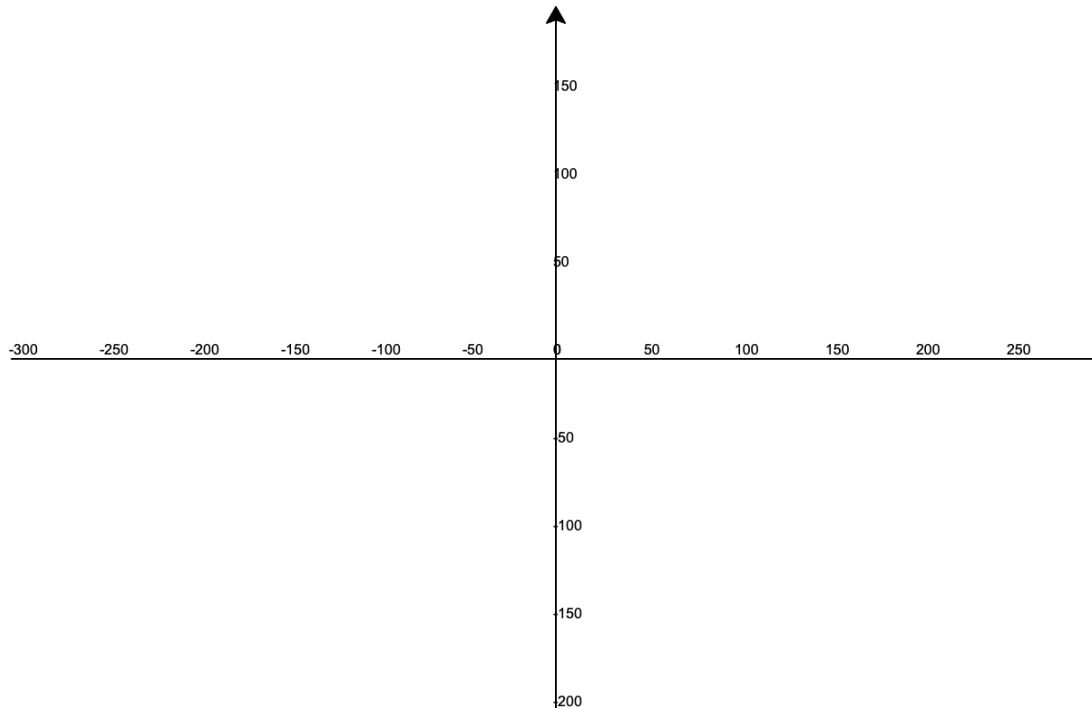
- a. Qu'est-ce qu'il affiche exactement si l'utilisateur tape **4** quand la ligne 1 s'exécute? Justifiez la réponse. (2 pts)
- b. Quelle est l'utilité de la fonction **int()** à la ligne 1? (2 pts)

Ex 11.**(5 pts)**

a. Tracez le dessin qui correspond au programme suivant. (bien visible!)

(3 pts)

```
1 from turtle import *
2
3 for i in range(3):
4
5     left(90)
6     forward(50)
7     left(90)
8     forward(50)
9     left(90)
10    forward(50)
11    left(90)
12    forward(50)
13
14    forward(100)
15
16 done()
```



b. Ci-dessous les lignes 5 à 12 du programme. Modifiez-les pour produire le même résultat en réduisant le nombre d'instructions. (Ecrivez votre version à droite)

(2 pts)

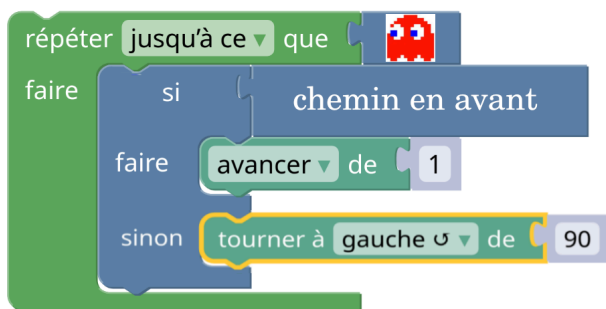
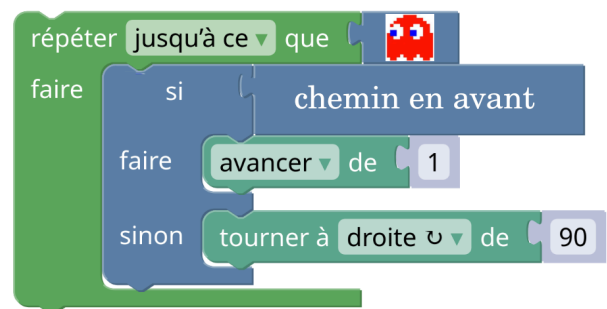
```
left(90)
forward(50)
left(90)
forward(50)
left(90)
forward(50)
left(90)
forward(50)
```

Ex 12. Algorithmes Pac-Man**(6 pts)**

Voici un labyrinthe dans lequel Pac-Man () doit rejoindre le fantôme ().
Les cases avec un **X** sont des murs infranchissables.

	X			
	X		X	
	X		X	
	X		X	
			X	

Et voici deux algorithmes candidats pour permettre à Pac-Man d'arriver jusqu'au fantôme:

Algorithme A:**Algorithme B:**

- a. Pour chaque algorithme, **dessinez** sur la grille le chemin que va parcourir Pac-Man. (4 pts)
- b. Aucun des deux algorithmes ne permet à Pac-Man d'arriver jusqu'au fantôme.
Entourez (avec la même couleur que le chemin dessiné en A) la case **X** qu'il faudrait supprimer pour que Pac-Man arrive jusqu'au fantôme avec cet algorithme. (2 pts)

FIN DE L'ÉPREUVE

Aide-mémoire informatique

1^{ère} année - 1^{er} semestre

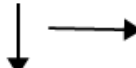
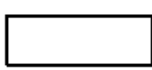
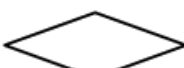
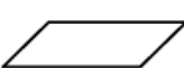
Préfixes:

Liste des préfixes très utilisés en informatique (où “o” signifie “octet”).:

To Go Mo ko o

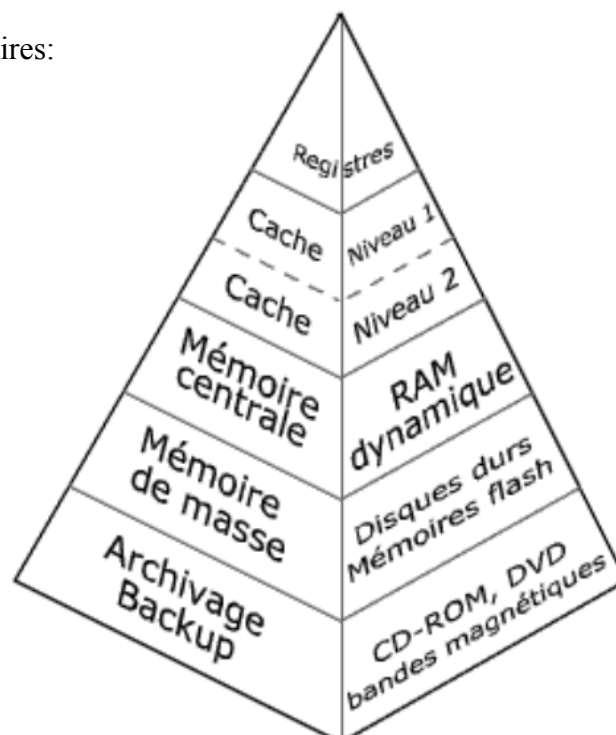
Algorithmes:

Voici les formes utilisables dans un logigramme:

Symbole	Nom	Rôle
	Flèche	Indique la séquence du programme
	Terminaison	Débute ou termine un programme
	Processus	Exécute une tâche
	Décision	Génère une décision (oui, non, etc.)
	Entrée/Sortie	Accepte une entrée ou produit une sortie

Mémoires:

Voici la pyramide des mémoires:



ASCII TABLE

Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]	48	30	110000	60	0	96	60	1100000	140	`
1	1	1	1	[START OF HEADING]	49	31	110001	61	1	97	61	1100001	141	a
2	2	10	2	[START OF TEXT]	50	32	110010	62	2	98	62	1100010	142	b
3	3	11	3	[END OF TEXT]	51	33	110011	63	3	99	63	1100011	143	c
4	4	100	4	[END OF TRANSMISSION]	52	34	110100	64	4	100	64	1100100	144	d
5	5	101	5	[ENQUIRY]	53	35	110101	65	5	101	65	1100101	145	e
6	6	110	6	[ACKNOWLEDGE]	54	36	110110	66	6	102	66	1100110	146	f
7	7	111	7	[BELL]	55	37	110111	67	7	103	67	1100111	147	g
8	8	1000	10	[BACKSPACE]	56	38	111000	70	8	104	68	1101000	150	h
9	9	1001	11	[HORIZONTAL TAB]	57	39	111001	71	9	105	69	1101001	151	i
10	A	1010	12	[LINE FEED]	58	3A	111010	72	:	106	6A	1101010	152	j
11	B	1011	13	[VERTICAL TAB]	59	3B	111011	73	;	107	6B	1101011	153	k
12	C	1100	14	[FORM FEED]	60	3C	111100	74	<	108	6C	1101100	154	l
13	D	1101	15	[CARRIAGE RETURN]	61	3D	111101	75	=	109	6D	1101101	155	m
14	E	1110	16	[SHIFT OUT]	62	3E	111110	76	>	110	6E	1101110	156	n
15	F	1111	17	[SHIFT IN]	63	3F	111111	77	?	111	6F	1101111	157	o
16	10	10000	20	[DATA LINK ESCAPE]	64	40	1000000	100	@	112	70	1110000	160	p
17	11	10001	21	[DEVICE CONTROL 1]	65	41	1000001	101	A	113	71	1110001	161	q
18	12	10010	22	[DEVICE CONTROL 2]	66	42	1000010	102	B	114	72	1110010	162	r
19	13	10011	23	[DEVICE CONTROL 3]	67	43	1000011	103	C	115	73	1110011	163	s
20	14	10100	24	[DEVICE CONTROL 4]	68	44	1000100	104	D	116	74	1110100	164	t
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]	69	45	1000101	105	E	117	75	1110101	165	u
22	16	10110	26	[SYNCHRONOUS IDLE]	70	46	1000110	106	F	118	76	1110110	166	v
23	17	10111	27	[END OF TRANS. BLOCK]	71	47	1000111	107	G	119	77	1110111	167	w
24	18	11000	30	[CANCEL]	72	48	1001000	110	H	120	78	1111000	170	x
25	19	11001	31	[END OF MEDIUM]	73	49	1001001	111	I	121	79	1111001	171	y
26	1A	11010	32	[SUBSTITUTE]	74	4A	1001010	112	J	122	7A	1111010	172	z
27	1B	11011	33	[ESCAPE]	75	4B	1001011	113	K	123	7B	1111011	173	{
28	1C	11100	34	[FILE SEPARATOR]	76	4C	1001100	114	L	124	7C	1111100	174	
29	1D	11101	35	[GROUP SEPARATOR]	77	4D	1001101	115	M	125	7D	1111101	175	}
30	1E	11110	36	[RECORD SEPARATOR]	78	4E	1001110	116	N	126	7E	1111110	176	~
31	1F	11111	37	[UNIT SEPARATOR]	79	4F	1001111	117	O	127	7F	1111111	177	[DEL]
32	20	100000	40	[SPACE]	80	50	1010000	120	P					
33	21	100001	41	!	81	51	1010001	121	Q					
34	22	100010	42	"	82	52	1010010	122	R					
35	23	100011	43	#	83	53	1010011	123	S					
36	24	100100	44	\$	84	54	1010100	124	T					
37	25	100101	45	%	85	55	1010101	125	U					
38	26	100110	46	&	86	56	1010110	126	V					
39	27	100111	47	'	87	57	1010111	127	W					
40	28	101000	50	(	88	58	1011000	130	X					
41	29	101001	51	)	89	59	1011001	131	Y					
42	2A	101010	52	*	90	5A	1011010	132	Z					
43	2B	101011	53	+	91	5B	1011011	133	[					
44	2C	101100	54	,	92	5C	1011100	134	\					
45	2D	101101	55	-	93	5D	1011101	135	]					
46	2E	101110	56	.	94	5E	1011110	136	^					
47	2F	101111	57	/	95	5F	1011111	137	_					

Programmation (Python)

Calculs :

opérations de base: + - * / **

reste et quotient de division euclidienne : % //

Rappel : $x \% y$ vaut 0 si x est divisible par y

exemples : $5 ** 2 \rightarrow 25$ $5 // 2 \rightarrow 2$ $5 \% 2 \rightarrow 1$

Variables et affectations :

types: int, float, str, bool, list ...

int(), float() pour **convertir** en int ou en float

affectation simple : `ma_variable = -5`

permutation de variables: `a, b = b, a`

affectations combinées: `+=`, `-=`, `*=`, etc.

exemple : `x += 1` est comme `x = x + 1`

Listes :

liste: `L = [-1, 3.2, "abc"]`

liste vide: `[]`

multiplication: `[0]*3 -> [0, 0, 0]`

définir un élément d'une liste :

`L[1] = 5` , `L[1]` doit déjà exister,
l'indice commence à 0.

ajouter un élément à une liste :

`L.append(-2.5)`

supprimer élément à la **position i** :

`del L[i]` ou `L.pop(i)`

supprimer élément de **valeur v** dans `L` :

`L.remove(v)`

supprimer le dernier élément dans `L` :

`L.remove()` ou `L.pop()`

longueur d'une liste `L` :

`len(L)`

somme des valeurs d'une liste:

`sum(liste)`

slice: `L[i:j]` (de i inclus à j exclu)

`L[i:]` (de i inclus à la fin)

`L[:j]` (du début à j exclu)

Chaînes de caractères :

chaîne: `txt="bonjour"` ou `txt='bonjour'`

concaténation: `txt + " est un texte"`

élément, slice d'une chaîne: `texte[i]`

comme listes mais non modifiable

Entrées/sorties :

sortie: `print(valeur, ..., end='\n')`

valeur de type quelconque, end optionnel

entrée: `texte = input("message")`

Attention: retourne toujours un str

Commentaires :

sur une ligne : `# texte`

plusieurs lignes : `"""texte"""` ou `"""texte"""`

Structures de contrôle :

1^{ère} ligne finit par `:` puis indentation des lignes du bloc

tests simples: `==`, `!=`, `<`, `<=`, `>`, `>=`, `in`

tests combinés: `and`, `or`, `not`, `a < b < c`

conditionnelle

if `condition1`:

 instructions

elif `condition2`:

 # elif = contraction de else if

 instructions

else:

 instructions

boucle while (boucle non bornée) :

while `condition`:

 instructions

boucle for (boucle bornée) :

for `i in liste`: # boucle parcourt la liste

 instructions

for `i in range(5)`: # pour i de 0 à 4

 instructions

`range(i,j,k)`: de i (inclus) à j (exclu) par pas de k

sauter, quitter une boucle : `continue`, `break`

Fonctions:

définition de la fonction - exemple d'une fonction `f` à deux paramètres :

def `f(x, y)`:

`reponse = x**2 + y**2`

`return reponse`

appel de la fonction- exemples avec la fonction `f` :

`variable = f(3, 5)`

`print( f(2, 6) )`

etc.