

ÉPREUVE REGROUPEE

Discipline: INFORMATIQUE

Date: 20/12/2024

Cours: 1 IN DP 06 / 07 / 08

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
- 10 pages dont 3 d'aide-mémoire à la fin

Nom et prénom:

Total: 45 points

Ex 1. Représentation de l'information

(7 pts)

a. Faire les conversions suivantes :

(4 pts)

- (2 pts) Du décimal₍₁₀₎ en binaire₍₂₎ : (justifier avec calcul, tableau, etc.)

$126_{(10)} = \dots\dots\dots_{(2)}$ justification:

- (2 pts) Du binaire₍₂₎ en décimal₍₁₀₎ : (justifier avec calcul, tableau, etc.)

$11110001_{(2)} = \dots\dots\dots_{(10)}$ justification:

b. Coder le mot suivant en binaire à l'aide de la table ASCII

(2 pts)

H E L L O

.....

c. Si un *mot* correspond à 32 bits, à combien d'octets correspond-il ? (1 pt)

Ex 2. Combien de bits faut-il pour encoder les douze mois de l'année ?

(2 pts)

Justifier la réponse avec un calcul.

Ex 3. Nommer deux personnages importants dans l'histoire des ordinateurs et indiquer brièvement pourquoi.

(4 pts)

i. (2 pts) personnage 1:

ii. (2 pts) personnage 2:

Ex 4. Nommer deux systèmes d'exploitation pour **ordinateurs** et deux systèmes d'exploitation pour **smartphones**.

(2 pts)

a. ordinateur:

i.

ii.

b. smartphone:

i.

ii.

Ex 5. Quelle est l'unité usuelle pour la fréquence de l'horloge qui cadence le processeur?

(1 pts)

- ☐ pixels
- ☐ bytes
- ☐ W
- ☐ GHz
- ☐ Go

Ex 6. Quelles sont les caractéristiques de la mémoire RAM ?

(3 pts)

Cocher **les** affirmations correctes: (chaque réponse juste = 1pt, chaque réponse fausse = -1pt)

- ☐ volatile (les données disparaissent en cas de coupure de courant)
- ☐ plus rapide que les disques durs
- ☐ plus lente que la mémoire cache
- ☐ Taille de l'ordre de quelques To
- ☐ optionnelle (dans un ordinateur, la mémoire RAM n'est pas indispensable)

Ex 7. Stockage de données

(3 pts)

Alice veut stocker de la musique sur son Smartphone.

La taille de chaque chanson fait 200 Mo en moyenne.

Combien de morceaux pourra-t-elle enregistrer sur un espace mémoire de 10 Go ?

justifier avec un calcul!

Ex 8. Comparer des ordinateurs (Lenovo vs. Asus):

(3 pts)

Lenovo LOQ 15IAX9 (83GS00F2FR)	Asus TUF Gaming F17 (2023) TUF707VI-HX064W
 Prix de lancement en nov. 2024 : 899€	 Prix de lancement en nov. 2024 : 1 899€
 En stock	 Prochainement disponible
899,99€	1 839,90€
	
	
• Écran 15.6" LCD - IPS (anti-reflets) 144 Hz • Résolution Full HD 1920 x 1080 pixels (16/9) • Windows 11 Famille • Intel Core i5-12450HX (8 coeurs x 4.40 GHz) • Nvidia GeForce RTX 3050 (6 Go) Version : 1740 MHz / 95 W • 24 Go DDR5 - 4800 MHz • 1 To (SSD M.2 PCI-e 4.0) • Wi-Fi 6 • Bluetooth 5.2	• Écran 17.3" LCD - IPS (anti-reflets) 144 Hz • Résolution Full HD 1920 x 1080 pixels (16/9) • Windows 11 Famille • Intel Core i7-13620H (10 coeurs x 4.90 GHz) • Nvidia GeForce RTX 4070 (Mobile) (8 Go) Version : 2225 MHz / 140 W • 16 Go DDR5 - 4800 MHz • 1 To (SSD M.2 PCI-e 4.0) • Wi-Fi 6 • Bluetooth 5.2

1. (1 pts) Je voudrais acheter un ordinateur portable pour pouvoir travailler avec le meilleur processeur possible. Quel serait mon meilleur choix et pourquoi ?
2. (1 pts) J'aimerais acheter un ordinateur pour jouer, quel ordinateur devrais-je choisir pour obtenir les meilleures performances graphiques ?
3. (1 pts) J'aimerais acheter un ordinateur doté d'une capacité de mémoire vive maximale. Quel serait mon meilleur choix et pourquoi ?

Ex 9. Analyser le code suivant:

(3 pts)

```
# Initialisation de deux variables pour stocker
# le nombre de notes et la somme des notes
nombre = 5
somme = 0

# Boucle pour demander les notes
for i in range(nombre):
    # Demander note et convertir en nombre à virgule
    note = float(input("note:"))
    somme += note

# Calculer la moyenne
moyenne = somme / nombre

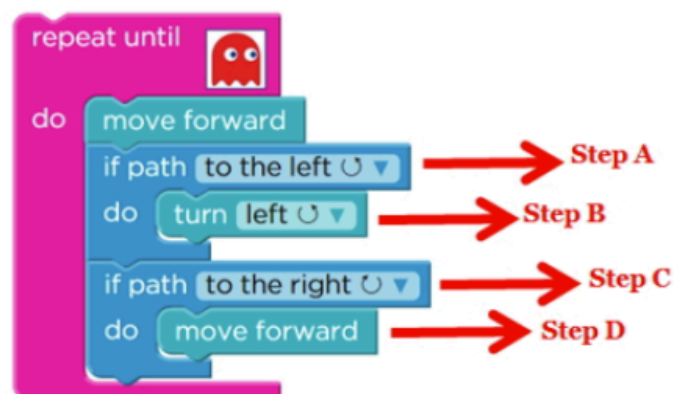
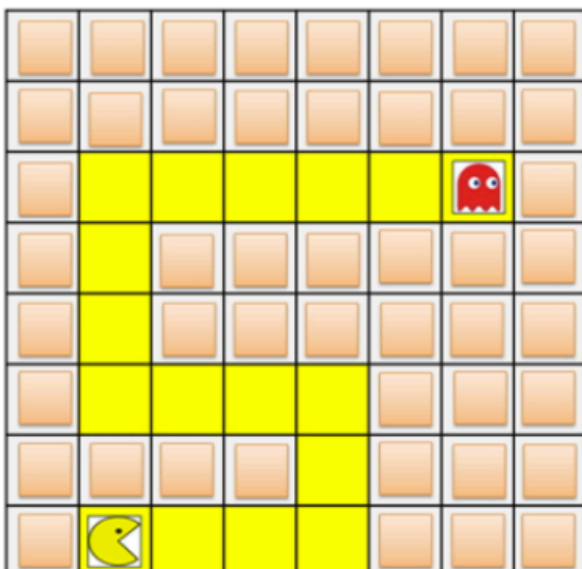
# Afficher la moyenne
print("Voici la moyenne des notes : ")
print(moyenne)
```

- a. (1 pt) Combien de notes ce programme demande-t-il à l'utilisateur?
- b. (1 pt) Pourquoi utilise-t-on la fonction `float` dans `note = float(input("note:"))` ?
- c. (1 pt) Que fait l'instruction `somme += note` ?

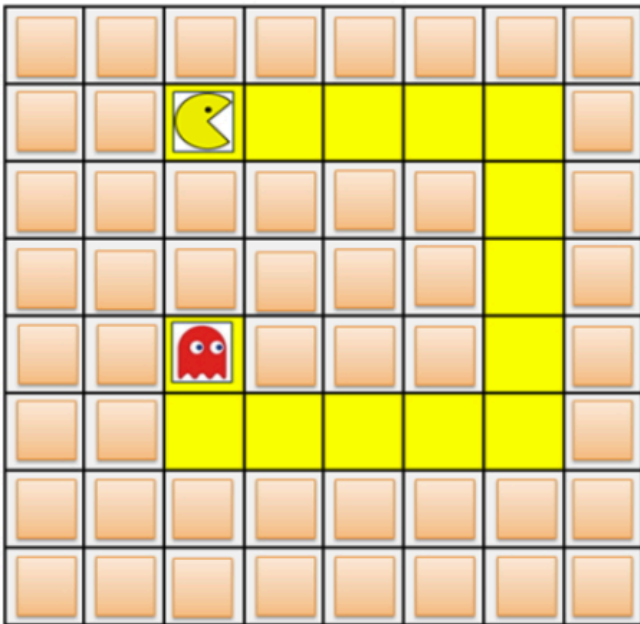
Ex 10. Algorithmes Pac-Man

(3 pts)

- a. (1 pt) Voici un programme censé permettre à Pac-Man d'atteindre le Fantôme () par le chemin jaune. Où se trouve une erreur pour y arriver (A, B, C ou D) ?



- b. (1 pt) Voici une autre situation où Pac-Man doit également atteindre le fantôme par le chemin jaune. Quelles instructions le lui permettent (A, B, C ou D) ?



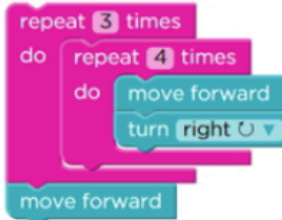
Option A



Option B



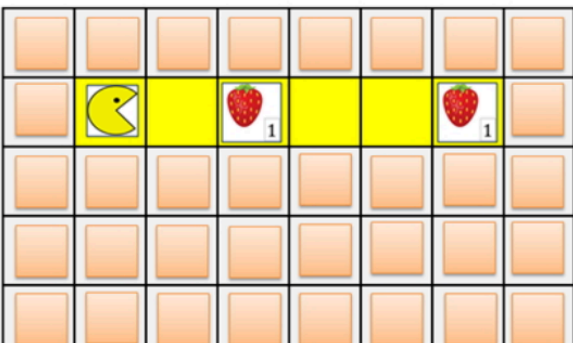
Option C



Option D



- c. (1 pt) Voici une autre situation où Pac-Man doit suivre le chemin jaune et manger toutes les fraises. Que faut-il à droite du **repeat** pour y parvenir (A, B, C ou D) ?



A. 1 fois

B. 2 fois

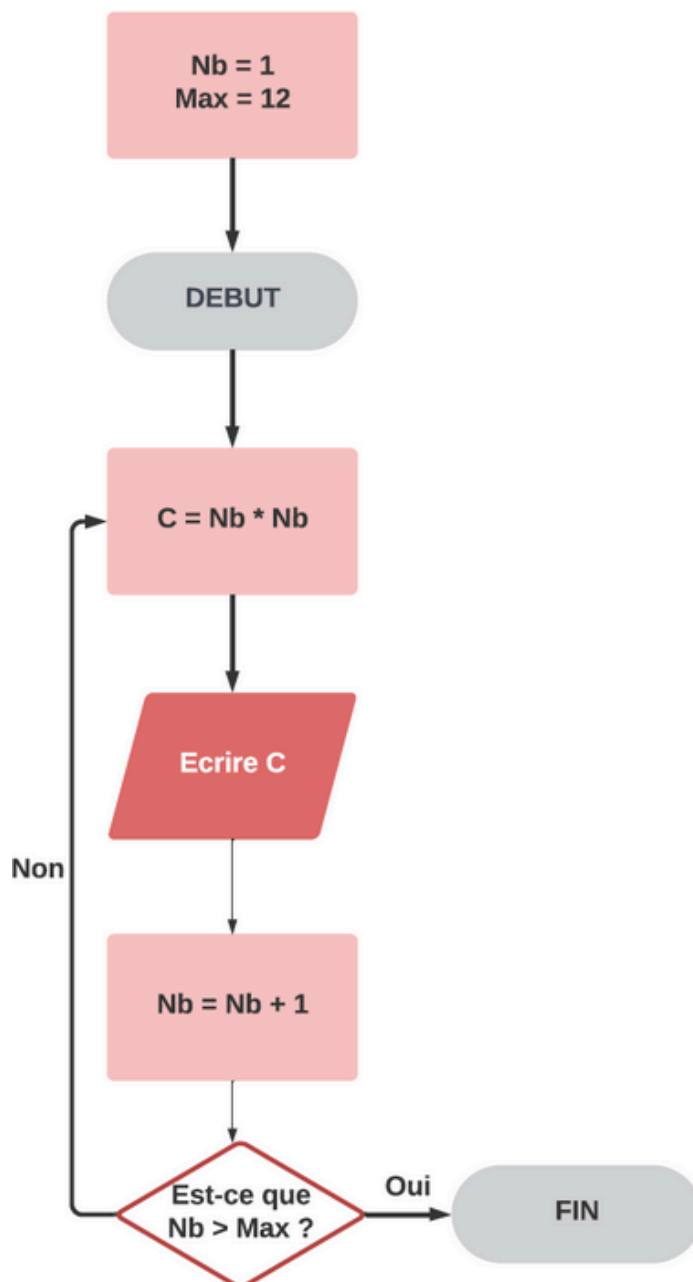
C. 3 fois

D. 5 fois

Ex 11. Analyse d'algorithme sous forme de logigramme

(6 pts)

Voici un algorithme sous forme de logigramme:



- (1 pt) Quel est le premier nombre que cet algorithme écrit ?
- (1 pt) Quel est le dernier nombre que cet algorithme écrit ?
- (1 pt) Entourer la boucle dans le logigramme en couleur
- (1 pt) Combien de fois se répète-t-elle?
- (2 pt) Décrire ce que fait cet algorithme ?

Ex 12. Analyse d'algorithme sous forme de pseudocode

(8 pts)

Voici un algorithme sous forme de pseudocode:

INITIALISATION:

$Nb \leftarrow 6$

$R \leftarrow 1$

DEBUT:

$R \leftarrow R * Nb$

SI $Nb > 1$:

$Nb \leftarrow Nb - 1$

SAUTER_VERS DEBUT

SINON:

ECRIRE R

FIN

- (2 pts) Quel nombre cet algorithme écrit-il ? (poser un calcul)
- (2 pts) Qu'est-ce qui change dans le résultat de l'algorithme si on modifie l'instruction **SI $Nb > 1$** par **SI $Nb > 0$** ? (justifier)
- (2 pts) Entourer la boucle dans l'algorithme et dire combien de fois elle se répète:
- (2 pt) Décrire ce que fait cet algorithme ?

FIN DE L'ÉPREUVE

Aide-mémoire informatique

1^{ère} année - 1^{er} semestre

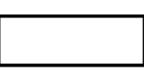
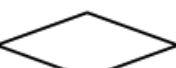
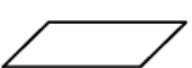
Préfixes:

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

To Go Mo ko o

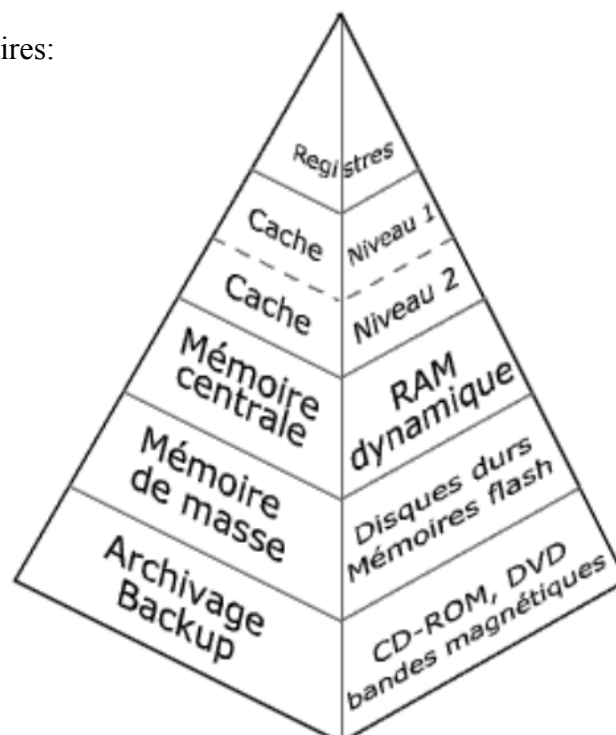
Algorithmes:

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)

Remarque: Certaines instructions n'ont peut-être pas encore été vues en classe

Calculs :

opérations de base: + - * / **

reste et quotient de division euclidienne: % //

var % div vaut 0 si var est divisible par div

ex: 5 // 2 = 2 5 % 2 = 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.

ex: x += 1 est comme x = x + 1

Listes :

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

liste vide: []

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

élément d'une liste: L[1] = 5

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)

longueur d'une liste L :

len(L)

somme des valeurs d'une liste:

sum(liste)

slice: L[i:j] (i inclus ! 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 f à deux paramètres:

def f(x, y):

reponse = x**2 + y**2

return reponse

appel de la fonction: f()

variable = f(3, 5)

print(f(2, 6))

etc.