Leçon II.1 et II.2 – Examen final 2018 1.3

Des séquences de cinq niveaux d'alerte météo doivent être transmises codées (code sans-préfixe et sans perte) sous forme de séquences de pastilles (ronds) rouges ou vertes. La table ci-dessous représente trois propositions de codes possibles. Malheureusement, ce sujet est tiré en noir et blanc ; la couleur verte ou rouge s'est donc perdue...

code I	code II	code III
niveau 1 : ●	niveau 1 : ●●	niveau 1 : ●●●
niveau 2 : ●●●●	niveau 2 : ●●●	niveau 2 : ●
niveau 3 : ●●	niveau 3 : ●●●	niveau 3 : ●●●
niveau 4 : ●●●●	niveau 4 : ●●	niveau 4 : ●
niveau 5 : ●●	niveau 5 : ●●	niveau 5 : ●●

Quel(s) code(s) (d'origine, avec les couleurs) êtes vous néanmoins sûr(e) de ne pas pouvoir utiliser pour la transmission désirée ?
Justifiez votre réponse.

Leçon II.4 – Examen final 2018 1.2

À partir d'un alphabet de 33 lettres, on compose un mot X de 128 lettres de long ; chacune des lettres de l'alphabet étant présente au moins une fois dans le mot X.
Le code de Huffman de ce mot a une longueur moyenne de 5.5 bits.

1. Est-ce possible ? **Justifiez** votre réponse.

2. Si **oui**, donnez les *meilleures* bornes (haute et basse) que vous pouvez pour
   2.1 l'entropie de ce mot :

$$\leq H(X) \leq$$

   2.2 la longueur moyenne d'un code de Shannon-Fano de ce mot :

$$\leq L_c(\text{Shannon-Fano}(X)) \leq$$

   et si c'est **non**, donnez les *meilleures* bornes (haute et basse) que vous pouvez pour la longueur moyenne d'un code de Huffman de ce mot :

$$\leq L_c(\text{Huffman}(X)) \leq$$

Leçon III.1 (Architecture des ordinateurs) – Étude de cas

Considérez le code assembleur suivant :

```
 1: charge   r2,  0
 2: charge   r3,  0
 3: charge   r4, r3
 4: somme    r3, r3, 1
 5: somme    r4, r4, 1
 6: cont_ppe r1, r4, 9
 7: somme    r2, r2, r4
 8: continue 5
 9: somme    r2, r2, r3
10: cont_pp  r3, r0, 3
```

où l'instruction « `cont_ppe a, b, N` » effectue le test « a≤b »
et l'instruction « `cont_pp a, b, N` » effectue le test « a<b ».
Lequel de ces algorithmes correspond au code ci-dessus
(avec *n* chargé dans `r0` et *m* dans `r1`) :

A.

algoA

entrée : *deux entiers* $1 \leq n < m$
sortie : S

```
S ⟵ 0
i ⟵ 0
Tant que i < n
    j ⟵ i
    Tant que j < m
        Si j < m
            S ⟵ S + j
        Sinon
            S ⟵ S + i
Sortir : S
```

B.

algoB

entrée : *deux entiers* $1 \leq n < m$
sortie : S

```
S ⟵ 0
Pour i de n à 1 en descendant (de 1 en 1)
    Pour j de m − 1 à i en descendant (de 1 en 1)
        S ⟵ S + i
    S ⟵ S + j
Sortir : S
```

C.

algoC

entrée : *deux entiers* $1 \leq n < m$

sortie : S

$S \longleftarrow 0$

Pour i de 1 à $n-1$

  Pour j de 1 à m

    $S \longleftarrow S + j$

  $S \longleftarrow S + i$

Sortir : S

D.

algoD

entrée : *deux entiers* $1 \leq n < m$

sortie : $\frac{n}{2}(m(m-1) + \frac{1}{6}(n+1)(8-2n))$

Si $n = 1$

  Sortir : $1 + \frac{(m-1) \cdot m}{2}$

$S \longleftarrow n$

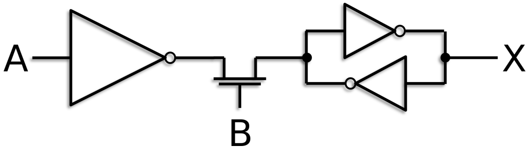
Pour i de $m-1$ à n en descendant (de 1 en 1)

  $S \longleftarrow S + i$

Sortir : $S + \textbf{algoD}(n-1, m)$

Leçon III.1 (Architecture des ordinateurs) – Étude de cas

On considère le système logique suivant :



auquel on soumet les entrées A et B suivantes à quatre instants consécutifs :

t	A	B	X
1	0	1	x_1
2	1	0	x_2
3	0	1	x_3
4	1	1	x_4

Quelles sont les quatre sorties (x_1, x_2, x_3, x_4) correspondantes ?

Leçon III.1 (Architecture des ordinateurs) – Étude de cas

Quelle est la table de vérité du programme ci-contre (sachant que `r1` et `r2` contiennent soit 0 soit 1) ?

A]

r1	r2	r3
0	0	0
1	0	0
0	1	0
1	1	10

B]

r1	r2	r3
0	0	0
1	0	1
0	1	1
1	1	1

C]

r1	r2	r3
0	0	0
1	0	1
0	1	1
1	1	0

D] Aucune des trois.

```
1: cont_egal r1, 0, 5
2: cont_egal r2, 0, 5
3: charge r3, 0
4: continue 6
5: somme r3 r1 r2
6: // fin (stop)
```