Épreuve blanche de type 2
26 janvier 2021
Durée 5 heures

Sujet proposé par L. Gonnord, R. Ledru et N. Louvet

Aucun document autorisé à part les programmes SNT et ISN (Première et Terminale)

Instructions **à lire attentivement!**

1. **Merci de traiter les parties 1 et 2 sur une (même) copie séparée**
2. Les annexes sont à partir de la page 13.
3. Les réponses aux questions seront numérotées, et on évitera le grappillage.
4. Les codes écrits seront expliqués (au besoin avec des exemples), *commentés*, indentés correctement. Lorsque la signature d'un algorithme n'est pas fournie, vous veillerez à bien justifier vos choix.
5. Tiers temps (100 min = 1heure 40 de plus).

1 Séquences pédagogiques “IHM” (SNT/NSI) 1h10

Sujet proposé par R. Ledru. Des réponses courtes mais justifiées de manière précises sont attendues.

Vous êtes nouvellement nommé·e sur un poste de certifié·e NSI en lycée. Vous avez 3 classes de Secondes SNT (35 élèves), 1 classe de Première NSI (24 élèves) et 1 classe de Terminale NSI (18 élèves). Votre emploi du temps est décrit à la figure 1. Les groupes, les horaires, ainsi que les salles sont indiqués. A partir de ce tableau, vous allez proposer diverses constructions pédagogiques pour des classes de Seconde et Première sur le thème de l'IHM sur le web.

	Semaine A			Semaine B		
	2nd	1iere	Terminale	2nd	1iere	Terminale
Lundi	1h 206	2h 1Nsi	2h T Nsi	1h206 1h208 G1 Salle info 1h208 G2 Salle info	2h 1Nsi	2h T Nsi
Mardi						
Mercredi			2h T Nsi Salle info			2h T Nsi
Jeudi	1h 207	2h 1 Nsi Salle info		1h207	2h 1 Nsi	
Vendredi	1h 208		2h T Nsi Salle info	1h208 1h206 G1 1h206 G2 1h207 G2 Salle info 1h207 G1 Salle info		2h T Nsi

FIGURE 1 – Votre emploi du temps

1.1 Fonctionnement pédagogique global

Question 1

À partir de cet emploi du temps, quelle(s) modification(s) devez-vous demander à l'administration du lycée dès la rentrée afin « d'équilibrer » vos enseignements ?

Question 2

L'arrêté du 16 juillet 2018 relatif à l'organisation et aux volumes horaires de la classe de seconde des lycées d'enseignement général et technologique et des lycées d'enseignement général et technologique agricole préconise une durée d'1h30 par semaine par élève de l'enseignement de Sciences Numériques et Technologique. Cet arrêté est-il respecté ?

Question 3

D'après cet emploi du temps, réalisez-vous des heures supplémentaires ?

Question 4

Est-il possible que le nombre d'élèves de Terminale NSI soit supérieur au nombre d'élèves de Première NSI ?

1.2 L'IHM en SNT

Le programme de SNT propose 7 thèmes à aborder avec les élèves.

Question 5

En rappelant le thème concerné proposé en seconde, proposez la progression, incluse dans l'année, de votre séquence pédagogique sur l'IHM. Vous proposerez un découpage de possible en séances de votre séquence. *max 10 lignes*

Question 6

Vous préciserez le résumé (*5 lignes max*), la durée (temps élève) et le mode d'évaluation de la séance dans laquelle l'interaction client-serveur est abordé.

1.3 L'IHM en NSI

Le programme de spécialité Première NSI précise les notions à aborder sur l'IHM durant l'année de première. Or, à la fin de la séquence sur l'IHM réalisée en janvier et qui a duré quatre semaines, un QCM d'évaluation sommative est proposé aux élèves. Les 5 premières questions vous sont données en annexe.

Question 7

En analysant les questions données aux élèves, repérer les capacités qui ne sont pas évaluées (ou qui vous semblent insuffisantes).

Question 8

Parmi les capacités repérées dans la question 1, proposer une question par capacité, type QCM qui permet de vérifier la compétence de l'élève. Justifier votre évaluation.

Question 9

Les résultats obtenus de ce QCM montre que 6 élèves ont acquis les notions vues et 3 qui n'ont pas acquis. Expliquer brièvement votre réaction pédagogique lors de la séance de 2h prévu le lundi en semaine A.

Question 10

Dans votre progression globale, quelle séquence proposeriez-vous après L'IHM en Première NSI? Détaillez les pré-requis si nécessaires.

2 Un test d'auto-diagnostic à l'entrée de Seconde (SNT) - 20 min

Dans l'annexe 4.2, on fournit un test d'auto diagnostic en seconde, à réaliser en début d'année. La grille "aide de correction" est fournie aussi, à la fin de la dernière page.

Question 1

Quels sont les éléments du programme de seconde SNT diagnostiqués dans ce test ? Pour quel niveau de maîtrise ? *3-4 lignes*

Question 2

Proposer un exercice de remédiation pour la conditionnelle (sans utiliser le langage Python). *10 à 15 lignes d'explication en plus de l'exercice, dont vous fournirez un corrigé.*

3 Concours de programmation et activités débranchées (1 heure)

Dans cette section on s'intéresse à quelques exercices proposés par des concours de programmation à destination d'élèves collégiens/lycéens, autour de parcours de robots, ou tortues.

3.1 Une question du concours Castor

Le concours Castor s'adresse aux jeunes algorithmicien·ne·s. La Figure 2 propose un exercice pris de la page web d'entraînement au concours [1].

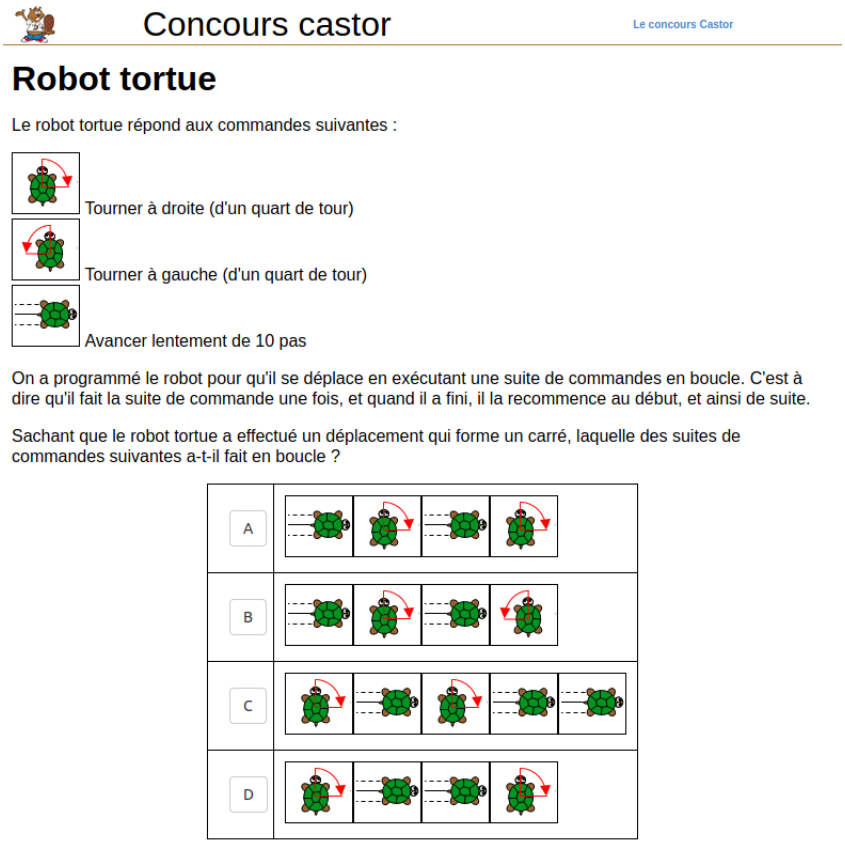


FIGURE 2 – Une question du concours Castor

Question 1

Donner une correction enseignante rédigée.

Question 2

Commenter le choix d'une boucle infinie pour cet exercice *3 lignes*

Question 3

Proposer une adaptation de cet exercice qui n'utilise pas la notion de boucle infinie (ni de répétition).
5 à 10 lignes, une ou deux questions, on ne demande pas de rédiger une correction.

1. http://castor-informatique.fr/

3.2 Une question du concours AlgoRea

Pour ce concours de programmation [2], les élèves peuvent choisir entre Scratch, un autre logiciel "block diagram", ou alors Python. La Figure 3 propose un exercice pris de la page web d'entraînement au concours [3]. Il s'agit du 4ième exercice de la série, dans lequel les tests sont autorisés.

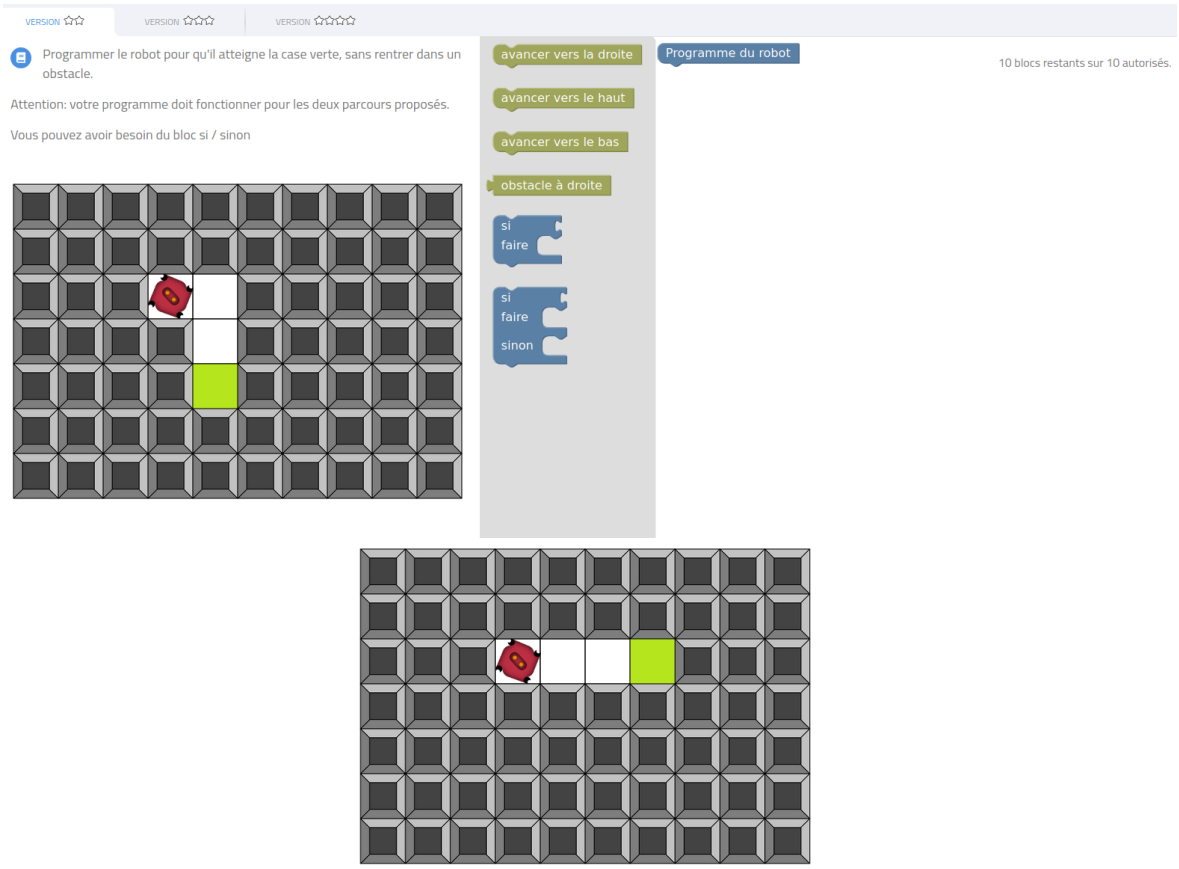


FIGURE 3 – Une question du concours Rea : on montre l'interface et les deux parcours à réaliser. Attention le robot doit s'arrêter sur la case verte, et il n'y a pas de boucle infinie implicite

Question 4

Proposer une consigne plus claire de niveau collège, qui remplacerait les consignes officielles et les précisions de la conceptrice de l'énoncé *3/4 lignes*

Question 5

Proposer une correction de l'exercice à destination d'élèves de niveau collège.

Question 6

Proposer un exercice du même style en utilisant la nouvelle brique `répéter xxx fois faire ....`, et deux parcours. *Expliquez comment vous choisissez vos deux parcours.*

4 Chiffrement, exponentiation (2 heures 30)

4.1 Utilisation d'un tableur pour calculer une fonction

Notations.
$\mathbb{N}$ désigne l'ensemble des entiers naturels.
Pour m et n deux entiers naturels, $[\![m, n]\!]$ désigne l'ensemble des entiers naturels k tels que $m \leqslant k \leqslant n$.

On souhaite crypter des messages, lettre à lettre. Pour écrire ces messages, on utilise 29 caractères différents : les 26 lettres de l'alphabet et les trois symboles espace, virgule et point. Pour faciliter le travail de cryptage, on code chacun de ces 29 caractères par un entier :

.	␣	A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

N	O	P	Q	R	S	T	U	V	W	X	Y	Z	,
15	16	17	18	19	20	21	22	23	24	25	26	27	28

On note R l'ensemble des entiers utilisés dans ce cryptage, c'est-à-dire l'ensemble $[\![0, 28]\!]$. Pour tout entier naturel k non nul, on note f_k l'application de R dans R qui à tout x de R associe le reste de la division euclidienne de x^k par 29.

Ces fonctions f_k, appelées *fonctions de cryptage*, sont utilisées pour crypter des messages.

Partie A : premiers essais

Albert souhaite utiliser comme fonction de cryptage l'application f_3. Benoît propose d'utiliser f_7. Camille choisit d'utiliser f_{19}.

I. Que devient la lettre E par la méthode de cryptage d'Albert ?

II. Montrer que, quelle que soit la fonction de cryptagef_k choisie, les symboles espace et point sont inchangés.

III. Un élève de troisième propose d'utiliser un tableur pour calculer les valeurs de f_k. Il prépare la feuille de calcul suivante :

	A	B	C	D	E	...	AC	AD	AE
1		.	␣	A	B	...	Z	,	
2	x	0	1	2	3	...	27	28	Exposant k
3	$f_k(x)$	0	1						3

Dans la cellule D3, il entre la formule =MOD(D2^AE3 ;29). Comment modifier cette formule afin de pouvoir la dupliquer en utilisant la poignée de recopie, sachant que le tableau doit rester correct lorsque le contenu de la cellule AE3 est modifié ?

On rappelle que MOD(a ;b) renvoie le reste de la division euclidienne de a par b.

IV. Benoît utilise la feuille de calcul précédente pour son cryptage avec f_7. Il obtient le tableau suivant :

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$f_k(x)$	0	1	12	12	28	28	28	1	17	28	17	12	17	28	12

x	15	16	17	18	19	20	21	22	23	24	25	26	27	28
$f_k(x)$	17	1	12	17	12	1	12	28	1	1	1	17	17	28

Crypter les mots CLE et LUC. Que constate t-on ?

V. Quelle propriété doit vérifier la fonction f_k pour assurer le décryptage ?

VI. Camille utilise la feuille de calcul de la question III avec k=19. Dans les cellules allant de E3 à AD3, il s'affiche `#NOMBRE!` [4]. Comment expliquer ce résultat ? On verra dans la partie B comment contourner ce problème.

Partie B : différents procédés de calcul de f_{19}

On décrit dans cette partie trois méthodes pour calculer f_{19} à l'aide d'un tableur.

VII - Première méthode On souhaite compléter la feuille de calcul suivante :

	A	B	C	D	E	...	AC	AD
1		.		A	B	...	Z	,
2	x	0	1	2	3	...	27	28
3	$f_2(x)$	0	1					
4	$f_3(x)$	0	1					
⋮	⋮	⋮	⋮					
20	$f_{19}(x)$	0	1					

1. Quelle formule doit-on écrire en D3 pour remplir le tableau en utilisant la poignée de recopie ?
2. Pour remplir chaque colonne, combien de multiplications et combien de divisions euclidiennes par 29 sont-elles effectuées ?

VIII - Seconde méthode On souhaite compléter la feuille de calcul suivante (on rappelle que $f \circ g(x) = f(g(x))$) :

	A	B	C	D	E	...	AC	AD
1		.		A	B	...	Z	,
2	x	0	1	2	3	...	27	28
3	$f_2(x)$	0	1					
4	$(f_2 \circ f_2)(x)$	0	1					
5	$(f_2 \circ f_2 \circ f_2)(x)$	0	1					
6	$(f_2 \circ f_2 \circ f_2 \circ f_2)(x)$	0	1					
7								

1. Quelle formule doit-on écrire en D3 pour remplir le tableau en utilisant la poignée de recopie ?
2. En constatant que $19 = 2^4 + 2^1 + 2^0$, montrer que pour tout $x \in \mathbb{R}$:

$$f_{19}(x) \equiv (f_2 \circ f_2 \circ f_2 \circ f_2)(x) \times f_2(x) \times x[29]$$

On pourra utiliser le fait que "le modulo commute avec la multiplication" (en termes savants, calculer dans Z/29Z))

4. Sous LibreOffice Calc, il ne s'affiche rien.

3. Quelle formule doit-on écrire en D7 pour remplir la ligne 7 pour obtenir f_{19} ? Il y avait un souci d'énoncé ici...
4. Pour remplir chaque colonne, combien de multiplications et combien de divisions euclidiennes par 29 sont-elles effectuées ?

IX - Troisième méthode On souhaite compléter la feuille de calcul suivante :

	A	B	C	D	E	...	AC	AD
1		.		A	B	...	Z	,
2	x	0	1	2	3	...	27	28
3	$f_3(x)$	0	1					
4	$(f_3 \circ f_3)(x)$	0	1					
5								

1. Quelle formule doit-on écrire en D3 pour remplir le tableau en utilisant la poignée de recopie ?
2. En constatant que $19 = 2 \times 3^2 + 3^0$, donner une formule permettant de calculer $f_{19}(x)$ à partir de la feuille de calcul précédente ?
3. Quelle formule doit-on écrire en D5 pour remplir la ligne 5 en utilisant la poignée de recopie et ainsi obtenir f_{19} ?
4. Pour remplir chaque colonne, combien de multiplications et combien de divisions euclidiennes par 29 sont-elles effectuées ?

Comparaison Quelle méthode vous semble la plus performante ? *On fera une comparaison de complexité "à la louche" mais avec des paramètres clairs et des arguments clairement formulés. 10 lignes max.*

4.2 Utilisation dans le cadre d'une ressource pédagogique de Terminale NSI

On désire adapter le problème précédent pour réaliser une séquence pédagogique autour de l'exponentiation rapide. On fournit en annexe la page Wikipedia fr de cet algorithme.

Question 1

L'élève Mauricette, en Terminale NSI, a vu la séquence proposée en troisième [5] par l'enseignant·e de Mathématiques de son cousin Camille. Voici sa remarque :

> En Python, il n'y a pas besoin de s'embêter à faire tout cela, on fait `(15**19)%29` dans l'évaluateur qui répond 19 et tout le monde est content!

Commenter la remarque de Mauricette dans un discours accessible en Terminale NSI. *5 lignes max*

Question 2

Rédiger en Python un algorithme naïf de calcul de puissance (non modulaire). *Ne pas oublier de spécifier correctement les entrées et leurs types, ...* Quelle est sa complexité algorithmique en nombre de multiplications?

Question 3

Donner $P(i)$ une propriété vraie à la fin de la boucle numéro i ($0 \leq i \leq n-1$) qui suffit à prouver par récurrence la correction [6] de l'algorithme précédent (on ne demande pas la rédaction de la preuve).

Question 4

On donne en annexe une documentation pour l'algorithme d'exponentiation rapide. Adapter l'algorithme récursif en une fonction récursive Python. Justifier la complexité en $O(log(n))$ donnée dans l'annexe.

Question 5

Les élèves Anaïs, Blaise et Chiara ont trouvé sur le web diverses implémentations de cet algorithme dans lesquelles le test de parité est réalisé de différentes façons :

```
if (type(n/2) == int): # v1
if (math.ceil(n/2) == n//2): # v2
if int(n/2) != n/2: # v3
```

Expliquer à ces élèves pourquoi ces versions sont inappropriées.

Question 6

Pourquoi réaliser une séquence sur l'exponentiation rapide en Terminale plutôt qu'en première?

Question 7

Commenter le rapport entre le problème précédent avec support tableur (section 4.1) et l'algorithme d'exponentiation rapide.

Question 8

Expliquer en un paragraphe maximum le calcul **itératif** de x^{18} par l'algorithme d'exponentiation rapide, accessible à des élèves de Première ou Terminale NSI. Compter le nombre d'opérations pour cette valeur de $n = 18$ particulière.

Question 9

On fournit le code suivant pour cet algorithme :

5. Oui, le programme de mathématiques de classe de troisième comporte les notions de pgcd, de modulo et la décomposition en facteurs premiers.

6. Le résultat est bien ce que l'on veut à la fin.

```
def puissance_rapide_iter(x0, n0):
""" calcul de x0^n0 par exponentiation rapide"""
r = 1
n = n0
x = x0
while (n>0):
    if n%2==1:
        r = r *x
    x = x *x
    n = n//2
return r
```

Donner une propriété $P(n)$ à la fin de la boucle n qui suffirait à montrer la correction de ce programme (invariant de boucle). Faire un nouveau calcul de complexité.

Question 10

Sans sourciller, lors d'un oral de rattrapage du BAC, Artémis affirme : "l'exposant n étant fixé, l'exponentiation rapide est l'algorithme le plus efficace, en termes de nombre de multiplications, pour calculer x^n". Proposer un exemple permettant de détromper Artémis, et commenter.

Question 11

Dans le célèbre *Art of Computer Programming*, Donald Knuth donne une explication (redonnée en annexe) de l'algorithme de l'exponentation rapide à base de décomposition en base 2. Sans traduire le texte mot à mot, donner une explication de l'algorithme d'exponentiation rapide basée sur la décomposition en base 2 de l'exposant n.

Question 12

L'élève Nicolas L. dont par pudeur on taira le nom ici produit le code suivant comme implémentation de l'algorithme d'exponentiation rapide basé sur la décomposition en base 2 de n (algorithme de Knuth en annexe) :

```
def powmod2(x, n):
  """exponentiation rapide (à la Knuth)"""
  if n == 0: return 1
  if n == 1: return x
  # On extrait n sous forme d'une chaîne de bits
  bitstring = bin(n)[2:]
  # On fabrique la chaîne des SX et X
  powstring = ""
  for b in bitstring:
    if b == '1':
      powstring = powstring + 'SX'
    else:
      powstring = powstring + 'S'
  # on vire le SX de tête
  powstring = powstring[2:]
  # on utilise la chaîne powstring pour l'exponentiation
  r = x
  for c in powstring: # On parcourt la chaîne.
    if c == 'S':
      r = (r *r)
    else:
      r = (r *x)
```

```
    return r
```

Par chance, cette implémentation est correcte, mais ce n'est pas vraiment celle que vous attendiez. Commenter le code de l'élève et donner des pistes d'amélioration claires.

Question 13

Afin de justifier aux élèves l'utilité d'améliorer la complexité algorithmique des algorithmes, on décide de leur proposer une étude expérimentale des temps d'exécution des deux algorithmes présentés plus haut (l'algorithme naïf, l'algorithme d'exponentiation rapide itératif). Un groupe d'élèves obtient les timings présentés à la Figure 4. Commenter ces résultats expérimentaux en les mettant en regard de la complexité algorithmique.

n	puissance naïve (7^n)	exponentiation rapide (7^n)
20000	0,02	0
40000	0,06	0,01
80000	0,24	0,01
160000	1,06	0,03
320000	3,84	0,09
640000	15,45	0,31
1280000	62,89	0,97
2560000	257,26	2,94
5120000	1017,28	8,78
10240000	4396,65	27,27

FIGURE 4 – Résultats expérimentaux de deux versions de calcul de puissance, temps en secondes obtenus avec `pytest`.

Question 14

Construire une séquence pédagogique "exponentiation rapide" en deux séances de 1h30. en vous inspirant des questions précédentes.*Vous détaillerez la chronologie, l'objectif de chaque séance et les questions posées aux élèves. Vous préciserez quels points du programme (compétences, thèmes) sont abordées dans votre séquence. Vous n'êtes clairement pas obligé·e·s d'utiliser toutes les ressources/points de vue de cet énoncé. 1 à 2 pages*

Nom :	Prénom :

Lycée xx	Questionnaire IHM	Première NSI

Question 1 : Parmi les affirmations suivantes, laquelle est fausse ?

- L'interactivité dans une page HTML ne peut se faire que grâce à JavaScript.
- L'interactivité dans une page HTML peut se faire avec des CSS.
- L'interactivité dans une page HTML ne concerne que les clics sur boutons.

Question 2 : Un site internet utilise une requête HTTP avec la méthode POST pour transmettre les données d'un formulaire. Laquelle des affirmations suivantes est incorrecte ?

- Les données envoyées ne sont pas visibles
- Il est possible de transmettre des données de type binaire
- Les données transmises sont cryptées
- Il n'y a pas de restriction de longueur pour les données transmises

Question 3 : Parmi GET et POST, quelle méthode d'envoi de formulaire crypte les informations envoyées au serveur ?

- Les deux : GET et POST
- GET seulement
- POST seulement
- Aucune des deux

Question 4 : Dans un formulaire sur une page web, pour transmettre des données sécurisées comme un mot de passe ou un numéro de carte bancaire, il vaut mieux utiliser la méthode :

- HEAD
- GET
- HTTPS
- POST

Question 5 : Parmi les balises HTML ci-dessous quelle est celle qui permet à l'utilisateur de saisir son nom dans un formulaire en respectant la norme HTML ?

- <select />
- <form />
- <input type='text' />
- <input type='name' />

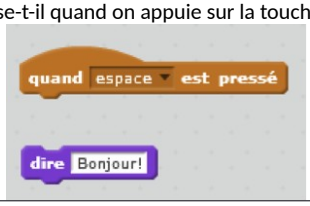
Test-diagnostic d'entrée de 2de sur Scratch

Ce test-diagnostic a été réalisé par le groupe « Apprentissage du code informatique au collège » 2016-2017 de l'IREM de Lorraine.

Il a pour objectif d'aider les professeurs de classe de 2de à déterminer le niveau de connaissance des élèves en programmation par rapport au programme de cycle 4.

Question n° 1

Que se passe-t-il quand on appuie sur la touche espace ?



A. Rien
B. Le lutin dit « Bonjour ! »
C. L'utilisateur doit dire « Bonjour ! »

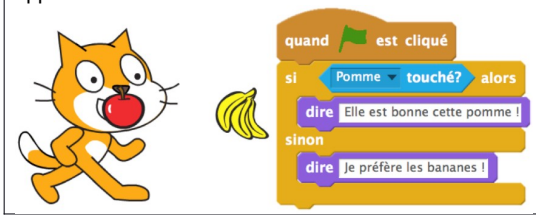
Question n° 2

Quand ce programme a fini de s'exécuter, quel message apparaît ?

A. Je suis un oiseau
B. Je suis un chien
C. Je suis un poisson.

Question n° 3

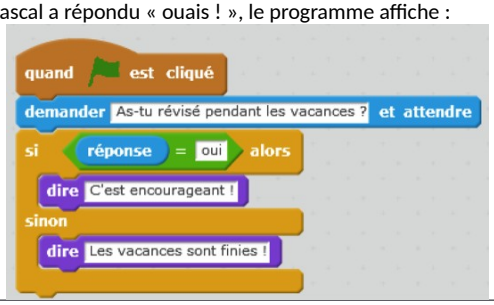
Quand ce programme a fini de s'exécuter, quel message apparaît ?



A. Je préfère les bananes.
B. Pomme touché ?
C. Elle est bonne cette pomme !
D. Rien

Question n° 4

Pascal a répondu « ouais ! », le programme affiche :



A. As-tu révisé pendant les vacances ?
B. Oui
C. C'est encourageant !
D. Les vacances sont finies !

Question n° 5

Que font ces programmes ?

A. Les deux programmes construisent un carré.
B. Le programme de gauche construit un carré, pas celui de droite.
C. Le programme de droite construit un carré, pas celui de gauche.
D. Aucun des deux programmes ne construit un carré.

Question n° 6

Dans ce programme, la boucle se répète jusqu'à ce que :

A. Le bord soit touché.
B. Parrot soit touché.
C. Duck soit touché.
D. La touche espace est pressée.

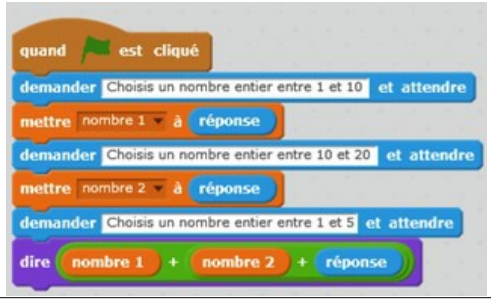
Question n° 7

Dans ce programme, quand Parrot est touché :

A. On augmente la valeur de Balles de 1.
B. On augmente la valeur de Points de 1.
C. On diminue la valeur de Balles de 1.
D. On diminue la valeur de Points de 1.

Question n° 8

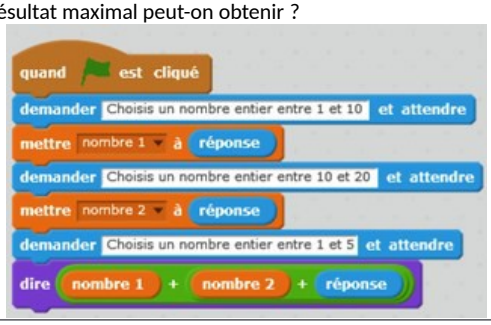
On a saisi « 8 » pour la variable « nombre 2 ». A-t-on eu raison ?



A. Oui
B. Non
C. Je ne sais pas

Question n° 9

En respectant les consignes de saisie des variables, quel résultat maximal peut-on obtenir ?



A. 205
B. 250
C. 35
D. 12

Question n° 10

Que fait ce programme ?

A. Il ajoute 5 fois 1 au nombre saisi par l'utilisateur.
B. Il multiplie 5 fois par 1 le nombre saisi par l'utilisateur.
C. Il effectue le produit de 5 facteurs égaux au nombre saisi par l'utilisateur.

Aide de correction pour les enseignants.

Réponses correctes : 1 : A – 2 : C – 3 : C – 4 : D – 5 : A – 6 : A – 7 : B – 8 : B – 9 : C – 10 : C

Éléments à retravailler en fonction de l'erreur produite :

Question 1 : les briques sont déconnectées donc il n'y a rien à exécuter quand la touche Espace est pressée ; être attentif aux éléments présents

Question 2 : les instructions s'exécutent successivement ; c'est la dernière qui contient le message qui reste à la fin

Question 3 : le lutin (chat) touche la pomme donc la condition est vraie ; être attentif aux éléments et au fonctionnement d'une conditionnelle

Question 4 : la condition est fausse (la saisie ne correspond pas au message dans le test) ; être attentif aux éléments et au fonctionnement d'une conditionnelle

Question 5 : ce sont les deux mêmes programmes, le 2e utilise simplement une boucle pour répéter la tracé, il est plus « économique »

Question 6 : bien repérer la condition de fin de l'instruction « répéter » ; être attentif aux instructions

Question 7 : bien repérer l'instruction « si parrot touché » puis ce qui est fait ; être attentif aux instructions

Question 8 : la saisie de « nombre 2 » correspond à l'instruction « demander » qui la précède ; être attentif aux consignes

Question 9 : le résultat correspond à l'addition des trois nombres saisis qui doivent être 10, 20 et 5 ; être attentif aux consignes et aux opérations effectuées

Question 10 : le résultat correspond à multiplier 1 par « réponse », ceci cinq fois de suite donc 5 facteurs égaux au nombre saisi ; être attentif à la signification du calcul dans la boucle et à l'initialisation de la variable « nombre »

Exponentiation rapide

En informatique, l'**exponentiation rapide** est un algorithme utilisé pour calculer rapidement, de grandes puissances entières. En anglais, cette méthode est aussi appelée ***square-and-multiply*** (« mettre au carré et multiplier »).

Sommaire

Écriture mathématique

La première façon de calculer une puissance n^p est de multiplier n par lui-même p fois. Cependant, il existe des méthodes bien plus efficaces, où le nombre d'opérations nécessaires n'est plus de l'ordre de p mais de l'ordre de $\log(p)$.

Par exemple, si l'on écrit $p = \sum_{i \leq d} a_i 2^i$ pour $a_i \in \{0, 1\}$, on constate que

$$n^p = n^{a_0} (n^2)^{a_1} (n^{2^2})^{a_2} \dots (n^{2^d})^{a_d}.$$

Il faut ainsi d opérations pour calculer tous les n^{2^i}, puis d opérations supplémentaires pour former le produit des $(n^{2^i})^{a_i}$. Le nombre total d'opérations est donc $2d$, qui est bien de l'ordre du logarithme de p. Cette simple remarque algébrique conduit à l'algorithme présenté dans la section suivante.

Algorithme

Soit n un entier strictement supérieur à 1, supposons que l'on sache calculer, pour chaque réel x, toutes les puissances x^k de x, pour tout k, tel que $1 \leq k < n$.

- Si n est pair alors $x^n = (x^2)^{n/2}$. Il suffit alors de calculer $y^{n/2}$ pour $y = x^2$.
- Si n est impair et $n > 1$, alors $x^n = x(x^2)^{(n-1)/2}$. Il suffit de calculer $y^{(n-1)/2}$ pour $y = x^2$ et de multiplier le résultat par x.

Cette remarque nous amène à l'algorithme récursif suivant qui calcule x^n pour un entier strictement positif n :

$$\text{puissance}(x, n) = \begin{cases} x, & \text{si } n = 1 \\ \text{puissance}(x^2, n/2), & \text{si } n \text{ est pair} \\ x \times \text{puissance}(x^2, (n-1)/2), & \text{si n est impair} \end{cases}$$

En comparant à la méthode ordinaire qui consiste à multiplier x par lui-même $n-1$ fois, cet algorithme nécessite de l'ordre de $\mathrm{O}(\log\ n)$ multiplications et ainsi accélère le calcul de x^n de façon spectaculaire pour les grands entiers.

La méthode fonctionne dans tout semi-groupe et est souvent utilisée pour calculer des puissances de matrices, et particulièrement en cryptographie, mais aussi pour calculer les puissances dans un anneau d'entiers modulo q. Elle peut être aussi utilisée pour calculer des puissances d'un élément dans un groupe, en utilisant pour les puissances négatives la règle : $\mathrm{puissance}(x, -n) = (\mathrm{puissance}(x, n))^{-1}$. C'est cette méthode que l'on applique lorsque l'on effectue la multiplication de deux nombres chiffre par chiffre en base 2 : le groupe est $(\mathbb{Z}, +)$.

Voir aussi

Articles connexes

- Suite de Fibonacci
- Exponentiation modulaire

The Art of Computer Programming, D. Knuth, chapitre 4.6 (extrait)

to recover one of its true factors from a factorization modulo p^e whenever $p^e \geq 2B^2$, by using the algorithm of exercise 4.5.3–51.

41. [*M47*] (Beauzamy, Trevisan, and Wang.) Prove or disprove: There is a constant c such that, if $f(x)$ is any integer polynomial with all coefficients $\leq B$ in absolute value, then one of its irreducible factors has coefficients bounded by cB.

4.6.3. Evaluation of Powers

In this section we shall study the interesting problem of computing x^n efficiently, given x and n, where n is a positive integer. Suppose, for example, that we need to compute x^{16}; we could simply start with x and multiply by x fifteen times. But it is possible to obtain the same answer with only four multiplications, if we repeatedly take the square of each partial result, successively forming x^2, x^4, x^8, x^{16}.

The same idea applies, in general, to any value of n, in the following way: Write n in the binary number system (suppressing zeros at the left). Then replace each "1" by the pair of letters SX, replace each "0" by S, and cross off the "SX" that now appears at the left. The result is a rule for computing x^n, if "S" is interpreted as the operation of *squaring*, and if "X" is interpreted as the operation of *multiplying by* x. For example, if $n = 23$, its binary representation is 10111; so we form the sequence SX S SX SX SX and remove the leading SX to obtain the rule SSXSXSX. This rule states that we should "square, square, multiply by x, square, multiply by x, square, and multiply by x"; in other words, we should successively compute x^2, x^4, x^5, x^{10}, x^{11}, x^{22}, x^{23}.

This binary method is easily justified by a consideration of the sequence of exponents in the calculation: If we reinterpret "S" as the operation of multiplying by 2 and "X" as the operation of adding 1, and if we start with 1 instead of x, the rule will lead to a computation of n because of the properties of the binary number system. The method is quite ancient; it appeared before 200 B.C. in Piṅgala's Hindu classic *Chandaḥ-sûtra* [see B. Datta and A. N. Singh, *History of Hindu Mathematics* **2** (Lahore: Motilal Banarsi Das, 1935), 76]. There seem to be no other references to this method outside of India during the next 1000 years, but a clear discussion of how to compute 2^n efficiently for arbitrary n was given by al-Uqlīdisī of Damascus in A.D. 952; see *The Arithmetic of al-Uqlīdisī* by A. S. Saidan (Dordrecht: D. Reidel, 1975), 341–342, where the general ideas are illustrated for $n = 51$. See also al-Bīrūnī's *Chronology of Ancient Nations*, edited and translated by E. Sachau (London: 1879), 132–136; this eleventh-century Arabic work had great influence.

The S-and-X binary method for obtaining x^n requires no temporary storage except for x and the current partial result, so it is well suited for incorporation in the hardware of a binary computer. The method can also be readily programmed; but it requires that the binary representation of n be scanned from left to right. Computer programs generally prefer to go the other way, because the available operations of division by 2 and remainder mod 2 will deduce the binary representation from right to left. Therefore the following algorithm, based on a right-to-left scan of the number, is often more convenient:

The Art of Computer Programming, D. Knuth, chapitre 4.6 (extrait)

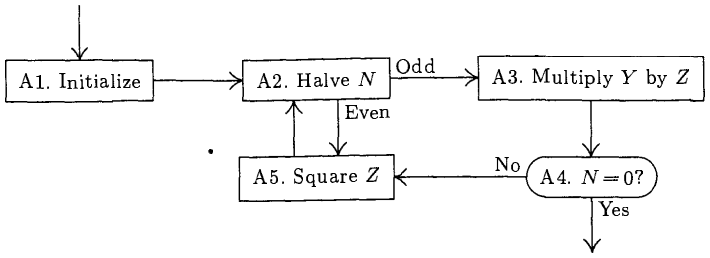


Fig. 13. Evaluation of x^n, based on a right-to-left scan of the binary notation for n.

Algorithm A (*Right-to-left binary method for exponentiation*). This algorithm evaluates x^n, where n is a positive integer. (Here x belongs to any algebraic system in which an associative multiplication, with identity element 1, has been defined.)

A1. [Initialize.] Set $N \leftarrow n$, $Y \leftarrow 1$, $Z \leftarrow x$.

A2. [Halve N.] (At this point, $x^n = Y\,Z^N$.) Set $N \leftarrow \lfloor N/2 \rfloor$, and at the same time determine whether N was even or odd. If N was even, skip to step A5.

A3. [Multiply Y by Z.] Set $Y \leftarrow Z$ times Y.

A4. [$N = 0$?] If $N = 0$, the algorithm terminates, with Y as the answer.

A5. [Square Z.] Set $Z \leftarrow Z$ times Z, and return to step A2. ▮

As an example of Algorithm A, consider the steps in the evaluation of x^{23}:

	N	Y	Z
After step A1	23	1	x
After step A5	11	x	x^2
After step A5	5	x^3	x^4
After step A5	2	x^7	x^8
After step A5	1	x^7	x^{16}
After step A4	0	x^{23}	x^{16}

A MIX program corresponding to Algorithm A appears in exercise 2.

The great calculator al-Kāshī stated Algorithm A in A.D. 1427 [*Istoriko-Mat. Issledovaniia* **7** (1954), 256–257]. The method is closely related to a procedure for multiplication that was actually used by Egyptian mathematicians as early as 2000 B.C.; for if we change step A3 to "$Y \leftarrow Y+Z$" and step A5 to "$Z \leftarrow Z+Z$", and if we set Y to zero instead of unity in step A1, the algorithm terminates with $Y = nx$. [See A. B. Chace, *The Rhind Mathematical Papyrus* (1927); W. W. Struve, *Quellen und Studien zur Geschichte der Mathematik* **A1** (1930).] This is a practical method for multiplication by hand, since it involves only the simple operations of doubling, halving, and adding. It is often called the "Russian peasant method" of multiplication, since Western visitors to Russia in the nineteenth century found the method in wide use there.