

Informatique

conformément au programme de la réforme
pour un apprentissage enrichissant

Instructions au candidat

א. Durée de l'examen : trois heures.

ב. Structure du questionnaire et barème de notation :

Ce questionnaire comporte trois sections.

Première section — Cette section comporte trois questions, répondez d'après les instructions énoncées dans cette section.

$(1 \times 10) + (1 \times 15) = 25$ points

Deuxième section — Cette section comporte trois questions, répondez à deux d'entre elles.

$(2 \times 25) = 50$ points

Troisième section — Cette section comporte des questions de quatre filières différentes.

Répondez à une question de la filière dans laquelle vous avez étudié.

$(1 \times 25) = 25$ points

Total — 100 points

ג. Matériel auxiliaire autorisé : Tout matériel auxiliaire, à l'exception d'un ordinateur programmable.

ד. Instructions particulières :

(1) Tous les programmes qu'il vous est demandé d'écrire en langage informatique dans la première section, doivent être écrits dans un seul langage – Java ou C#.

(2) Inscrivez sur la couverture de votre cahier d'examen dans quel langage vous écrivez – Java ou C#.

(3) Inscrivez sur la couverture de votre cahier d'examen le nom de la filière dans laquelle vous avez étudié.

Cette filière est l'une des quatre suivantes :

מבוא לחקר ביצועים, מערכות מחשב ואסמבלי

תכנות מונחה עצמים, מודלים חשיביים

Remarque : dans les programmes que vous écrivez, vous ne perdrez pas de points si vous écrivez une majuscule au lieu d'une minuscule ou l'inverse.

Tout ce que vous voulez écrire en tant que brouillon (plans, calculs, etc...), doit être écrit dans le cahier d'examen uniquement, sur des pages séparées. Indiquez "טיוטה" en haut de chaque page de brouillon. Écrire des notes de brouillon sur d'autres feuilles que celles du cahier d'examen peut entraîner votre disqualification !

מדעי המחשב

על פי תכנית הרפורמה ללמידה משמעותית

הוראות לנבחן

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח הערכה:

בשאלון זה שלושה פרקים.

פרק ראשון – בפרק זה שלוש שאלות, ענה על פי ההוראות בפרק.

$(10 \times 1) + (15 \times 1) = 25$ נקודות

פרק שני – בפרק זה יש שלוש שאלות,

ומהן עליך לענות על שתיים.

$(25 \times 2) = 50$ נקודות

פרק שלישי – בפרק זה שאלות בארבעה מסלולים שונים.

ענה על שאלה אחת במסלול שלמדת.

$(25 \times 1) = 25$ נקודות

סה"כ — 100 נקודות

ג. חומר עזר מותר בשימוש:

כל חומר עזר, חוץ ממחשב הניתן לתכנות.

ד. הוראות מיוחדות:

(1) את כל התכניות שאתה נדרש לכתוב בשפת מחשב

בפרק הראשון כתוב בשפה אחת בלבד –

Java או C#.

(2) רשום על הכריכה החיצונית של המחברת באיזו

שפה אתה כותב – Java או C#.

(3) רשום על הכריכה החיצונית של המחברת את שם

המסלול שלמדת.

המסלול הוא אחד מארבעת המסלולים האלה:

מערכות מחשב ואסמבלי, מבוא לחקר ביצועים,

מודלים חשיביים, תכנות מונחה עצמים.

הערה: בתכניות שאתה כותב לא יורדו לך נקודות, אם

תכתוב אות גדולה במקום אות קטנה או להפך.

כתוב במחברת הבחינה בלבד, בעמודים נפרדים, כל מה שברצונך לכתוב כטיוטה (ראשי פרקים, חישובים וכדומה). רשום "טיוטה" בראש כל עמוד טיוטה. רישום טיוטות כלשהן על דפים מחוץ למחברת הבחינה עלול לגרום לפסילת הבחינה!

Bonne chance!

בהצלחה!

Questions

Ce questionnaire comporte trois sections.

Répondez à des questions des trois sections, en suivant les instructions énoncées dans chaque section.

Première section (25 points)

Remarque : pour chaque question où une saisie est requise, il n'est pas nécessaire de vérifier la validité de la saisie.

Pour ceux qui composent en Java : pour chaque question où une saisie est requise, supposez que le programme comporte l'instruction :

Scanner input = new Scanner (System.in);

Répondez à la question 1 – obligatoire (10 points).

1. Écrivez une méthode externe **big** en Java ou **Big** en C# acceptant un tableau de nombres entiers. La méthode retournera l'indice de l'élément dont la valeur est la plus grande de ce tableau.
Si ce nombre apparaît dans le tableau plus d'une fois, la méthode retournera l'indice le plus petit parmi ces occurrences.

Répondez à l'une des questions 2-3 (15 points par question).

2. En prévision d'une compétition sportive internationale, on a défini une classe **Game** comportant trois attributs : le nom de l'épreuve – `gameName` de type chaîne ; le nombre de joueurs – `numPlayers` de type entier ; si l'épreuve est aquatique – `isWater` de type booléen.
Supposez que pour chaque attribut, les méthodes `get` et `set` en Java et les méthodes `Get` et `Set` en C#, ont été définies.
 - ⌘. Écrivez en Java ou en C#, une méthode constructeur dans la classe **Game** recevant des valeurs pour chaque attribut.
 - ⌘. Définissez une classe pays – **Country** comportant deux attributs : le nom du pays – `countryName` de type chaîne ; un tableau unidimensionnel – `games` de longueur 43 de type **Game**. Le tableau contient les épreuves auxquelles ce pays participe.
Supposez que pour chaque attribut, les méthodes `get` et `set` en Java et les méthodes `Get` et `Set` en C#, ont été définies.
 - (1) Écrivez en Java ou en C# la déclaration de la classe **Country** ainsi que ses attributs.
 - (2) Écrivez en Java ou en C# dans la classe **Country** une méthode constructeur acceptant le nom du pays. La méthode construira un objet pour un pays qui ne participe à aucune épreuve.
 - (3) Écrivez en Java ou en C# dans la classe **Country** une méthode booléenne acceptant le nom d'une épreuve et retournant `true` – si le pays participe à cette épreuve, autrement la méthode retournera `false`.

/suite en page 3/

3. Voici une portion de programme écrite en Java et en C#.

<u>Java</u>	<u>C#</u>
<pre>int i , j; for (i = 1; i <= n; i++) { a[i] = i; } for (i = 2; i < n; i++) { if (a[i] != 0) { for (j = i + 1; j <= n; j++) { if (a[j] % a[i] == 0) { a[j] = 0; } } } } for (i = 1; i <= n; i++) { if (a[i] != 0) { System.out.println(a[i]); } }</pre>	<pre>int i , j; for (i = 1; i <= n; i++) { a[i] = i; } for (i = 2; i < n; i++) { if (a[i] != 0) { for (j = i + 1; j <= n; j++) { if (a[j] % a[i] == 0) { a[j] = 0; } } } } for (i = 1; i <= n; i++) { if (a[i] != 0) { Console.WriteLine(a[i]); } }</pre>

- א. Écrivez ce qui s'affichera pour $n = 15$.
- ב. Écrivez en une phrase ce qu'effectue cette portion de programme, c'est à dire, formulez le problème dont la solution est cette portion de programme.

Deuxième section (50 points)

Attention : Dans chaque question où une implémentation est requise, vous pouvez utiliser les méthodes des classes File, Pile, Arbre binaire et Chaîne sans les implémenter. Si vous utilisez des méthodes supplémentaires, implémentez-les.

Répondez à deux des questions 4-6 (25 points par question).

4. Voici la déclaration de cinq méthodes agissant sur une structure de données quelconque.

Attention : les noms des méthodes ci-dessous ne sont pas écrits en Java ou en C#.

- insert(x) – méthode qui insère dans la structure un élément de valeur x de type entier.
- showMin() – méthode qui retourne la valeur la plus petite de la structure, sans modifier la structure.
- getMax – méthode qui retourne l'élément dont la valeur est la plus grande de la structure, puis l'extraie de la structure. S'il existe plus d'un élément de ce type, la méthode retournera puis extraira celui qui figure en premier.
- exists(x) – méthode booléenne retournant `true` si l'élément de valeur x existe dans la structure. Autrement – la méthode retourne `false`.
- div7() – méthode booléenne retournant `true` s'il existe dans la structure un élément dont la valeur est divisible par 7 sans reste. Autrement – la méthode retourne `false`.

On veut proposer des structures de données répondant à diverses exigences de complexité afin d'implémenter des méthodes parmi les cinq méthodes qu'on a définies.

Exemple : on veut proposer une structure de données sur laquelle il est possible d'exécuter les méthodes `insert`, `showMin` avec une complexité $O(1)$, et les méthodes `getMax`, `exists` avec une complexité $O(n)$. Pour cela, on définira la structure de données, puis on expliquera comment les méthodes seront implémentées.

Attention : dans cette structure, il n'est pas nécessaire de considérer la méthode `div7`.

Une structure de données adéquate est constituée :

- d'une liste chaînée bi-directionnelle, lst de type entier.
- d'un pointeur sur l'élément minimal inséré dans la liste, min .

Les méthodes seront exécutées ainsi :

Méthode	Explication sur l'implémentation	Raison pour laquelle l'implémentation répond aux exigences de complexité
insert(x)	— insertion de l'élément x en tête de la liste. — si l'élément est inférieur au minimum actuel, actualisation du pointeur min de sorte qu'il pointe sur le nouvel élément.	— insertion d'un élément en tête de la liste – $O(1)$. — si l'élément inséré est inférieur au minimum : actualisation du pointeur vers l'élément minimal – $O(1)$. — si l'élément inséré est le premier – insertion en tête de liste en $O(1)$ et actualisation du pointeur vers le premier élément qui est aussi l'élément minimal. Au final – $O(1)$
showMin()	retour de la valeur de l'élément sur lequel pointe min .	retour de la valeur – $O(1)$.
exists(x)	parcours de la liste lst et recherche de l'élément de valeur x .	dans le pire des cas – parcours de toute la liste – $O(n)$.
getMax()	parcours de la liste lst , recherche de l'élément de valeur maximale et extraction de celui-ci de la liste.	— recherche de l'élément dont la valeur est maximale – $O(n)$. — extraction de la liste – $O(1)$.

Voici deux articles 8-1. Pour chacun de ces articles, proposez une structure de données adéquate répondant aux exigences détaillées dans cet article. La structure pourra être constituée d'une combinaison de plusieurs structures et types que vous avez appris. Pour chacune des méthodes, expliquez comment vous l'implémenterez, puis expliquez pourquoi cette implémentation répond aux exigences (telles qu'elles sont exposées dans le tableau de l'exemple).

Il n'est pas nécessaire d'implémenter les méthodes.

8. Exécution des méthodes `insert` , `exists` avec une complexité $O(n)$, et exécution des méthodes `showMin` , `getMax` avec une complexité $O(1)$.

1. Exécution des méthodes `insert` , `getMax` avec une complexité $O(n)$, et exécution de la méthode `div7` avec une complexité $O(1)$. /suite en page 6/

5. Voici un programme écrit en Java et en C# .

La méthode `what` en Java et `What` en C#, accepte un tableau unidimensionnel contenant des nombres entiers, et un nombre entier k , $k > 0$. Tous les nombres du tableau sont compris entre 0 et k (inclus).

Java

```
public class Program
```

```
{  
    public static int[] what(int[] arr, int k)  
    {  
        int n = arr.length;  
        int[] b = new int[n];  
        int[] c = new int[k+1];  
  
        for (int i = 0; i < k+1; i++) c[i] = 0;  
        for (int j = 0; j < n; j++) c[arr[j]] = c[arr[j]] + 1;    /*  
  
        for (int i = 1; i < k+1 ; i++) c[i] = c[i] + c[i-1];    /**  
  
        for (int j = n-1 ; j >= 0; j--)    /****  
        {  
            b[c[arr[j]] - 1] = arr[j];  
            c[arr[j]] = c[arr[j]] - 1;  
        }  
        return b;  
    }  
  
    public static void main(String[] args)  
    {  
        int[] arr = new int[] {5,0,2,1,3,0,5};  
        arr = what(arr, 5);  
        for (int i = 0; i < arr.length; i++) System.out.println(arr[i]);  
    }  
}
```

C#

```
public class Program
```

```
{
```

```
    public static int[] What(int[] arr, int k)
```

```
    {
```

```
        int n = arr.Length;
```

```
        int[] b = new int[n];
```

```
        int[] c = new int[k+1];
```

```
        for (int i = 0; i < k+1; i++) c[i] = 0;
```

```
        for (int j = 0; j < n; j++) c[arr[j]] = c[arr[j]] + 1;    /**
```

```
        for (int i = 1; i < k+1 ; i++) c[i] = c[i] + c[i-1];    /***
```

```
        for (int j = n-1 ; j >= 0; j--)                        /****
```

```
        {
```

```
            b[c[arr[j]] - 1] = arr[j];
```

```
            c[arr[j]] = c[arr[j]] - 1;
```

```
        }
```

```
        return b;
```

```
    }
```

```
    public static void Main(string[] args)
```

```
    {
```

```
        int[] arr = new int[] {5,0,2,1,3,0,5};
```

```
        arr = What(arr, 5);
```

```
        for (int i = 0; i < arr.Length; i++) Console.WriteLine(arr[i]);
```

```
    }
```

```
}
```

⌘. (1) Donnez le tableau `c` suite à l'exécution de la boucle notée `/**` .

(2) Donnez le tableau `c` suite à l'exécution de la boucle notée `/***` .

(3) Tracez l'exécution de la boucle notée `/****` .

Au cours de la trace, montrez : `j` , `arr[j]` , `c[arr[j]]` , `b[c[arr[j]] - 1]` ,
le tableau `b` et le tableau `c` .

⌚. Qu'exécute la méthode `what` en Java ou `What` en C# ?

⌚. Quelle est la complexité d'exécution de la méthode `what` en Java ou
`What` en C# ? Justifiez votre réponse.

6. א. Implémentez une méthode externe `exist` en Java ou `Exist` en C#.
- Cette méthode acceptera un arbre binaire `t` de type entier et un nombre entier `x`. La méthode retournera `true` s'il existe dans cet arbre un nœud de valeur `x`, autrement – la méthode retournera `false`. Si l'arbre est vide – la méthode retournera `false`.
- ב. Voici la méthode `check(t1, t2)` en Java ou `Check(t1, t2)` en C#.
- Cette méthode accepte deux arbres binaires non-vides de type entier, `t1` et `t2`, puis retourne une liste contenant tous les nombres contenus dans l'arbre `t1` et qui ne sont pas contenus dans l'arbre `t2`. La méthode invite une méthode supplémentaire acceptant trois paramètres.

Java

```
public static Node<Integer> check(BinNode<Integer> t1, BinNode<Integer> t2)
{
    Node<Integer> first = new Node<Integer> (- 1);
    first = check(t1 , t2 , first);
    return first.getNext();
}
```

C#

```
public static Node<int> Check(BinNode<int> t1 , BinNode<int> t2)
{
    Node<int> first = new Node<int> (- 1);
    first = Check(t1 , t2 , first);
    return first.GetNext();
}
```

Implémentez la méthode :

En Java :

```
public static Node<Integer> check(BinNode<Integer> t1,
                                   BinNode<Integer> t2 , Node<Integer> list)
```

En C# :

```
public static Node<int> Check(BinNode<int> t1 , BinNode<int> t2 , Node<int>list)
```

Vous pourrez utiliser la méthode que vous avez implémentée à l'article א.

- א. Quelle est la complexité d'exécution de la méthode que vous avez implémentée à l'article ב ? Justifiez votre réponse.

Troisième section (25 points)

Cette section comporte des questions concernant quatre filières :

מערכות מחשב ואסמבלי, pages 9-11.

מבוא לחקר ביצועים, pages 12-16.

מודלים חישוביים, page 17.

Java-ב, תכנות מונחה עצמים ב-C#, pages 18-21 ; C#-ב, תכנות מונחה עצמים ב-Java, pages 22-25.

Répondez à une question de la filière dans laquelle vous avez étudié.

מערכות מחשב ואסמבלי

Si vous avez étudié dans cette filière, répondez à l'une des questions 7-8 (25 points).

7. Cette question comporte deux articles, א-ב, qui ne sont pas liés.

Répondez aux deux.

- א. Voici une portion de programme en assembleur dans laquelle AL et AH contiennent des nombres différents les uns des autres, non signés.

```
MOV AX , 7F80H
CMP AH , AL
JL  SMALL
BIG:  MOV BL , AH
      JMP  SOF
SMALL: MOV BL , AL
SOF:
```

- (1) Quelle sera la valeur de BL au terme de l'exécution de la portion de programme ?
- (2) Qu'exécute cette portion de programme ?
- (3) Dans cette portion de programme, on remplace l'instruction JL SMALL par l'instruction JB SMALL .
Au terme de l'exécution de la portion de programme, la valeur de BL sera-t-elle différente après et avant le remplacement de l'instruction ? Expliquer votre réponse.

- ב. (Il n'y a pas de lien avec l'article א)

Voici six affirmations. Pour chacune d'entre-elles, déterminez si elle est vraie ou fausse. Si l'affirmation est fausse, expliquez pourquoi.

(1) $10011110101110_2 > 27AE_{16}$

(2) $27AE_{16} > 10159_{10}$

- (3) Les fonctions des registres BP et SP sont :

BP détermine à quelle adresse de la pile sera inscrit un mot, et de quelle adresse de la pile un mot sera récupéré.

SP permet à une procédure de récupérer des paramètres de la pile, sans faire intervenir BP .

- (4) Au terme de l'appel et de l'exécution de chaque instruction, la valeur de IP est toujours incrémentée de 1 .

- (5) Voici deux portions de programme en assembleur.

Portion 1

MOV DX , 0000H

MOV AX , 0064H

DIV AL

Portion 2

MOV DX , 0000H

MOV AX , 0064H

DIV AX

Au terme de l'exécution des deux portions, le contenu de AX sera 0001H .

- (6) Voici deux portions de programme en assembleur.

Portion 1

MOV AL , 5BH

MOV CL , 8

ROL AL , CL

Portion 2

MOV AL , 5BH

MOV CL , 9

ROL AL , CL

Au terme de l'exécution de la portion 1, la valeur de AL sera égale à la valeur qui lui a été attribuée dans la première instruction, et également au terme de l'exécution de la portion 2, la valeur de AL sera égale à la valeur qui lui a été attribuée dans la première instruction.

8. Soit deux tableaux ARR1 et ARR2 . Ces deux tableaux sont de même longueur et chacun de leurs éléments est de la taille d'un mot. Dans chaque élément de chacun des tableaux, un nombre hexadécimal entier à quatre chiffres, situé dans l'intervalle 1000H - FFFFH est stocké.
- De plus, on donne une variable K de la taille d'un octet dans laquelle est stockée la longueur des tableaux ARR1 et ARR2 .

Écrivez en assembleur une portion de programme qui stockera dans le registre DX le nombre de paires de nombres (l'un provenant de ARR1 et l'autre de ARR2) qui vérifient les conditions suivantes :

- (i) les deux nombres de cette paire ont le même indice dans leur tableau.
- (ii) les deux nombres de cette paire sont constitués des mêmes chiffres mais en ordre inverse.

Exemple : pour les tableaux ARR1 et ARR2 de longueur 5 ci-dessous, le nombre 2 sera stocké dans le registre DX .

ARR1	2025H	1061H	1492H	5777H	1948H
ARR2	1984H	1601H	2914H	9999H	8491H

מבוא לחקר ביצועים

Si vous avez étudié dans cette filière, répondez à l'une des questions 9-10 (25 points).

9. Soit le problème de programmation linéaire :

$$\max \{z = 5x_1 - x_2\}$$

obéissant aux contraintes suivantes :

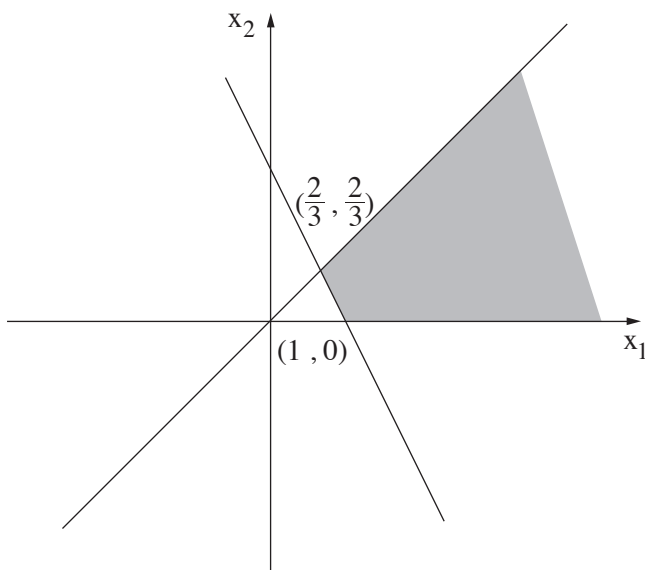
$$x_2 \leq x_1$$

$$2x_1 + x_2 \geq 2$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

Voici le diagramme des solutions admissibles du problème donné.



Chacun des articles κ -ה de la page suivante se rapporte au problème de programmation linéaire donné.

Les articles κ -ה ne sont pas liés. Répondez à tous les articles.

Pour les articles κ -ג, quatre affirmations i-iv sont proposées, et pour chaque article, une seule affirmation est vraie.

- i La solution optimale est unique.
- ii Il existe une infinité de solutions optimales.
- iii La solution optimale n'est pas bornée.
- iv Il n'existe pas de solution optimale.

Pour chacun des articles א-ז , déterminez laquelle des affirmations est la vraie, copiez-la dans votre cahier, puis justifiez votre réponse.

- Si vous avez établi pour un article quelconque, que l'affirmation i est la vraie – trouvez la solution optimale unique de cet article, ainsi que la valeur de la fonction objectif de cette solution.
- Si vous avez établi pour un article quelconque, que l'affirmation ii est la vraie – trouvez la solution optimale générale du problème, ainsi que la valeur de la fonction objectif dans l'intervalle des solutions optimales.
- א. Quelle affirmation est vraie pour le problème de programmation linéaire donné au début de la question ? Justifiez votre réponse.
- ב. On remplace uniquement la fonction objectif du programme donné au début de la question par $\max \{z = -5x_1 + x_2\}$.
Quelle affirmation est vraie suite au remplacement de la fonction objectif ? Justifiez votre réponse.
- ג. On remplace uniquement la fonction objectif du programme donné au début de la question par $\min \{z = -5x_1 + x_2\}$.
Quelle affirmation est vraie suite au remplacement de la fonction objectif ? Justifiez votre réponse.
- ד. On remplace uniquement la fonction objectif du programme donné au début de la question par $\max \{z = -2x_1 + 2x_2\}$.
Quelle affirmation est vraie suite au remplacement de la fonction objectif ? Justifiez votre réponse.
- ה. On ajoute une contrainte supplémentaire au problème de programmation linéaire donné au début de la question : $2x_1 - x_2 \leq 2$ et l'on remplace la fonction objectif du programme donné au début de la question par $\max \{z = ax_1 + x_2\}$ avec $(-1) < a < 2$ et la valeur maximale que la fonction objectif accepte est 4 sur l'intervalle des solutions admissibles.
Calculez la valeur de a .

10. Cette question comporte deux articles, א-ב, qui ne sont pas liés.

Répondez aux deux.

א. $G = (V, E)$ est un graphe **non-orienté** représenté par la liste d'adjacences ci-dessous :

a	→	b	→	c	→	d	→	
b	→	a	→	c	→			
c	→	a	→	b	→			
d	→	a	→	f	→	e	→	
e	→	d	→	f	→			
f	→	e	→	d	→			

- (1) Tracez le graphe G représenté par la liste d'adjacences.
- (2) Le graphe G donné est-il connexe ? Justifiez.
- (3) Appliquez un algorithme de recherche en profondeur (DFS) sur le graphe donné, en commençant par le sommet a .
Tracez dans votre cahier uniquement l'arbre couvrant (DFS) obtenu.
Basez-vous sur la représentation donnée par la liste d'adjacences.
- (4) Appliquez un algorithme de recherche en largeur (BFS) sur le graphe donné, en commençant par le sommet a .
Tracez dans votre cahier uniquement l'arbre couvrant (BFS) obtenu.
Basez-vous sur la représentation donnée par la liste d'adjacences.

ב. (Il n'y a pas de lien avec l'article א)

(1) Dans le tableau ci-dessous, on donne une solution de base admissible partielle à un problème de transport, ainsi que les valeurs de $u_1, u_2, u_3, v_1, v_2, v_3$.

Origines	Destinations			Offre	u_i
	1	2	3		
1	2 20	5	7	20	2
2	1	1 10	4	10	0
3	0	1	8 10	15	0
Demande	20	15	10		
v_j	0	1	8		

Recopiez le tableau dans votre cahier, puis complétez-le en tenant compte des valeurs des u_i et des v_j , afin d'obtenir une solution de base admissible.

(Attention : la suite de l'article se trouve à la page suivante)

- (2) Dans le tableau ci-dessous, on donne une solution de base admissible à un problème de transport, ainsi que les valeurs de $u_1, u_2, u_3, v_1, v_2, v_3$.

Origines	Destinations			Offre	u_i
	1	2	3		
1	14	15 130	17 50	180	0
2	10 80	8 20	14	100	-7
3	15	20	18 80	80	1
Demande	80	150	130		
v_j	17	15	17		

Cette solution est-elle optimale ? Justifiez votre réponse.

- (3) Dans le tableau ci-dessous, on donne une solution de base admissible à un problème de transport, et l'on donne $v_1 = 0$.

Origines	Destinations			Offre	u_i
	1	2	3		
1	10 20	25	30	20	
2	10 30	22	14 20	50	
3	18	20 40	20 20	60	
Demande	50	40	40		
v_j	0				

Recopiez le tableau dans votre cahier, puis complétez-y les valeurs de u_1, u_2, u_3, v_2, v_3 .

מודלים חישוניים

Si vous avez étudié dans cette filière, répondez à l'une des questions 11-12 (25 points).

11. Cette question comporte deux articles, א-ב, qui ne sont pas liés.

Répondez aux deux.

א. Soit une méthode écrite en Java et en C# .

Java

```
boolean foo(String str) {  
    int cntrA = 0;  
    int cntrC = 0;  
    for (int i = 0; i < str.length; i++) {  
        if (str[i] == 'a') cntrA++;  
        if (str[i] == 'c') cntrC++;  
    }  
    if ((cntrA % 2 == 0) &&  
        (cntrC % 3 == 0))  
        return true;  
    return false;  
}
```

C#

```
bool Foo(string str) {  
    int cntrA = 0;  
    int cntrC = 0;  
    for (int i = 0; i < str.Length; i++) {  
        if (str[i] == 'a') cntrA++;  
        if (str[i] == 'c') cntrC++;  
    }  
    if ((cntrA % 2 == 0) &&  
        (cntrC % 3 == 0))  
        return true;  
    return false;  
}
```

(1) Écrivez le langage L sur l'alphabet $\{a, c\}$ regroupant tous les mots pour lesquels la méthode donnée retourne `true` .

(2) Construisez un automate fini déterministe acceptant le langage L .

ב. (Il n'y a pas de lien avec l'article א)

Construisez un automate fini non-déterministe sur l'alphabet $\{a, b\}$ acceptant tous les mots contenant au moins une occurrence de l'une de ces séquences : `bbb` , `aaba` , `ababa` .

12. Voici le langage L sur l'alphabet $\{a, b, c\}$.

$$L = \{a^n b^m c^{n+m} \mid n, m > 0\}$$

א. Construisez un automate pile acceptant le langage L .

ב. Construisez une machine de Turing acceptant le langage L .

תכנות מונחה עצמים

Si vous avez étudié dans cette filière et que vous rédigez en Java, répondez à l'une des questions 13-14 (25 points).

13. Voici les interfaces IOne, ITwo, IThree et IFour ainsi que les classes Alpha , Beta , Gamma et Program.

```
public interface IOne {  
    public boolean firstA(Object x);  
    public void firstB(int num);  
}
```

```
public interface ITwo {  
    public int second();  
}
```

```
public interface IThree {  
    public int third();  
}
```

```
public interface IFour {  
    public int fourth();  
}
```

```
public class Alpha implements IOne {}  
public class Beta implements ITwo, IFour {}  
public class Gamma implements ITwo, IThree {}  
public class Program {  
    public static void main(String[] args) {}  
}
```

- ⌘. Écrivez les noms des méthodes qu'il faut implémenter dans chacune des classes Alpha , Beta et Gamma , puis expliquez pourquoi.

Supposez que dans chacune des classes, les méthodes que vous avez écrites à l'article 8 sont implémentées.

- ב. La déclaration `public class Omega implements IFour extends Beta { }` est-elle valide ?

Si elle n'est pas valide – expliquez pourquoi.

- ג. Pour chacun des segments i-iv ci-dessous, si on l'écrit dans la méthode `main`, déterminez si la méthode `main` ainsi obtenue est valide ou non.

Justifiez vos réponses

i `ITwo x1 = new ITwo();`

ii `Beta b = new Beta();`

iii `Alpha a = new Alpha();`

`IOne x2 = a;`

iv `Gamma c = new Gamma ();`

`IOne x3 = c ;`

- ד. Pour chacune des exigences i-ii ci-dessous, déterminez s'il est possible de les exécuter en écrivant une instruction ou des instructions dans la méthode `main`. Si cela est possible – écrivez l'instruction ou les instructions qui conviennent. Si cela est impossible – expliquez pourquoi.

i L'exécution de la méthode `second` sur un objet de type `Beta`.

ii La conversion d'un objet de type `Gamma` en objet de type `Alpha`.

14. Cette question comporte quatre articles, א-ד, qui ne sont pas liés.

Répondez à tous les articles.

א. Soit la classe A et la classe B héritant de A .

L'instruction `A a1 = new B();` est compilée puis s'exécute normalement.

Pour chacune des phrases i-iii suivantes, déterminez si elle est vraie ou fausse. Expliquez vos choix.

i Il est impossible d'écrire l'instruction `Object obj = a1;` , car cela nécessite un transtypage explicite (casting).

ii Il est impossible d'écrire l'instruction `A a2 = a1;` , car cela nécessite un transtypage explicite (casting).

iii Il est impossible d'écrire l'instruction `B b1 = a1;` , car cela nécessite un transtypage explicite (casting).

ב. Soit la méthode `main` dans la classe `Program` :

```
public static void main(String[] args) {  
    C c = new A();  
    B b1 = (B) (new A());  
    B b2 = new D();  
    A a = new D();  
}
```

Déterminez laquelle des propositions i-v suivantes décrit le lien entre les classes, d'après la méthode `main` donnée.

i C hérite de A , B hérite de A , D hérite de A .

ii A hérite de C , B hérite de A , D hérite de A .

iii C hérite de A , B hérite de A , D hérite de B .

iv A hérite de C , C hérite de B , D hérite de A .

v A hérite de C , B hérite de A , D hérite de B , C hérite de B .

ג. Soit les déclarations des classes School et City :

```
public class City{_____(1)____int students; }  
public class School extends City {  
    void poll() { System.out.println("There are "+ students + " in the school ") ; }  
}
```

Afin d'éviter une erreur de compilation, il est possible de compléter la partie de ligne notée (1) au moyen de l'une des propositions suivantes. Choisissez la proposition qui convient puis recopiez-la dans votre cahier.

- Écrire public , private ou protected .
- Écrire uniquement public ou private .
- Écrire uniquement public ou protected .
- Écrire uniquement protected .
- Écrire uniquement public .
- Écrire uniquement private .

ד. Voici la déclaration de la classe Quest :

```
public class Quest {  
    private int num;  
    private static int count = 0;  
  
    public Quest()  
    {  
        count++;  
        num = count;  
    }  
  
    public void printNow()  
    {  
        System.out.println (num + "" + count);  
    }  
}
```

Déterminez combien d'objets doit-on créer à partir de la classe Quest , et sur lesquels il faut appliquer la méthode printNow , afin que la sortie soit 35 .

תכנות מונחה עצמים

Si vous avez étudié dans cette filière et que vous rédigez en C#, répondez à l'une des questions 15-16 (25 points).

15. Voici les interfaces IOne, ITwo, IThree et IFour ainsi que les classes Alpha , Beta , Gamma et Program.

```
interface IOne {  
    bool FirstA(Object x);  
    void FirstB(int num);  
}  
  
interface ITwo {  
    int Second();  
}  
  
interface IThree {  
    int Third();  
}  
  
interface IFour {  
    int Fourth();  
}  
  
public class Alpha : IOne {}  
public class Beta : ITwo, IFour {}  
public class Gamma : ITwo, IThree {}  
public class Program {  
    public static void Main() {}  
}
```

- ⌘. Écrivez les noms des méthodes qu'il faut implémenter dans chacune des classes Alpha , Beta et Gamma , puis expliquez pourquoi.

Supposez que dans chacune des classes, les méthodes que vous avez écrites à l'article 8 sont implémentées.

ב. La déclaration `public class Omega : IFour, Beta { }` est-elle valide ?
Si elle n'est pas valide – expliquez pourquoi.

ג. Pour chacun des segments i-iv ci-dessous, si on l'écrivait dans la méthode `Main`, déterminez si la méthode `Main` ainsi obtenue est valide ou non. Justifiez vos réponses.

i `ITwo x1 = new ITwo();`

ii `Beta b = new Beta();`

iii `Alpha a = new Alpha();`

`IOne x2 = a;`

iv `Gamma c = new Gamma ();`

`IOne x3 = c ;`

ד. Pour chacune des exigences i-ii ci-dessous, déterminez s'il est possible de les exécuter en écrivant une instruction ou des instructions dans la méthode `Main`. Si cela est possible – écrivez l'instruction ou les instructions qui conviennent. Si cela est impossible – expliquez pourquoi.

i L'exécution de la méthode `Second` sur un objet de type `Beta`.

ii La conversion d'un objet de type `Gamma` en objet de type `Alpha`.

16. Cette question comporte quatre articles, א-ד, qui ne sont pas liés.

Répondez à tous les articles.

א. Soit la classe A et la classe B héritant de A .

L'instruction `A a1 = new B();` est compilée puis s'exécute normalement.

Pour chacune des phrases suivantes, déterminez si elle est vraie ou fausse. Expliquez vos choix.

- i Il est impossible d'écrire l'instruction `Object obj = a1;` , car cela nécessite un transtypage explicite (casting).
- ii Il est impossible d'écrire l'instruction `A a2 = a1;` , car cela nécessite un transtypage explicite (casting).
- iii Il est impossible d'écrire l'instruction `B b1 = a1;` , car cela nécessite un transtypage explicite (casting).

ב. Soit la méthode Main dans la classe Program :

```
public static void Main(string[] args) {  
    C c = new A();  
    B b1 = (B) (new A());  
    B b2 = new D();  
    A a = new D();  
}
```

Déterminez laquelle des propositions i-v suivantes décrit le lien entre les classes, d'après la méthode Main donnée.

- i C hérite de A , B hérite de A , D hérite de A .
- ii A hérite de C , B hérite de A , D hérite de A .
- iii C hérite de A , B hérite de A , D hérite de B .
- iv A hérite de C , C hérite de B , D hérite de A .
- v A hérite de C , B hérite de A , D hérite de B , C hérite de B .

ג. Soit les déclarations des classes School et City :

```
public class City{__(1)__int students; }  
public class School : City {  
    void Poll() { Console.WriteLine("There are "+ students + " in the school ") ; }  
}
```

Afin d'éviter une erreur de compilation, il est possible de compléter la partie de ligne notée (1) au moyen de l'une des propositions suivantes. Choisissez la proposition qui convient puis recopiez-la dans votre cahier.

- Écrire public , private ou protected .
- Écrire uniquement public ou private .
- Écrire uniquement public ou protected .
- Écrire uniquement protected .
- Écrire uniquement public .
- Écrire uniquement private .

ד. Voici la déclaration de la classe Quest :

```
public class Quest {  
    private int num;  
    private static int count = 0;  
  
    public Quest()  
    {  
        count++;  
        num = count;  
    }  
    public void PrintNow()  
    {  
        Console.WriteLine(num+" "+count);  
    }  
}
```

Déterminez combien d'objets doit-on créer à partir de la classe Quest , et sur lesquels il faut appliquer la méthode PrintNow , afin que la sortie soit 35 .

Bonne chance!

בהצלחה!