État d'Israël
Ministère de l'Éducation
Type d'examen : *Bagrout* pour écoles secondaires
Session de l'examen : été 2016
Numéro de l'examen : 899381
Traduction française (4)

מדינת ישראל
משרד החינוך
סוג הבחינה: בגרות לבתי ספר על־יסודיים
מועד הבחינה: קיץ תשע"ו, 2016
מספר השאלון: 899381
תרגום לצרפתית (4)

Informatique

d'après le programme de la reforme pour un apprentissage concret

Instructions au candidat

א. Durée de l'examen : trois heures.

ב. Structure du questionnaire et barème de notation :
Ce questionnaire comporte trois parties.
Première partie — Cette partie comporte trois questions, répondez d'après les instructions enoncées dans cette partie.
$(1 \times 10) + (1 \times 15)$ — 25 points
Deuxième partie — Cette partie comporte trois questions, répondez à deux d'entre elles.
(2×25) — 50 points
Troisième partie — Cette partie comporte des questions de quatre filières différentes.
Répondez à une question de la filière dans laquelle vous avez étudié
(1×25) — 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 partie, 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 חישוביים.

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 !

Bonne Chance!

מדעי המחשב

על פי תכנית הרפורמה ללמידה משמעותית

הוראות לנבחן

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח ההערכה:
בשאלון זה שלושה פרקים.
פרק ראשון – בפרק זה שלוש שאלות, ענה על פי ההוראות בפרק.
(10×1) + (15×1) — 25 נקודות

פרק שני – בפרק זה יש שלוש שאלות, ומהן עליך לענות על שתיים.
(25×2) — 50 נקודות

פרק שלישי – בפרק זה שאלות בארבעה מסלולים שונים.
ענה על שאלה אחת במסלול שלמדת.
(25×1) — 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 ה**מסלול שלמדת.**
המסלול הוא אחד מארבעת המסלולים האלה:
מערכות מחשב ואסמבלי, מבוא לחקר ביצועים,
מודלים חישוביים, 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!

בהצלחה!

Questions

Ce questionnaire comporte trois parties.
Répondez à des questions des trois parties, en suivant les instructions énoncées dans chaque partie.

Première partie (25 points)

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

À 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. Soit la classe élève – **Student** comportant deux attributs :
le nom de l'élève – name de type chaîne, et un tableau unidimensionnel arrTest de longueur 3 de type entier. Chaque case du tableau contient une note de l'élève à un examen.
Écrivez en Java ou en C#, une méthode dans la classe **Student** qui renverra la note moyenne de l'élève aux trois examens.
Remarque : il n'est pas nécessaire de vérifier la validité des données contenues dans le tableau.

/suite en page 3/

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

2. Soit la classe acteur – **Actor** comportant trois attributs :
le numéro d'identité – id de type chaîne, le sexe de l'acteur – gender de type chaîne ("F" représente une femme, "M" représente une homme), le nombre de films dans lesquels l'acteur a joué – numFilms de type entier.
Voici l'interface de la classe **Actor** écrite en Java et en C#.

Déclaration de la méthode en Java	Description de la méthode
public Actor(String id, String gender, int numFilms)	Méthode construisant un acteur dont le numéro d'identité est id , dont le sexe est gender , dont le nombre de films dans lequel il a joué est numFilms .
public void addFilm()	Méthode ajoutant 1 au nombre de films dans lesquels l'acteur a joué.
public int compare(Actor other)	Méthode renvoyant : 1 – si le nombre de films dans lesquels l'acteur actuel a joué est supérieur au nombre de films dans lesquels l'acteur other a joué. 2 – si le nombre de films dans lesquels l'acteur actuel a joué est inférieur au nombre de films dans lesquels l'acteur other a joué. 3 – si le nombre de films dans lesquels l'acteur actuel a joué est égal au nombre de films dans lesquels l'acteur other a joué.

(Attention : la suite de la question se trouve à la page suivante)

/suite en page 4/

Déclaration de la méthode en C#	Description de la méthode
public Actor(string id, string gender, int numFilms)	Méthode construisant un acteur dont le numéro d'identité est id , dont le sexe est gender , dont le nombre de films dans lequel il a joué est numFilms .
public void AddFilm()	Méthode ajoutant 1 au nombre de films dans lesquels l'acteur a joué.
public int Compare(Actor other)	Méthode renvoyant : 1 – si le nombre de films dans lesquels l'acteur actuel a joué est supérieur au nombre de films dans lesquels l'acteur other a joué. 2 – si le nombre de films dans lesquels l'acteur actuel a joué est inférieur au nombre de films dans lesquels l'acteur other a joué. 3 – si le nombre de films dans lesquels l'acteur actuel a joué est égal au nombre de films dans lesquels l'acteur other a joué.

Supposez que pour chaque attribut, les méthodes get et set en Java et les méthodes Get et Set en C#, ont été définies.

א. Implémentez la méthode constructeur de la classe **Actor** .
En Java, la méthode constructeur :
public Actor(String id, String gender, int numFilms)
En C#, la méthode constructeur :
public Actor(string id, string gender, int numFilms)

ב. Implémentez en Java la méthode compare ou en C# la méthode Compare .

ג. Dans la méthode principale de la classe Program , on a défini un tableau unidimensionnel actArr de taille 37 de type **Actor** .
Écrivez en Java ou en C# une méthode dans la classe Program qui recevra un tableau unidimensionnel de type **Actor** et un nombre num entier supérieur à 0 .
La méthode affichera le nombre d'acteurs dans le tableau qui ont joué dans plus de num films.

/suite en page 5/

3. Soit la classe programme télévisé – **TvProgram** comportant trois attributs : le code du programme – code de type entier, le jour de la semaine où est diffusé ce programme – day de type entier, recevant des valeurs entre 1 et 7 (inclus), si c'est un programme sportif – isSport de type booléen, dont la valeur est true s'il s'agit d'un programme sportif, et false si ce n'est pas le cas.
Supposez que pour chaque attribut, les méthodes get et set en Java et les méthodes Get et Set en C#, ont été définies.

א. Écrivez en Java ou en C#, une méthode constructeur dans la classe **TvProgram** recevant des valeurs pour chaque attribut.

ב. Soit la classe programmes de la semaine – **TvWeek** comportant deux attributs :
un tableau unidimensionnel – arrProg de longeur 100 de type **TvProgram** , le nombre de programmes actuels par semaine – current de type entier, dont la valeur est inférieure à la longueur du tableau.
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 méthode **TvWeek** ainsi que ses attributs.

(2) Écrivez en Java ou en C#, dans la classe **TvWeek** , une méthode qui recevra un programme **TvProgram** , l'ajoutera aux programmes de la semaine puis actualisera le nombre de programmes actuels de la semaine.
Supposez qu'il existe de la place pour ajouter ce programme.

(3) Écrivez en Java ou en C#, dans la classe **TvWeek** , une méthode qui renverra le nombre de programmes sportifs diffusés au cours de la semaine à la télévision.

/suite en page 6/

Deuxième partie (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 classe Anneau – **Ring** comportant deux attributs :

La taille de l'anneau de type chaîne ("S" – petit anneau, "L" – grand anneau);

Un nombre entier représentant la couleur de l'anneau.

Java

```
public class Ring
{
   private String size;  //taille de l'anneau
   private int color; //couleur de l'anneau
   public Ring()
   {
      this.size = "L";
      this.color = 0;
   }
   public Ring(String str, int c)
   {
      this.size = str;
      this.color = c;
   }
   public String getSize()
   {
      return this.size;
   }
   public int getColor()
   {
      return this.color;
   }
}
```

C#

```
public class Ring
{
   private string size;  //taille de l'anneau
   private int color; //couleur de l'anneau
   public Ring()
   {
      this.size = "L";
      this.color = 0;
   }
   public Ring(string str, int c)
   {
      this.size = str;
      this.color = c;
   }
   public string GetSize()
   {
      return this.size;
   }
   public int GetColor()
   {
      return this.color;
   }
}
```

/suite en page 7/

Voici l'interface de la classe Barre – **Pole** .

Java	C#	Méthode
public Pole()	public Pole()	Méthode construisant une barre vide. La complexité de l'exécution de la méthode est O(1)
public void add(Ring r)	public void Add(Ring r)	Méthode enfilant un anneau r au sommet de la barre. La complexité de l'exécution de la méthode est O(1)
public Ring remove()	public Ring Remove()	Méthode renvoyant l'anneau du sommet de la barre. De plus, la méthode sort cet anneau du sommet de la barre. La complexité de l'exécution de la méthode est O(1)
public boolean isEmpty()	public bool IsEmpty()	Si la barre est vide, la méthode renvoie true , autrement, elle renvoie false. La complexité de l'exécution de la méthode est O(1)
public void sort()	public void Sort()	Méthode rangeant les anneaux enfilés sur la barre ainsi : les grands anneaux "sont posés" à la base de la barre et les petits anneaux sont par-dessus.

א. Implémentez la méthode sort() en Java et Sort() en C#, de la classe **Pole** .

Vous pouvez utiliser les autre méthodes de la classe **Pole** sans les implémenter.

Utilisez uniquement les méthodes des classes **Pole** et **Ring** dans votre réponse.

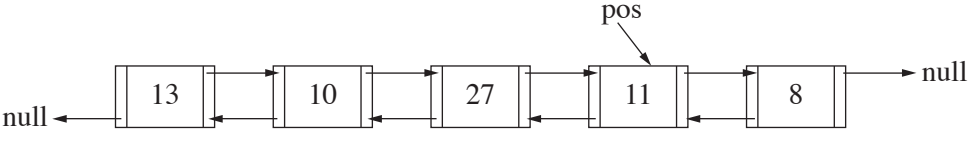
ב. Quelle est la complexité d'exécution de la méthode que vous avez implémentée en א ? Justifiez votre réponse.

/suite en page 8/

5. **Attention** : Cette question comporte deux versions :
une version en Java aux pages 8-9, et une autre version en C#
aux pages 10-11.
Composez selon le langage que vous avez étudié.

Pour ceux qui composent en Java

On définit une **liste bi-directionnelle** comme un ensemble ordonné de maillons de type BinNode<Integer> liés ainsi :
Pour chaque paire de maillons p1 , p2 de la liste, si p1.getRight() = = p2 est vérifié, alors p2.getLeft() = = p1 est également vérifié.
Une **liste bi-directionnelle** comporte au minimum deux maillons.
C'est-à-dire : chaque maillon de la liste, à l'exception du maillon situé à l'extrémité droite de la liste et le maillon situé à l'extrémité gauche de la liste, pointe sur le maillon qui le précède et sur le maillon qui le suit.
Voici un exemple d'**une liste bi-directionnelle** et d'une variable pos de type BinNode<Integer> pointant sur un maillon quelconque dans la liste bi-directionnelle.



La méthode firstLeft reçoit un pointeur pos différent de null de type BinNode<Integer> pointant sur un maillon quelconque dans la liste bi-directionnelle, puis renvoie le maillon situé le plus à gauche dans la liste.
La méthode firstRight reçoit un pointeur pos différent de null de type BinNode<Integer> pointant sur un maillon quelconque dans la liste bi-directionnelle, puis renvoie le maillon situé le plus à droite dans la liste.

א. Voici le squelette de la méthode firstLeft .
Copiez-le dans votre cahier, puis complétez-le, de sorte que la méthode effectue ce qui est requis.

```
public static BinNode<Integer> firstLeft(BinNode<Integer> pos)
{
    while (_____ )
        pos = _____ ;
    return _____ ;
}
```

/suite en page 9/

ב. Voici la méthode what(BinNode<Integer> pos) recevant un pointeur sur un maillon quelconque dans une **liste bi-directionnelle** et qui renvoie une valeur booléenne. La **liste bi-directionnelle** comporte au moins 3 maillons.

(1) Tracez l'exécution de la méthode pour la variable pos et la liste de l'exemple donné en début d'énoncé.
Lors de la trace, montrez la liste bi-directionnelle ainsi que les valeurs des variables sum, right, left, pos .

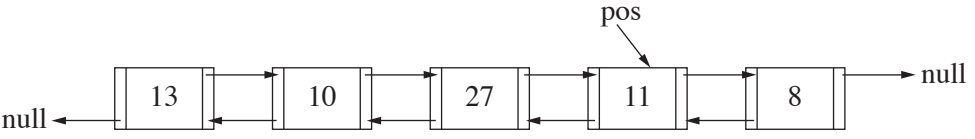
```
public static boolean what(BinNode<Integer> pos)
{
    BinNode<Integer> left = firstLeft(pos);
    BinNode<Integer> right = firstRight(pos);

    int sum = left.getValue() + right.getValue();
    left = left.getRight();
    right = right.getLeft();

    while ((left != right) && (left.getRight() != right) &&
           (left.getValue() + right.getValue() == sum))
    {
        left = left.getRight();
        right = right.getLeft();
    }
    if (left == right)
        return right.getValue() == sum;

    if (left.getRight() == right)
            return left.getValue() + right.getValue() == sum;
    return false;
}
```

(2) Déterminez s'il est possible ou non de remplacer les 3 dernières lignes de la méthode – les lignes encadrées – par l'instruction :
return left.getValue() + right.getValue() == sum;
Justifiez votre réponse.

/suite en page 10/

Pour ceux qui composent en C#

On définit une **liste bi-directionnelle** comme un ensemble ordonné de maillons de type BinNode<int> liés ainsi :
Pour chaque paire de maillons p1 , p2 de la liste, si p1.GetRight() = = p2 est vérifié, alors p2.GetLeft() = = p1 est également vérifié.
Une **liste bi-directionnelle** comporte au minimum deux maillons.
C'est-à-dire : chaque maillon de la liste, à l'exception du maillon situé à l'extrémité droite de la liste et le maillon situé à l'extrémité gauche de la liste, pointe sur le maillon qui le précède et sur le maillon qui le suit.
Voici un exemple d'**une liste bi-directionnelle** et d'une variable pos de type BinNode<int> pointant sur un maillon quelconque dans la liste bi-directionnelle.



La méthode FirstLeft reçoit un pointeur pos différent de null de type BinNode<int> pointant sur un maillon quelconque dans la liste bi-directionnelle, puis renvoie le maillon situé le plus à gauche dans la liste.
La méthode FirstRight reçoit un pointeur pos différent de null de type BinNode<int> pointant sur un maillon quelconque dans la liste bi-directionnelle, puis renvoie le maillon situé le plus à droite dans la liste.

א. Voici le squelette de la méthode FirstLeft .
Copiez-le dans votre cahier, puis complétez-le, de sorte que la méthode effectue ce qui est requis.

```
public static BinNode<int> FirstLeft(BinNode<int> pos)
{
    while (_____________________________ )
        pos = _____________________ ;
    return _____________________________ ;
}
```

/suite en page 11/

ב. Voici la méthode What(BinNode<int> pos) recevant un pointeur sur un maillon quelconque dans une **liste bi-directionnelle** et qui renvoie une valeur booléenne. La **liste bi-directionnelle** comporte au moins 3 maillons.

(1) Tracez l'exécution de la méthode pour la variable pos et la liste de l'exemple donné en début d'énoncé.
Lors de la trace, montrez la liste bi-directionnelle ainsi que les valeurs des variables sum, right, left, pos .

```
public static bool What(BinNode<int> pos)
{
    BinNode<int> left = FirstLeft(pos);
    BinNode<int> right = FirstRight(pos);

    int sum = left.GetValue() + right.GetValue();
    left = left.GetRight();
    right = right.GetLeft();

    while ((left != right) && (left.GetRight() != right) &&
           (left.GetValue() + right.GetValue() == sum))
    {
        left = left.GetRight();
        right = right.GetLeft();
    }
    if (left == right)
        return right.GetValue() == sum;

    if (left.GetRight() == right)
            return left.GetValue() + right.GetValue() == sum;
    return false;
}
```

(2) Déterminez s'il est possible ou non de remplacer les 3 dernières lignes de la méthode – les lignes encadrées – par l'instruction :
return left.GetValue() + right.GetValue() == sum;
Justifiez votre réponse.

/suite en page 12/

6. Un **arbre de nombres** est un arbre binaire non vide de type entier, dont les valeurs contenues dans les nœuds sont des nombres entiers supérieurs à 0 et différents les uns des autres.

On a défini, sur un **arbre de nombres**, une méthode "parcours ascendant" renvoyant true s'il existe dans cet arbre un parcours débutant à la racine de l'arbre et se terminant sur l'une de ses feuilles, et que les valeurs des nœuds sont triées par ordre croissant de la racine à cette feuille.

Si un tel parcours n'existe pas, la méthode renvoie false .

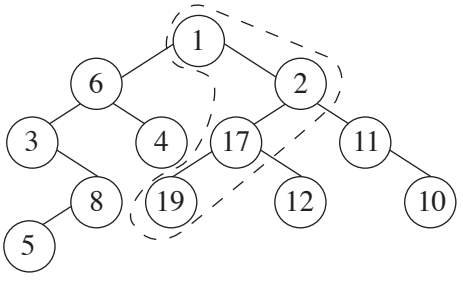
Par exemple :

Pour l'**arbre de nombres** tr1 la méthode "parcours ascendant" renvoie true .

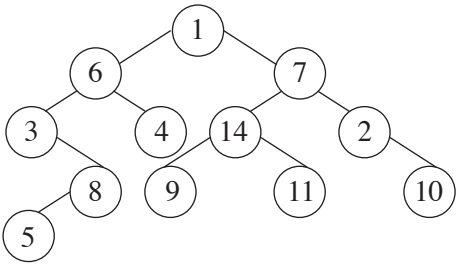
Le parcours est entouré d'une ligne en pointillé.

Pour l'**arbre de nombres** tr2 la méthode "parcours ascendant" renvoie false .

tr1



tr2



Implémentez en Java ou en C#, la méthode "parcours ascendant" pour un **arbre de nombres** tr .

Déclaration de la méthode en Java :

```
public static boolean upPath(BinNode<Integer> tr)
```

Déclaration de la méthode en C# :

```
public static bool UpPath(BinNode<int> tr)
```

/suite en page 13/

Troisième partie (25 points)

Cette partie comporte des questions concernant quatre filières :

מערכות מחשב ואסמבלי, pages 13-16.

מבוא לחקר ביצועים, pages 17-20.

מודלים חישוביים, pages 21.

תכנות מונחה עצמים ב–Java, pages 22-26 ; תכנות מונחה עצמים ב–C#, pages 27-31.

Répondez uniquement aux questions de la filière dans laquelle vous avez étudié.

מערכות מחשב ואסמבלי

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

7. Cette question comporte deux paragraphes, א-ב, qui ne sont pas liés.
Répondez aux deux.

א. Écrivez en assembleur une portion de programme cryptant un caractère.
Le caractère est défini ainsi dans le segment de données :
TAV DB ?
Le cryptage s'effectue ainsi :
On compte le nombre de bits allumés dans le caractère.
Si ce nombre est pair, on effectue un déplacement circulaire vers la droite des bits du caractère, un nombre de fois égal au nombre de bits allumés.
Si ce nombre est impair, on effectue un déplacement circulaire vers la gauche des bits du caractère, un nombre de fois égal au nombre de bits allumés.

ב. (Il n'y a pas de lien avec le paragraphe א)
Pour chacune des affirmations (1)-(6) ci-dessous, déterminez si elle est vraie ou fausse. Si l'affirmation est fausse, expliquez pourquoi.

(1) Voici une portion de programme en assembleur.

```
MOV  AX , 8
MOV  BX , 2
DIV  BX
```

Au terme de la portion de programme, le registre AX contiendra nécessairement 4 .

(Attention : la suite de la question se trouve à la page suivante)

/suite en page 14/

(2) Voici une portion de programme en assembleur.

```
        MOV   AL , 56
        ADD   AL , 200
        JZ    STOP
        INC   AL
STOP:
```

Au terme de la portion de programme, le registre AX contiendra 1 .

(3) Voici une portion de programme en assembleur.

```
ARRAY   DW 1, 2, 3, 4
MOV     BX , ARRAY
ADD     BX , 2
MOV     AX , [BX]
```

Au terme de la portion de programme, le registre AX contiendra 3 .

(4) Voici une portion de programme en assembleur.

```
      MOV   CX , 3
      MOV   AX , 1
DO:
      SHL   AX , 1
      LOOP  DO
```

Au terme de la portion de programme, le registre AX contiendra 8 .

(5) Voici une portion de programme en assembleur.

```
MOV  AX , 11000001B
MOV  BX , 01000001B
```

Le nombre contenu dans AX est nécessairement supérieur au nombre contenu dans BX .

(6) Soit l'instruction :

```
OR   AL , 3
```

Après l'exécution de l'instruction, la valeur de AL est toujours impaire.

/suite en page 15/

8. Cette question comporte deux paragraphes, א-ב, qui ne sont pas liés.
Répondez aux deux.

א. Dans le segment de données, les données ont été définies ainsi :

```
VEC1    DB   45H , 26H , 32H , 82H
VEC2    DB   4 DUP(0)
```

Voici une portion de programme en assembleur.

```
START:  MOV     CL , 4
        XOR     CH , CH
        XOR     SI , SI
NEXT:   PUSH    CX
        MOV     AL , VEC1[SI]
        MOV     AH , AL
        AND     AL , 0FH
        MOV     CL , 4
        SHR     AH , CL
        MUL     AH          ;(*)
        MOV     VEC2[SI] , AL
        INC     SI
        POP     CX
        LOOP    NEXT
SOF:    NOP
```

(1) À l'aide d'un tableau de trace, tracez l'exécution de la portion de programme, puis dessinez VEC2 suite à cette exécution.
Le tableau de trace devra comporter une colonne pour chacun de ces registres : AL , AH , CL , CH , SI .
Dessinez la pile à chaque étape.

(2) Qu'exécute la portion de programme ?

(3) Dans la portion de programme, l'instruction marquée par (*) a été remplacée par l'instruction IMUL AH .
L'exécution de la portion de programme a-t-elle changé ? Justifiez votre réponse.

(Attention : le paragraphe ב de la question se trouve à la page suivante)

/suite en page 16/

ב. Le nombre 2 est stocké dans le registre AL et le nombre 5 est stocké dans le registre BL .
Vous devez stocker dans le registre DX la somme des nombres allant de AL à BL (inclus).
Voici trois portions i-iii écrites en assembleur. À l'aide d'un tableau de trace, tracez l'exécution de chacune de ces portions, puis déterminez si elle effectue ou non l'action requise.

```
i                MOV     DX , 0
                 MOV     AH , 0
                 MOV     CL , AL
                 SUB     CL , BL
        AGAIN:   ADD     DX , AX
                 INC     AL
                 LOOPE   AGAIN
                 NOP

ii               XOR     DX , DX
                 MOV     BH , 0
        AGAIN:   ADD     DX , BX
                 DEC     BL
                 CMP     BL , AL
                 JGE     AGAIN
                 NOP

iii              XOR     DX , DX
                 XOR     AX , AX
                 MOV     BX , 0
        AGAIN:   ADD     DX , AX
                 ADD     AX , 1
                 CMP     AX , BX
                 JL      AGAIN
                 NOP
```

/suite en page 17/

מבוא לחקר ביצועים

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

9. Soit le problème de programmation linéaire :

$$\max\{z = (2 + 2k)x_1 + 2x_2\}$$

obéissant aux contraintes suivantes :

(1) $2x_1 + x_2 \leq 10$

(2) $x_1 + x_2 \leq 6$

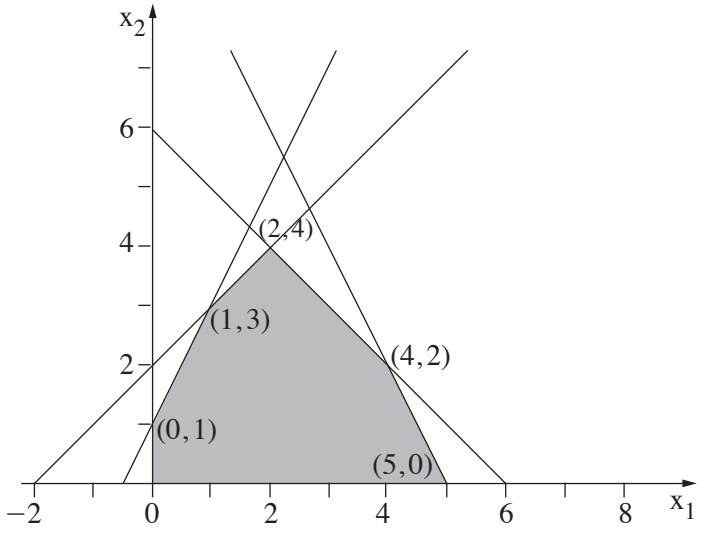
(3) $-x_1 + x_2 \leq 2$

(4) $-2x_1 + x_2 \leq 1$

$x_1 \geq 0$

$x_2 \geq 0$

Voici le diagramme des solutions admissibles du problème donné.



(Attention : les paragraphes de la question se trouvent à la page suivante)

/suite en page 18/

Chacun des paragraphes א-ד se rapporte au problème de programmation linéaire donné.

Ces questions ne sont pas liées. Répondez à tous les paragraphes.

א. Voici deux sous-paragraphes (1)-(2) qui **ne sont pas liés**. Dans chacun d'entre eux, une certaine valeur du paramètre k est donnée.

(1) $k = -1$

(2) $k = -3$

Pour chacun des sous-paragraphes (1)-(2), les quatre affirmations i-iv suivantes sont proposées :

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 chaque sous-paragraphe, une seule des affirmations i-iv est vraie.

Pour chacun des sous-paragraphes (1)-(2), déterminez laquelle des affirmations est la vraie, copiez-la dans votre cahier, puis justifiez votre réponse.

– Si vous avez établi que l'affirmation i est la vraie – vous devez trouver la solution optimale unique de ce sous-paragraphe, ainsi que la valeur de la fonction objectif de la solution que vous avez trouvée.

– Si vous avez établi que l'affirmation ii est la vraie – vous devez trouver la solution optimale générale du problème, ainsi que la valeur de la fonction objectif dans l'intervalle des solutions optimales.

ב. Pour quelle valeurs de k , $(2\ ,4)$ sera une solution optimale au problème de programmation linéaire donné en début d'énoncé ? Justifiez votre réponse.

ג. On supprime la contrainte $x_1 \geq 0$ du problème donné en début d'énoncé, c'est à dire que $-\infty \leq x_1 \leq \infty$.
Existe-t-il une valeur de k pour laquelle la solution optimale soit non bornée ? Justifiez votre réponse.

ד. À la place de la contrainte $x_2 \geq 0$ du problème donné en début d'énoncé, on écrit : $x_2 \geq 4$.
Le problème admet-il une solution optimale après la modification de la contrainte ? Si oui, donnez cette solution, si non, expliquez pourquoi il n'existe pas de solution optimale.

/suite en page 19/

10. Cette question comporte trois paragraphes, א-ג, qui ne sont pas liés.
Répondez aux trois paragraphes.

א. Soit G = (V , E) , un graphe **orienté** représenté par la matrice d'adjacences ci-contre :

	a	**b**	**c**	**d**	**e**
a	0	1	1	0	0
b	0	0	0	1	0
c	0	1	0	0	0
d	0	0	1	0	1
e	0	1	0	0	0

(1) Tracez le graphe G représenté par la matrice.
(2) Trouvez les composantes fortement connexes (Strong Connected Components – CFC) du graphe donné. Pour chaque CFC que vous avez trouvé, donnez le groupe de ses sommets.
(3) Trouvez dans le graphe donné, un cycle de longueur minimale et dont le nombre d'arcs est pairs, puis tracez-le dans votre cahier.
(4) Quel est le nombre minimal d'arcs qu'il faut ajouter au graphe donné afin que celui-ci ne comporte qu'un seul CFC ?
Quel est l'arc ou quels sont les arcs qu'il faut ajouter ?

ב. Dans le tableau ci-dessous, on donne une partie d'une solution de base admissible à un problème de transport : $x_{11} = 10$.

Origines	**Destinations**				**Offre**
	1	2	3	4	
1	20 (10)	20	30	21	20
2	22	17	29	30	30
3	10	24	26	38	10
Demande	10	20	20	10	

(1) Recopiez le tableau dans votre cahier, puis complétez-y la solution de base admissible selon la méthode du coin nord-ouest.
(2) On change le prix de la case (3 , 1) de 10 en 20 . Cela modifiera-t-il la solution de base admissible que vous avez trouvée en **(1)** ? Justifiez.

(Attention : le paragraphe ג de la question se trouve à la page suivante)

/suite en page 20/

ג. Dans le tableau ci-dessous, on donne une solution de base admissible à un problème de transport, et on donne la valeur de u_3 .

Origines	**Destinations**			**Offre**	u_i
	1	2	3		
1	10	8	3 (10)	10	
2	12	15	9 (3)	3	
3	2 (20)	7 (40)	1 (30)	90	0
Demande	20	40	43		
v_j					

(1) Recopiez le tableau dans votre cahier, puis complétez-y les valeurs de u_1 , u_2 , v_1 , v_2 , v_3 .

(2) Expliquez pourquoi la solution donnée n'est pas optimale.

/suite en page 21/

מודלים חישוביים

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

11. Soit les deux langages réguliers L_1 et L_2 .

$L_1 = \{a^{2n} \mid n \geq 0\}$ sur l'alphabet $\{a\}$

$L_2 = \{b^{2n+1} \mid n \geq 0\}$ sur l'alphabet $\{b\}$

Soit le langage L sur l'alphabet $\{a , b\}$.

$L = \{a^n b^k \mid n \geq 0 , k \geq 0 ,$ n est pair et k est impair ou n est impair et k est pair$\}$

א. Démontrez, au moyen des langages L_1 et L_2 uniquement et au moyen des propriétés de clôture, que le langage L est régulier.

ב. Construisez un automate fini non déterministe acceptant le langage L .

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

La méthode reçoit trois nombres entiers supérieurs à 0 .

Java

```
public static int foo(int x , int y , int z)
{
    if ((x % 3) == 0) return x;
    if ((x % 3) == 1) return y;
    return z;
}
```

C#

```
public static int Foo(int x , int y , int z)
{
    if ((x % 3) == 0) return x;
    if ((x % 3) == 1) return y;
    return z;
}
```

Écrivez une machine de Turing qui implémentera la méthode donnée.

L'entrée de la machine consiste dans trois nombres x , y , z que la méthode reçoit et est inscrite au début du ruban. Chaque nombre est écrit sous forme unaire. Les nombres sont séparés par le signe # .

Par exemple, si la méthode reçoit 2 pour x , 3 pour y , et 1 pour z , le ruban apparaîtra ainsi :

⊢	1	1	#	1	1	1	#	1	△	△	△	.....

↑

tête de lecture/écriture

La sortie est la valeur que la méthode renvoie et elle sera inscrite à un endroit quelconque du ruban en valeur unaire entre deux signes $.

/suite en page 22/

תכנות מונחה עצמים

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

13. Cette question comporte deux paragraphes, א-ב, qui ne sont pas liés. Répondez aux deux.

א. Soit cinq classes B , X , Y , Z , Run . Les sous-paragraphes (1)-(4) ci-dessous se rapportent à ces classes. Les sous-paragraphes ne sont pas liés les uns aux autres. Répondez à tous.

(1) Dans chacune des classes B , X , Y , Z a été définie la méthode
public void foo() {}
et dans la classe Run a été définie la méthode
public void bar (Object g) {g.foo(); }

Obtiendra-t-on une erreur de compilation ?
Si oui, expliquez pourquoi.

(2) Supposez que les classes X , Y , Z héritent de la classe B .
Dans chacune des classes X , Y , Z a été définie la méthode
public void foo() {}
et dans la classe Run a été définie la méthode
public void bar(B g) {g.foo(); }

Obtiendra-t-on une erreur de compilation ?
Si oui, expliquez pourquoi.

(3) Supposez que les classes X , Y , Z héritent de la classe B .
Dans chacune des classes B , X , Y , Z a été définie la méthode
public void foo() {}
et dans la classe Run a été définie la méthode
public void bar(B g) {g.foo(); }

Obtiendra-t-on une erreur de compilation ?
Si oui, expliquez pourquoi.

(4) Supposez que les classes X , Y , Z héritent de la classe B .
Dans chacune des classes B , X , Y , Z a été définie la méthode
public void foo() {}
et dans la classe Run a été définie la méthode
public void bar(Object g) {g.foo(); }

Obtiendra-t-on une erreur de compilation ?
Si oui, expliquez pourquoi.

/suite en page 23/

ב. (Il n'y a pas de lien avec le paragraphe א)

Voici une classe **Singer** héritant de la classe **Artist** et l'interface **IPerform** que la classe **Singer** implémente.

```
interface IPerform
{
   void act();
   int train();
}
public class Singer extends Artist implements IPerform
{
      private int hits;
      public Singer(String name, double sal)
      {
            super(sal, name);
            this.hits = 5;
      }
      public Singer(double sal, int hits)
      {
            super(sal, "Singer Name");
            this.hits = hits;
      }
      public Singer(int hits)
      {
            super(6532.6, "Some", "One");
            this.hits = hits;
      }
      public double value() {return this.hits * this.price();}
      public int getNum()    {return Artist.num;                              }
      public double calc(double d)        {return d * super.calc(10.2);      }
      public void print()      {super.print(); System.out.println("Singer");}
      public void act()        {System.out.println("I am singing");          }
}
```

Écrivez en Java la déclaration de la classe **Artist** , les attributs et les déclarations des méthodes requises de la classe **Singer** et de l'interface **IPerform** données. Il n'est pas nécessaire d'implémenter les méthodes de la classe **Artist** .

Ne modifiez pas la classe **Singer** et l'interface **IPerform** .

/suite en page 24/

14. Voici les classes A , B .

Les trois paragraphes א-ג ci-dessous se rapportent à ces classes, mais ne sont pas liés les uns aux autres. Répondez aux trois.

```
public class A    {
      private int x;
      public A()                { this.x = 0;        }
      public A(int x)           { this.x = x;                  }
      public int getX()         { return x;                    }
      public void doubleX()     { this.x = 2 * getX();         }
      public void tripleX()     { this.x = 3 * getX();         }
      public void sub()         { this.x = x-1;                }
      public void calc()        { sub();                       }
      public String toString()     { return "xA="+this.x;  }
}
public class B extends A  {
      private int x;
      public B()                    { super();   this.x = 1;              }
      public B(int x)               { super(x);  this.x =-x;              }
      public B(int xA, int xB)      { super(xA); this.x = xB;             }
      public int getX()             { return x;                           }
      public int baseX()            { return super.getX();                }
      public void tenTimesX()       { this.x = 10 * getX();               }
      public void tripleX()         { this.x = 3 * getX();                }
      public void sub()             { this.x = x-2;                       }
      public String toString()      { return super.toString()+" xB="+this.x+";";       }
}
```

/suite en page 25/

א. Voici une série d'instructions pour laquelle la sortie doit être :

xA=1 xB=20; xA=1 xB=20; xA=1 xB=20; xA=1 xB=20;

Cette série d'instructions comporte une erreur de compilation. Corrigez l'erreur afin que l'on obtienne la sortie correcte.

```
A a1 = new B(1, 20);
Object obj = a1;
B b1 = a1;
A a2 = a1;
System.out.println(a1+" "+obj+" "+a2+" "+b1);
```

ב. Voici une série d'instructions. Donnez les objets créés, et pour chaque objet, donnez les valeurs de ses attributs.

```
A aa = new B(3, 10);
aa.sub();
Object[] ar = new Object[6];
ar[0] = new A();
ar[1] = new A(5);
ar[2] = new B();
ar[3] = new B(5);
ar[4] = new B(2, 4);
ar[5] = aa;
((A)ar[3]).tripleX();
((B)ar[4]).tenTimesX();
```

(Attention : le paragraphe ג de la question se trouve à la page suivante)

/suite en page 26/

ג. Voici une méthode principale.

```
public static void main(String[] args)
{
    A a1 = new A(1);
    A a2 = new B(2, 99);
    //***
}
```

Voici les segments (i)-(vi)

(i)
```
a2.doubleX();
System.out.println(a2);
```

(ii)
```
a2.tenTimesX();
System.out.println(a2.tenTimesX());
```

(iii)
```
if (a2 instanceof B)
{
  a2.tenTimesX();
  System.out.println(a2);
}
```

(iv)
```
((B)a1).tenTimesX();
System.out.println(a1);
```

(v)
```
a2.calc();
System.out.println(a2);
```

(vi)
```
B bb = (B)a2;
System.out.println(bb.baseX());
```

Pour chacun des segments (i)-(vi) :

- Écrivez le segment à la place de //*** dans la méthode principale.
- Déterminez si le segment est valide ou non.
- Si le segment est valide, écrivez la sortie obtenue suite à son exécution.
  Si le segment n'est pas valide, écrivez s'il s'agit d'une erreur de compilation ou d'une erreur d'exécution.

/suite en page 27/

תכנות מונחה עצמים

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

15. Cette question comporte deux paragraphes, א-ב, qui ne sont pas liés. Répondez aux deux.

א. Soit cinq classes B , X , Y , Z , Run . Les sous-paragraphes (1)-(4) ci-dessous se rapportent à ces classes. Les sous-paragraphes ne sont pas liés les uns aux autres. Répondez à tous.

(1) Dans chacune des classes B , X , Y , Z a été définie la méthode
public virtual void Foo(){}
et dans la classe Run a été définie la méthode
public void Bar(Object g) {g.Foo(); }

Obtiendra-t-on une erreur de compilation ?
Si oui, expliquez pourquoi.

(2) Supposez que les classes X , Y , Z héritent de la classe B .
Dans chacune des classes X , Y , Z a été définie la méthode
public virtual void Foo() {}
et dans la classe Run a été définie la méthode
public void Bar(B g) {g.Foo(); }

Obtiendra-t-on une erreur de compilation ?
Si oui, expliquez pourquoi.

(3) Supposez que les classes X , Y , Z héritent de la classe B .
Dans chacune des classes B , X , Y , Z a été définie la méthode
public virtual void Foo(){}
et dans la classe Run a été définie la méthode
public void Bar(B g) {g.Foo(); }

Obtiendra-t-on une erreur de compilation ?
Si oui, expliquez pourquoi.

(4) Supposez que les classes X , Y , Z héritent de la classe B .
Dans chacune des classes B , X , Y , Z a été définie la méthode
public virtual void Foo(){}
et dans la classe Run a été définie la méthode
public void Bar(B g) {g.Foo(); }

Obtiendra-t-on une erreur de compilation ?
Si oui, expliquez pourquoi.

(Attention : le paragraphe ב de la question se trouve à la page suivante)

/suite en page 28/

ב. (Il n'y a pas de lien avec le paragraphe א)

Voici une classe **Singer** héritant de la classe **Artist** et l'interface **IPerform** que la classe **Singer** implémente.

```
interface IPerform
{
    void Act();
    int Train();
}
public class Singer : Artist , IPerform
{
    private int hits;
    public Singer(string name, double sal) : base(sal, name)
    {
        this.hits = 5;
    }
    public Singer(double sal, int hits) : base(sal, "Singer Name")
    {
        this.hits = hits;
    }
    public Singer(int hits) : base(6532.6, "Some", "One")
    {
        this.hits = hits;
    }
    public double Value()                {return this.hits * this.Price(); }
    public int GetNum()                  {return Artist.num;        }
    public override double Calc(double d)      {return d * base.Calc(10.2);       }
    public override void Print()         {base.Print(); Console.WriteLine("Singer");    }
    public void Act()                    {Console.WriteLine("I am singing");
}
```

Écrivez en C# la déclaration de la classe **Artist** , les attributs et les déclarations des méthodes requises de la classe **Singer** et de l'interface **IPerform** données. Il n'est pas nécessaire d'implémenter les méthodes de la classe **Artist** .

Ne modifiez pas la classe **Singer** et l'interface **IPerform** .

/suite en page 29/

16. Voici les classes A , B .

Les trois paragraphes א-ג se rapportent à ces classes, mais ne sont pas liés les uns aux autres. Répondez aux trois.

```
public class A    {
      private int x;
      public A()                        { this.x = 0;              }
      public A(int x)                   { this.x = x;              }
      public virtual int GetX()         { return x;                }
      public void DoubleX()             { this.x = 2 * GetX();     }
      public virtual void TripleX()     { this.x = 3 * GetX();     }
      public virtual void Sub()         { this.x = x−1;            }
      public void Calc()                { Sub();                   }
      public override string ToString(){ return "xA="+this.x;  }
}
public class B:A  {
      private int x;
      public B() : base()               { this.x = 1;              }
      public B(int x) : base(x)         { this.x =−x;              }
      public B(int xA, int xB) : base(xA)       { this.x = xB;     }
      public override int GetX()        { return x;                }
      public int BaseX()                { return base.GetX();      }
      public void TenTimesX()           { this.x = 10 * GetX();    }
      public override void TripleX()    { this.x = 3 * GetX();     }
      public override void Sub()        { this.x = x−2;            }
      public override string ToString(){ return base.ToString()+" xB="+this.x+";";}
}
```

(Attention : les paragraphes de la question se trouvent à la page suivante)

/suite en page 30/

א. Voici une série d'instructions pour laquelle la sortie doit être :

xA=1 xB=20; xA=1 xB=20; xA=1 xB=20; xA=1 xB=20;

Cette série d'instructions comporte une erreur de compilation. Corrigez l'erreur afin que l'on obtienne la sortie correcte.

```
A a1 = new B(1, 20);
Object obj = a1;
B b1 = a1;
A a2 = a1;
Console.WriteLine(a1+" "+obj+" "+a2+" "+b1);
```

ב. Voici une série d'instructions. Donnez les objets créés, et pour chaque objet, donnez les valeurs de ses attributs.

```
A aa = new B(3, 10);
aa.Sub();
Object[] ar = new Object[6];
ar[0] = new A();
ar[1] = new A(5);
ar[2] = new B();
ar[3] = new B(5);
ar[4] = new B(2, 4);
ar[5] = aa;
((A)ar[3]).TripleX();
((B)ar[4]).TenTimesX();
```

/suite en page 31/

ג. Voici une méthode principale.

```
public static void Main()
{
      A a1 = new A(1);
      A a2 = new B(2, 99);
      //***
}
```

Voici les segments (i)-(vi)

```
(i)   a2.DoubleX();
      Console.WriteLine(a2);
(ii)  a2.TenTimesX();
      Console.WriteLine(a2.TenTimesX());
(iii) if (a2 is B)
      {
       a2.TenTimesX();
       Console.WriteLine(a2);
      }
(iv)  ((B)a1).TenTimesX();
      Console.WriteLine(a1);
(v)   a2.Calc();
      Console.WriteLine(a2);
(vi)  B bb = (B)a2;
      Console.WriteLine(bb.BaseX());
```

Pour chacun des segment (i)-(vi) :

- Écrivez le segment à la place de //*** dans la méthode principale.
- Déterminez si le segment est valide ou non.
- Si le segment est valide, écrivez la sortie obtenue suite à son exécution.
  Si le segment n'est pas valide, écrivez s'il s'agit d'une erreur de compilation ou d'une erreur d'exécution.

Bonne chance!

בהצלחה!