



OFPPT

مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle et de la Promotion du Travail

Direction Recherche et Ingénierie de la Formation

Examen de passage à la 2^{ème} année

Session Juillet 2018

Filière : Techniques de Développement Informatique

Epreuve : Synthèse

Niveau : TS

Variante : V2

Durée : 5 heures

Barème : / 120 pts

Eléments de réponse

Partie I : Théorie

40 pts

Dossier 1 : L'essentiel en technologies de l'information

12 pts

1. 0.5 pt pour chaque conversion

6 pts

Binaire	Octal	Décimal	Hexadécimal
111100	74	60	3C
1011001	131	89	59
10000011	203	131	83
100110001	461	305	131

2.

a. La fonction f :

$$f(a, b, c, d) = \bar{a}b\bar{c} + \bar{a}bc + \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c$$

1.5 pts

Simplification :

$$f(a, b, c, d) = \bar{a}b\bar{c} + \bar{a}bc + a\bar{b}\bar{c} + a\bar{b}c$$

$$f(a, b, c, d) = \bar{a}b(\bar{c}+c) + a\bar{b}(\bar{c}+c)$$

1.5 pts

$$f(a, b, c, d) = \bar{a}b + a\bar{b}$$

b. Simplification moyennant le tableau de Karnaugh :

	$\bar{a}\bar{b}$	$\bar{a}b$	ab	$a\bar{b}$
$\bar{c}$	0	1	0	1
c	0	1	0	1

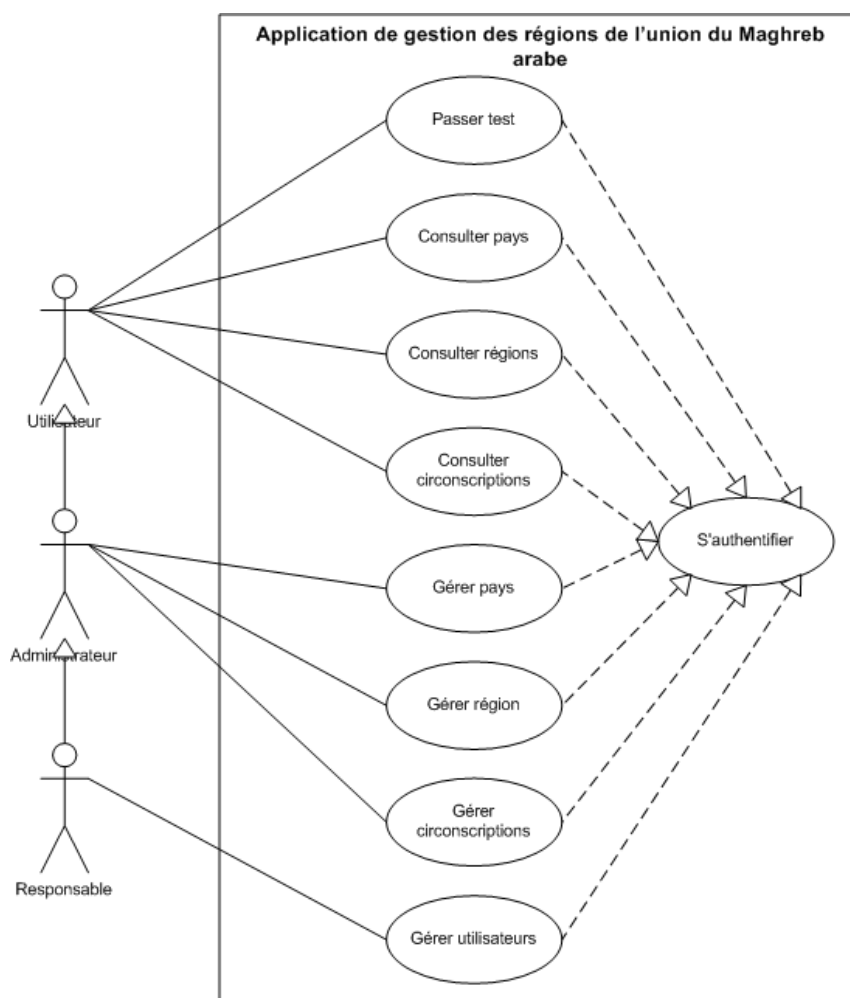
$$f(a, b, c, d) = \bar{a}b + a\bar{b}$$

3 pts

Filière	Epreuve	Session	1/9
TDI	Synthèse - V2 - Eléments de réponse	Juillet 2018	

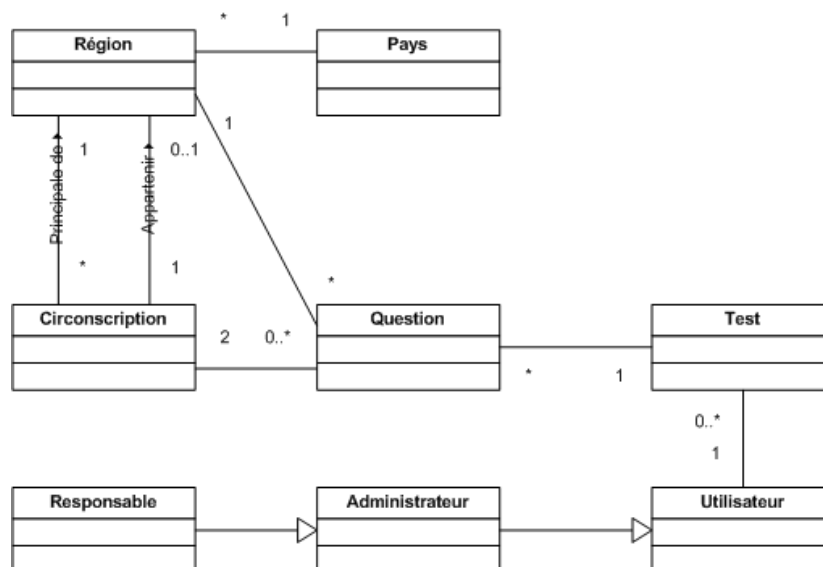
1. Diagramme de cas d'utilisation : 0.25 pt pour chaque élément (acteur, Cas d'utilisation et association)

7.5 pts



2. Diagramme de classes : 0.5 pt pour chaque élément (classe et association)

8.5 pts



Filière	Epreuve	Session	2 / 9
TDI	Synthèse - V2 - Eléments de réponse	Juillet 2018	

Algorithme	Gestion des Circonscriptions	
Structure	Circonscription	
	Nom : Chaîne	
	Population : Entier	1 pt
	Region : Chaîne	
FinStructure		
Constante	nMax $\leftarrow$ 20	
Variables	circonscriptions[nMax] : Circonscription	1 pt
	n, i	
	populationTotal,	
	populationMin,	
	indiceCirconscriptionPeuplee : Entier	
	nomRegion : Chaîne	
Début		
	Répéter	
	Ecrire("Nombre de circonscriptions ? ")	
	Lire(n)	1 pt
	TantQue(NON(n > 0 ET n <= nMax))	
	Pour i $\leftarrow$ 0 jusqu'à n - 1	
	Ecrire("Circonscription ", i + 1, " : \n")	
	Ecrire("Nom : ")	
	Lire(circonscriptions[i].Nom)	
	Ecrire("Population : ")	2 pts
	Lire(circonscriptions[i].Population)	
	Ecrire("Région : ")	
	Lire(circonscriptions[i].Region)	
	FinPour	
	Ecrire("Région dont vous voulez calculer la population totale : ")	
	Lire(nomPays)	
	populationTotale $\leftarrow$ 0	
	Pour i $\leftarrow$ 0 jusqu'à n - 1	
	Si(nomRegion = circonscriptions[i].Region)	3 pts
	populationTotale += circonscriptions[i].Population	
	FinSi	
	FinPour	
	Ecrire("La population totale est : ", populationTotale)	
	populationMin $\leftarrow$ 0	4 pts

Filière	Epreuve	Session	3/9
TDI	Synthèse - V2 - Eléments de réponse	Juillet 2018	

```

    Pour i ← 0 jusqu'à n - 1
        Si(populationMin > circonscriptions[i].Population)
            populationMin ← circonscriptions[i].Population
            indiceCirconscriptionPeuplee ← i
        FinSi
    FinPour
    Ecrire("Circonscription la moins peuplée : ",
        circonscriptions[indiceCirconscriptionPeuplee].Nom)
Fin

```

Partie II : Pratique

80 pts

Dossier 1 : Programmation structurée

16 pts

1.

- | | | | |
|----|----------------------|---|-------|
| a. | $f(387) \rightarrow$ | 1 | 2 pts |
| b. | $f(38) \rightarrow$ | 2 | 2 pts |
| c. | $f(3) \rightarrow$ | 3 | 2 pts |

2.

```

int estUniforme(int nombre)
{
    if(nombre < 0) return 0;
    else
    {
        char nombreToString[10], longueur, i;

        sprintf(nombreToString, "%d", nombre);

        longueur = strlen(nombreToString);

        for(i = 0; i <= longueur - 2; i++)
        {
            if(nombreToString[i] != nombreToString[i + 1])
                return 0;
        }

        return 1;
    }
}

```

3.

Filière	Epreuve	Session	4/9
TDI	Synthèse - V2 - Eléments de réponse	Juillet 2018	

```

void main()
{
    int nombre, nombreChiffres;

    //a. 2 pts
    do
    {
        printf("Donner un nombre qui appartient a l'intervalle [1, 100000[ : ");
        scanf("%d", &nombre);
    }while(!(nombre >= 1 && nombre < 100000));

    //b. 4 pts
    nombreChiffres = f(nombre);

    printf("%d appartient a l'intervalle [%.0f, %.0f[",
        nombre, pow(10, nombreChiffres - 1), pow(10, nombreChiffres));

    getch();
}

```

Dossier 2 : Programmation événementielle et orientée objet

64 pts

```

class Région
{
    //1. 1.5 pts
    private string _Nom;
    public string Nom
    {
        get { return _Nom; }
        set
        {
            //1.a 3 pts
            if (value.Length < 4)
                throw new Exception("Le nom doit contenir au moins 4 caractères");

            foreach (char caractère in value)
            {
                if (char.IsDigit(caractère))
                    throw new Exception("Le nom ne peut pas contenir des chiffres");
            }

            _Nom = value;
        }
    }

    //1.(suite) 1.5 pts
    private List<Circonscription> _Circonscriptions = new List<Circonscription>();
    public List<Circonscription> Circonscriptions
    {
        get { return _Circonscriptions; }
        set { _Circonscriptions = value; }
    }

    //1.(suite) 1.5 pts
    private Circonscription _CirconscriptionPrincipale;
    public Circonscription CirconscriptionPrincipale
    {
        get { return _CirconscriptionPrincipale; }
        set

```

Filière	Epreuve	Session	5/9
TDI	Synthèse - V2 - Eléments de réponse	Juillet 2018	

```

{
    //1.b
    Ajouter(value);
    //La méthode vérifie l'existence de la circonscription

    _CirconscriptionPrincipale = value;
}
}

//2.a
public Région(string nom)
{
    Nom = nom;
}

//2.b
public Région(string nom, Circonscription circonscriptionPrincipale)
{
    Nom = nom;
    CirconscriptionPrincipale = circonscriptionPrincipale;
}

//3
public int NombreCirconscriptions
{
    get
    {
        return Circonscriptions.Count;
    }
}

//4
public int Rechercher(String nomCirconscription)
{
    foreach (Circonscription circonscription in Circonscriptions)
        if (circonscription.Nom == nomCirconscription)
            return Circonscriptions.IndexOf(circonscription);

    return -1;
}

//5
public void Ajouter(Circonscription circonscription)
{
    if (Rechercher(circonscription.Nom) == -1)
        Circonscriptions.Add(circonscription);
}

//6
public void Supprimer(String nomCirconscription)
{
    int indiceCirconscription = Rechercher(nomCirconscription);
    if (indiceCirconscription != -1)
    {
        Circonscriptions.RemoveAt(indiceCirconscription);
        if (CirconscriptionPrincipale.Nom == nomCirconscription)
            CirconscriptionPrincipale = null;
    }
}

//7
public List<Circonscription> TrierParPopulation(string ordre)

```

3 pts

1.5 pts

2 pts

2 pts

3 pts

3 pts

4 pts

5 pts

Filière	Epreuve	Session	6/9
TDI	Synthèse - V2 - Eléments de réponse	Juillet 2018	

```

{
    List<Circonscription> circonscriptions = new List<Circonscription>();

    circonscriptions.AddRange(Circonscriptions);

    if (ordre.ToLower() == "croissant")
        circonscriptions.Sort(new PopulationCroissantComparer());
    else if (ordre.ToLower() == "décroissant")
        circonscriptions.Sort(new PopulationDécroissantComparer());

    return circonscriptions;
}

//8
public int PopulationTotale()
{
    int populationTotale = 0;

    foreach (Circonscription circonscription in Circonscriptions)
        populationTotale += circonscription.Population;

    return populationTotale;
}
}

//7(suite)
class PopulationCroissantComparer : IComparer<Circonscription>
{
    public int Compare(Circonscription x, Circonscription y)
    {
        return x.Population.CompareTo(y.Population);
    }
}

//7(suite)
class PopulationDécroissantComparer : IComparer<Circonscription>
{
    public int Compare(Circonscription x, Circonscription y)
    {
        return -x.Population.CompareTo(y.Population);
    }
}
}

-----

public partial class Gestion_des_Régions : Form
{
    //9.a
    List<Région> LesRégions = new List<Région>();

    public Gestion_des_Régions()
    {
        InitializeComponent();
    }

    //9.b
    private void Gestion_des_Régions_Load(object sender, EventArgs e)
    {
        //9.b.i
        Région région1 = new Région("Casablanca-Settat");
        région1.Ajouter(new Circonscription("Préfecture de Mohammédia", 322286));
        région1.Ajouter(new Circonscription("Préfecture de Casablanca", 2949805));
        région1.Ajouter(new Circonscription("Province de Nouaceur", 236119));
    }
}

```

Filière	Epreuve	Session	7/9
TDI	Synthèse - V2 - Eléments de réponse	Juillet 2018	

```

région1.Ajouter(new Circonscription("Province de Médiouna", 122851));
région1.Ajouter(new Circonscription("Marrakech", 928850));
région1.CirconscriptionPrincipale = région1.Circonscriptions[1];
LesRégions.Add(région1);

Région région2 = new Région("Marrakech-Safi");
LesRégions.Add(région2);

Région région3 = new Région("Laâyoune-Sakia El Hamra");
région3.CirconscriptionPrincipale =
    new Circonscription("Province de Laâyoune", 210023);
région3.Ajouter(new Circonscription("Province de Boujdour", 46129));
région3.Ajouter(new Circonscription("Province de Tarfaya", 10410));
LesRégions.Add(région3);

//9.b.ii 3 pts
comboBox_Régions.DataSource = LesRégions;
comboBox_Régions.DisplayMember = "Nom";
}

//9.c 4 pts
private void comboBox_Région_SelectedIndexChanged(object sender, EventArgs e)
{
    Région Région = comboBox_Régions.SelectedItem as Région;

    listBox_Circonscriptions.Items.Clear();

    if (Région.NombreCirconscriptions != 0)
    {
        foreach (Circonscription Circonscription in Région.Circonscriptions)
        {
            //9.c.i 2 pts
            if (Région.CirconscriptionPrincipale != null &&
                Circonscription.Nom == Région.CirconscriptionPrincipale.Nom)
                listBox_Circonscriptions.Items.Add("--> " + Circonscription);
            else
                listBox_Circonscriptions.Items.Add(Circonscription);
        }
    }
    //9.c.ii 2 pts
    else
        listBox_Circonscriptions.Items.Add("Aucune circonscription !!!");
}

//9.d 4 pts
private void button_Trier_Click(object sender, EventArgs e)
{
    Région Région = comboBox_Régions.SelectedItem as Région;

    if (Région.NombreCirconscriptions != 0)
    {
        string ordre = "croissant";
        List<Circonscription> Circonscriptions;

        if (radioButton_Décroissant.Checked) ordre = "décroissant";

        Circonscriptions = Région.TrierParPopulation(ordre);

        listBox_Circonscriptions.Items.Clear();

        foreach (Circonscription Circonscription in Circonscriptions)

```

Filière	Epreuve	Session	8/9
TDI	Synthèse - V2 - Eléments de réponse	Juillet 2018	

```

        {
            if (Région.CirconscriptionPrincipale != null &&
                Circonscription.Nom == Région.CirconscriptionPrincipale.Nom)
                listBox_Circonscriptions.Items.Add("--> " + Circonscription);
            else
                listBox_Circonscriptions.Items.Add(Circonscription);
        }
    }
}

//9.e 4 pts
private void button_OrdreInitial_Click(object sender, EventArgs e)
{
    comboBox_Région_SelectedIndexChanged(comboBox_Régions, new EventArgs());
}

//9.f 3 pts
void Gestion_des_Régions_FormClosing(object sender, FormClosingEventArgs e)
{
    if (MessageBox.Show("Voulez-vous vraiment fermer l'application ?",
        "Attention",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Question) == DialogResult.No)
        e.Cancel = true;
}
}

```

Filière	Epreuve	Session	9/9
TDI	Synthèse - V2 - Eléments de réponse	Juillet 2018	