***Direction Recherche et Ingénierie de la Formation***

**Examen de Passage(Correction)**

**Session Juillet 2015**

|  |  |
| --- | --- |
| ***Filière : Techniques de Développement Informatique*** | ***Epreuve : Synthèse*** |
| ***Niveau: TS*** | ***Variante: V2*** |
| ***Durée : 5 heures*** | ***Barème : / 120*** |

* **Partie I : Théorie**
* **Dossier 1: Notions de mathématiques appliquées à l’informatique**
* **Exercice 1: Conversion numérique**

Remplir le tableau suivant:

|  |  |  |  |
| --- | --- | --- | --- |
| Décimal | Binaire | Octal | Hexadécimal |
| **2379** | **100101001011** | **4513** | **94B** |
| **219** | **11011011** | **333** | **DB** |
| **860** | **1101011100** | **1534** | **35C** |
| **318** | **100111110** | **476** | **13E** |

* **Exercice 2: Algèbre de Boole**

1. Simplifier analytiquement la fonction logique ***Q***.
2. Construire la table de vérité.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***A*** | ***B*** | ***C*** | ***F*** |  |
| 0 | 0 | 0 | 0 |  | |
| 0 | 0 | 1 | 0 |  | |
| 0 | 1 | 0 | 1 |  | |
| 0 | 1 | 1 | 1 |  | |
| 1 | 0 | 0 | 0 |  | |
| 1 | 0 | 1 | 1 |  | |
| 1 | 1 | 0 | 1 |  | |
| 1 | 1 | 1 | 1 |  | |
|  |  |  |  |  | |
|  |  |  |  |  | |
|  |  |  |  |  | |
|  |  |  |  |  | |
|  |  |  |  |  | |
|  |  |  |  |  | |
|  |  |  |  |  | |
|  |  |  |  |  | |

1. Simplifier avec la méthode de Karnaugh la fonction logique ***f***.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***F*** |  |  |  |  |  |
|  | ***AB***  ***C*** | 00 | 01 | 11 | 10 |
|  | 0 | **0** | **1** | **1** | **0** |
|  | 1 | **0** | **1** | **1** | **1** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

* **Dossier 2: Techniques de programmation structurée**
* **Exercice 1: Résultat de l’algorithme**

**#O#O#O#O**

**O#O#O#O#**

**#O#O#O#O**

**O#O#O#O#**

**#O#O#O#O**

**O#O#O#O#**

**#O#O#O#O**

**O#O#O#O#**

* **Exercice 2: Suppression des singnes de ponctuation**

Tableau T() : Caractère

Variables N,i : Entier

Début

Répéter

écrire ("Donner la taille du tableau: " )

lire (N)

Pour i=1 à N

Lire (T(i))

Fin Pour

**SupSignes** **( T,N)**

Fin

Procédure **SupSignes** (T : tableau() ; N : Entier )

i=1

Répéter

Si( t(i) = ‘,’ ou t(i)=’;’ ou t(i)=’.’ ou t(i) = ‘:’ ou t(i)=’?’ ou t(i)=’ !’) alors

Pour j=i j<n

T(j)🡨t(j+1)

FinPour

N🡨N-1

FinSi

Jusqu’à (i<=N)

Redim t(N)

Fin

* **Dossier 3: Conception et modélisation d'un système d'information**

**Conception d’un système d’information pour une société informatique**

**MCD**



**MLD**

****

* **Partie II: Pratique**
* **Dossier 1: Langage de programmation structurée**

#include<stdio.h>

#include <string.h>

#include <stdlib.h>

#define max 50

///===============les declaration=============

int n,choix;

struct Quartier {

char nom[20];

int surface;

int nbrhabitant;

};

struct Quartier les\_quartiers[max];

//=========procedure de saisie==============

void saisie\_un\_quartier(){

printf("Saisir les information du quartier : \n\n");

printf("Nom du quartier : ");

scanf("%s",&les\_quartiers[n+1].nom);

printf("Surface : ");

scanf("%d",&les\_quartiers[n+1].surface);

printf("Nombre d'habitants : ");

scanf("%d",&les\_quartiers[n+1].nbrhabitant);

n=n+1;

}

//=========procedure d'affichage==============

void afficher\_les\_quartiers(){

int i;

for(i=1;i<n+1;i++)

printf("Le quartier :%s & Sa surface : %d & son nombre d'habitants : %d \n",les\_quartiers[i].nom,les\_quartiers[i].surface,les\_quartiers[i].nbrhabitant);

}

//=========procedure de recheche==============

void rechercher\_quartier(){

int i;

printf("\nRechercher un quartier :\n");

int nbrh;

printf("Entrer le nombre d'habitants : ");

scanf("%d",&nbrh);

for(i=1;i<=n;i++){

if(les\_quartiers[i].nbrhabitant<= nbrh){

printf("----------------------------------\n");

printf("Nom : %s\n",les\_quartiers[i].nom);

printf("Surface : %d\n",les\_quartiers[i].surface);

printf("Nombre d’habitants : %d\n",les\_quartiers[i].nbrhabitant);

}

}

}

//=========procedure de suppression==============

void supprimer\_quartier(){

char nom[20];

int i,j,k;

printf("Saisir le nom du quartier que vous voulez supprimer : ");

scanf("%s",&nom);

for(i=1;i<=n;i++){

if(strcmp(les\_quartiers[i].nom,nom)==0){

k=i;

n=n-1;

for(j=k;j<=n;j++){

les\_quartiers[j]=les\_quartiers[j+1];

}

}

}

}

//=========Sauvegarder dans un fichier==============

void sauvegarder()

{int i;

char nomf[20];

printf("Saisir le nom du fichier: ");

scanf("%s",&nomf);

FILE\* fichier = NULL;

fichier = fopen(nomf,"w");

if (fichier != NULL)

{

for(i=1;i<=n;i++)

{

fprintf(fichier,"%s/%d/%d \n",les\_quartiers[i].nom,les\_quartiers[i].surface,les\_quartiers[i].nbrhabitant);

}

fclose (fichier);

}

else

printf("Impossible d'ouvrir le fichier test.txt");

}

}

//=========Quitter==============

void quitter(){

system("cls");

printf("\n\n \t------------FIN-----------\n\n\n");

printf("\n\n \t------------MERCI-----------\n\n\n");

}

**Main**

int main ()

{

do{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*Gestion des quartiers\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n");

printf(" <1. Saisir un quartier---------------------------- \n");

printf(" <2. Afficher la liste des quartiers-------------- \n");

printf(" <3. Rechercher un quartier ---------------------\n");

printf(" <4. Supprimer un quartier-----------------------\n");

printf(" <5. Sauvegarder---------------------------------- \n");

printf(" <6. Quitter----------------------------------------- \n");

printf(" ====== \n");

printf("Entre votre choix : ");

scanf("%d",&choix);

printf("\n\n");

switch(choix){

case 1: system("cls");saisie\_un\_quartier();break;

case 2: system("cls");afficher\_les\_quartiers();break;

case 3: system("cls");rechercher\_quartier();break;

case 4: system("cls");supprimer\_quartier();break;

case 5: system("cls");sauvegarder();break;

case 6:system("cls");quitter();break;

default : system("cls");printf("votre choix n'est pas valide !!!\n");

}

}while(choix!=6);

system("PAUSE>null");

return 0;

}

* **Dossier 2: Programmation orientée objet**

**Développement d’une application orientée objet pour la gestion d’un magasin**

**Classe “Article”**

class Article

{

private string nserie;

private float pht;

private int qte\_stk;

private int qte\_min;

//public Article ();

public Article(string nserie, float pht, int qte\_stk, int qte\_min)

{

this.nserie = nserie;

this.pht = pht;

this.qte\_stk = qte\_stk;

this.qte\_min = qte\_min;

}

public new string ToString()

{

return this.nserie +" "+this.pht +" "+ this.qte\_stk +" "+ this.qte\_min ;

}

public void Sapprovisionner (int qte)

{

this.qte\_stk=this.qte\_stk+qte;

}

public void Achat(int qte)

{

this.qte\_stk = this.qte\_stk - qte;

if (this.qte\_stk < this.qte\_min)

Console.WriteLine("Attention! il faut s'approvisionner");

}

}

**Classe « Habit »**

class Habit : Article

{

private int taille;

private string couleur;

public Habit (string nserie, float pht, int qte\_stk, int qte\_min,int taille,string couleur)

:base( nserie, pht, qte\_stk, qte\_min)

{

this.taille = taille;

this.couleur = couleur;

}

public new string ToString()

{

return base.ToString() + " " + this.taille + " " + this.couleur;

}

}

**Classe « Electroménager »**

class Electromenager : Article

{

private float poids;

private int duree;

public Electromenager(string nserie, float pht, int qte\_stk, int

qte\_min, float poids, int duree)

: base(nserie, pht, qte\_stk, qte\_min)

{

this.poids = poids;

this.duree = duree;

}

public DateTime datefinGarantie()

{

DateTime res = DateTime.Now.AddMonths(duree);

return (res);

}

public Boolean Sousgarantie(DateTime date\_achat)

{

DateTime d = DateTime.Now;

if (date\_achat.AddMonths(duree) > d)

return true;

else return false;

}

public new string ToString()

{

return base.ToString() + " " + this.poids + " " + this.duree;

}

}

class Program

{

static void Main(string[] args)

{

Television t1 = new Television("tv123", 80,13.50F, 3500, "lcd", 100);

Console.WriteLine(t1.ToString());

Console.WriteLine(t1.ClasseEneergétique());

VeloElec v1 = new VeloElec("v122", 250, 25, 11500, 50.50F, 560);

v1.Rouler(20.50F);

v1.Charger(120);

Console.WriteLine(v1.ToString());

Console.ReadLine();

}

}

* **Dossier 3: Programmation événementielle**

1. le code permettant de remplir la liste

comboBox1.Items.Add("Studio");

comboBox1.Items.Add("Appartement");

comboBox1.Items.Add("Maison");

1. le code du bouton "**Nouveau** » :

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

comboBox1.Text = "";

textBox4.Text = "";

1. le code du bouton " **Ajouter**» :

dataGridView1.Rows.Add(textBox1.Text, textBox2.Text,

textBox3.Text,comboBox1.Text, textBox4.Text );

1. le code du bouton " **Supprimer**» :

if (MessageBox.Show( "voulez vous supprimer?","Suppression",MessageBoxButtons.YesNo) == DialogResult.Yes)

{

foreach (DataGridViewRow dr in dataGridView1.Rows)

{

if (dr.Cells[0].Value.ToString() == textBox1.Text)

{

dataGridView1.Rows.Remove(dr);

break;

}

}

}

5. le code du bouton " **Sauvegarder**» :

.BinaryFormatter f = new BinaryFormatter();

FileStream fs = new FileStream("Logements.dat", FileMode.OpenOrCreate, FileAccess.Write);

f.Serialize(fs, Logements);

fs.Close();