

Annexe3 : Anciens DS et Examens Corrigés

Examen Janvier 2015**Examen**Matière: **Langage de programmation**Documents: **Non autorisés**Enseignants : **S.Boubaker; M.Ben Romdhan; H.Haddaji; M.Khemiri;**Durée: **1h00'**Classes : **TI.11, 12,13, 14, 15, 16**Date: **Janvier 2015***Il vous est demandé d'apporter un soin particulier à la présentation de votre copie.***EXERCICE 1 : LES TABLEAUX****6 Pts**

Ecrire un programme C qui permet de saisir N éléments d'un tableau T d'entiers (au maximum 50) et d'une matrice M a deux dimensions LxC (dimensions maximales: 10 lignes et 10 colonnes) et d'afficher si tous les éléments du tableau T figurent dans la matrice M ou non.

Exemple 1:

Tableau T:

2	10	20	13	15
---	----	----	----	----

Matrice M:

13	11	4	6	7
2	20	10	23	88
14	15	55	5	32
0	32	33	19	1

Résultat: Tous les éléments du tableau T existent dans la matrice M.

Exemple 2:

Tableau T:

2	10	20	13	15
---	----	----	----	----

Matrice M:

1	11	4	6	7
2	26	10	23	88
14	15	55	5	32
0	32	33	19	1

⇒ **Résultat:** 3 éléments du tableau T existent dans la matrice M.

EXERCICE 2 : LES CHAINES DE CARACTERES**5 Pts**

Ecrire un programme qui lit deux chaînes de caractères CH1 et CH2 au clavier et qui vérifie l'existence de la chaîne CH2 dans CH1.

Exemples:

- CH1="Alphonse "
CH2="phon" ==> CH2 est une sous chaîne de CH1
- CH1="totalement"
CH2="t" ==> CH2 est une sous chaîne de CH1

- CH1="abacab"
- CH2="aa" ==> CH2 n'existe pas dans CH1

EXERCICE 3 : LES FONCTIONS

9 PTS

Un registre peut être représenté par un tableau à une dimension contenant N valeurs binaires.

Par exemple:

- R1 =

1	1	1	0	1	0	1	1
---	---	---	---	---	---	---	---

, registre de taille 8.

- R2 =

1	1	1	0	0	0	1	1	1	0	1	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 registre de taille 16.

Les registres offrent les opérations suivantes :

- L'opération **ET** réalise un *et logique* bit à bit entre les bits de même position de deux registres. Par exemple,

R1 =

1	1	1	0	1	0	1	1
---	---	---	---	---	---	---	---

R2 =

1	1	1	0	0	0	1	1	1	0	1	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

R1 ET R2 =

0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

- L'opération **OU** réalise un *ou logique* bit à bit entre les bits de même position de deux registres. Par exemple,

R1 =

1	1	1	0	1	0	1	1
---	---	---	---	---	---	---	---

R2 =

1	1	1	0	0	0	1	1	1	0	1	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

R1 OU R2 =

1	1	0	0	0	1	1	1	1	1	1	1	1	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

- L'opération **SHIFT_GAUCHE** permet de réaliser un décalage vers la gauche de n positions avec introduction de zéros du côté droit. Par exemple,

R1 =

1	1	1	0	1	0	1	1
---	---	---	---	---	---	---	---

R1 SHIFT_GAUCHE 3 =

0	1	0	1	1	0	0	0
---	---	---	---	---	---	---	---

Réaliser les fonctions suivantes:

- 1) **void SaisirRegistre(int R[],int N):** permettant de charger le registre R par N bits.
- 2) **void AfficherRegistre(int R[],int N):** permettant d'afficher les N bits du registre R.
- 3) **void ET(int R1[],int N1,int R2[],int N2):** permettant d'afficher le et logique entre R1 de taille N1 et R2 de taille N2.
- 4) **void OU(int R1[],int N1,int R2[],int N2):** permettant d'afficher le ou logique entre R1 de taille N1 et R2 de taille N2.
- 5) **void SHIFT_GAUCHE (int R[],int N,int D):** permettant de faire un décalage à gauche de D bits dans le registre R composé par N bits.
- 6) En utilisant ces fonctions, préparer la fonction principale (**main**) permettant de saisir les deux registres R1 (de taille 8) et R2 (de taille 16), de décaler à gauche 5 bits dans R2 et d'afficher le OU logique entre R1 et R2.

DS Novembre 2017**Devoir Surveillé**

Matière: Langage de programmation

Documents: Non autorisés

Enseignants: BenRomdhanM,Challouf S,FerjaniN,Fkiri A,Haddaji H

Durée: 1h00'

Classes : TI.11, 12,13, 14, 15, 16

Date:08/11/2017

*Il vous est demandé d'apporter un soin particulier à la présentation de votre copie.***EXERCICE 1 :****6 PTS**1) Quelles sont les valeurs des expressions suivantes, pour `int i=1,j=4` ; ?

- a) `(float)i/j`
- b) `(float)(i/j)`
- c) `(float)i/(float)j`

2) Qu'affiche le programme suivant ?

```
#include<stdio.h>
int main()
{
    int n=3;
    n++;
    if (n > 3) {
        printf("n=%d\n", --n);
    } else {
        printf("n-2=%d\n", n-2);
    }
    return 0;
}
```

3) Soient les déclarations suivantes :

```
int n = 5, p = 9 ;
int q ;
float x ;
```

Quelle est la valeur affectée aux différentes variables concernées par chacune des instructions suivantes ?

- a) `q = n < p ;`
- b) `q = n == p ;`
- c) `q = p % n + p > n ;`
- d) `x = p / n ;`
- e) `x = (p + 0.5) / n ;`
- f) `x = (int)(p + 0.5) / n ;`
- g) `q = n * (p > n ? n : p) ;`

EXERCICE 2 :**5 PTS**

Chercher dans le programme C suivant les erreurs qui peuvent causer son dysfonctionnement, et proposer une correction :

```
#include<studio.h>
#define PI=3.14
int main()
{
    Float rayon,Périm,long;
    printf("Donner le rayon du cercle:");
    do
    {
        scanf("%f",&rayon);
        printf("Donner PI:");
    }while (PI!=3.14)
    scanf("%f",&PI);
    //calcul du diamètre de cercle
    long=2*rayon;
    //calcul du périmètre
    Périm=long*PI;
    printf("Diamètre=",long,"\n Périmètre=", Périm);
    switch (Périm)
    {
        case 50: printf ("Cercle 50"); break;
        case 100: printf ("Cercle 100");
    default: printf ("Autre cercle");
    }
    return 0;
}
```

EXERCICE 3 :**4 PTS**

Ecrire un programme qui pour une valeur de **N** choisie au clavier (**N>0**), affiche **N** fois la valeur **N** sur la **N**^{ème} ligne.

Par exemple, pour **N = 3**, le programme affiche les messages suivants :

```
1
2 2
3 3 3
```

EXERCICE 4 :**5 PTS**

Soit la suite numérique définie comme suit:

$$U_0=1$$

$$U_n=2*U_{n-1}$$

Ecrire le programme C qui réalise les traitements suivants:

1. Pour un entier **n** donné strictement positif, calculer et afficher le **n**^{ème} terme de cette suite.
2. Calculer la valeur **n** du premier terme U_n tel que pour un entier **p** donné

$$U_n \leq p < U_{n+1} \quad (\text{la valeur de } p \text{ est entrée au clavier avec } p > 0).$$

Par exemple : pour $p = 5$, $U_2 = 4$ et $U_3 = 8$ donc $n = 2$.

Bon Courage

Examen Janvier 2018**EXAMEN**Matière: **LANGAGE DE PROGRAMMATION**

Enseignants: Ben Romdhan M, Challouf S, Ferjani N, Fkiri A, Haddaji H

Classes : TI.11, 12, 13, 14, 15, 16

Documents: **Non autorisés**Durée: **1h00'**

Date: /01/2018

*Il vous est demandé d'apporter un soin particulier à la présentation de votre copie.***EXERCICE 1 :****6 PTS**

Ecrire un programme C qui permet de lire à partir du clavier une suite de noms ne comportant pas d'espaces et d'afficher le nombre de noms ayant plus de dix caractères. La suite se termine lorsqu'on trappe le mot "FIN".

On supposera que chaque nom ne dépasse pas 20 caractères.

NB : Vous pouvez utiliser les fonctions de traitement des chaînes de caractères.

Exemple : Pour la suite de noms suivants :

Mohamed Ali
Chokri
Fatma
Safia
Salaheddine
FIN

⇒ On a 2 noms qui ont plus de 10 caractères.

EXERCICE 2 :**4 PTS**

Soit le programme C suivant :

```
#include <stdio.h>
void Aire(float A, float B, float C)
{
    C = A * B;
}
void main()
{
    float Large=10, Longueur=20, P=0;
    Aire(Large, Longueur, P);
    Printf(«Le périmètre du rectangle est %f » ,P) ;
}
```

1. Donnez les valeurs des variables Large, Longueur et P en sortie.
2. Quelles modifications doit-on apporter à la fonction et au programme principal pour qu'après l'appel de la fonction **Aire**, la variable P contient l'aire du rectangle.

PROBLEME :**10 Pts**

On souhaite modéliser des opérations permettant de gérer les articles dans un stock de magasin à l'aide de tableaux à une dimension.

On considère pour cela un tableau *TabArt* de chaînes de caractères où les éléments représentent les codes des articles (10 caractères maximum), un tableau *TabStock* d'entiers représentant la quantité en stock de chaque article, et un tableau *TabPrix* de réels représentant les prix des articles.

Ainsi, un élément à l'indice *i* du tableau *TabStock* représente la quantité en stock de l'article de la case *i* du tableau *TabArt* et qui a comme prix la case *i* du tableau *TabPrix*.

On suppose que les tableaux ont une capacité maximale définie par une constante *max* donnée (*max* =100).

1. Ecrire une fonction *void LireArticles (char TabArt [max][10], int TabStock[], float TabPrix, int N)* permettant de remplir un tableau de chaînes de caractères *TabArt* de *N* articles, ainsi que les tableaux *TabStock* et *TabPrix* (pour chaque article lire son code, sa quantité en stock et son prix).
2. Ecrire une fonction *void AfficherArticles (char TabArt[max][10], int TabStock[], float TabPrix[], int N)* permettant d'afficher la liste des articles avec leurs quantités en stock et leurs prix.
3. Ecrire une fonction *int QuantitéStock(char TabArt[max][10], int TabStock[], char codeart[], int N)* permettant de rechercher la quantité en stock d'un article donné (*codeart*) et retourne sa quantité en stock s'il existe et -1 sinon.
4. Créer une procédure *void AjouterArticle (char TabArt[max][10], int Tabstock[], float TabPrix[], char codeart[], int qs, float px, int *N)* permettant d'ajouter un article (*codeart*). Cette opération consiste à ajouter le code de l'article *codeart* à la fin du tableau *TabArt*, la quantité en stock *qs* dans le tableau *TabStock* et le prix *px* dans le tableau *TabPrix*.
5. Ecrire une fonction *void ApprovisionnerStock(char TabArt[max][10], int TabStock[], char codeart[], int qs, int N)* permettant d'approvisionner le stock d'un article en ajoutant une quantité donnée au stock existant. Cette fonction effectue une recherche de code de l'article dans le tableau *TabArt*, s'il existe alors elle ajoute la quantité *qs* au stock correspondant dans le tableau *TabStock*, s'il n'existe pas elle affiche un message d'erreur.
6. Ecrire une fonction *void MaxPrix(char TabArt[max][10], float TabPrix[], int N)* qui permet d'afficher le code et le prix de l'article le plus cher.
7. Ecrire une fonction *void SupprimerArticle (char TabArt[max][10], int TabStock[], float TabPrix[], char code[], int *N)* permettant de supprimer un article stocké dans le tableau *TabArt* en fournissant code. Sa suppression entraîne la suppression de sa quantité en stock dans le tableau *TabStock* et de son prix du tableau *TabPrix*.
8. Ecrire un programme principale *main()* qui permet de :

- a) Remplir les tableaux TabArt, TabPrix et TabStock.
- b) Mettre à jour le stock d'un article donné par son code CD1 avec la quantité en stock *QTI* donnée.
- c) Ajouter un article dont le code CD2, la quantité QT2 et le prix PX sont données par l'utilisateur.
- d) Supprimer les articles dont la quantité en stock est égale à 0.
- e) Afficher tous les articles (code, prix et quantité).

Bon Travail