

LIFAP1 – CC mi-parcours – Séquence 5

Contrôle Continu (Durée totale : 1h)

Judi 14 novembre 2024

NOM

PRENOM

Numéro Etudiant

Groupe TD

Recommandations : Les documents, calculatrice, téléphone portable sont interdits. La qualité de l'écriture et de la présentation seront prises en compte dans la note finale. Vous veillerez à **respecter** les notations et les règles d'écriture des algorithmes vues en cours et en TD. Un soin tout particulier devra être apporté à l'écriture des entêtes des différents sous-programmes.

Partie A – Algorithmique

Conversion décimal → Binaire

Nous allons dans cet exercice convertir un entier n positif strictement inférieur à 256 en nombre binaire codé sur un octet (= 8 bits). Pour y parvenir nous allons successivement diviser n par des puissances de 2. Le quotient nous indiquera si ce bit vaut 1 ou 0 et le reste nous permettra de continuer le processus.

Exemple : pour $n = 155$

$$155 / 2^7 = 1 \text{ reste } 27$$

$$27 / 2^6 = 0 \text{ reste } 27$$

$$27 / 2^5 = 0 \text{ reste } 27$$

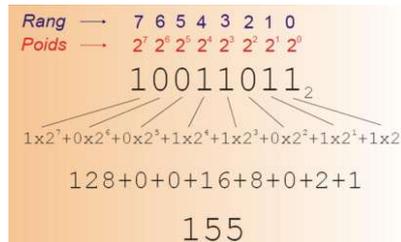
$$27 / 2^4 = 1 \text{ reste } 11$$

$$11 / 2^3 = 1 \text{ reste } 3$$

$$3 / 2^2 = 0 \text{ reste } 3$$

$$3 / 2^1 = 1 \text{ reste } 1$$

$$1 / 2^0 = 1 \text{ reste } 0$$



On construira le tableau suivant

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

1. Ecrire l'algorithme d'une **procédure** SaisieOctet qui demande à l'utilisateur et "retourne" un entier n compris entre 0 et 255 inclus. On recommencera la saisie tant que la valeur n proposée n'est pas dans $[0 ; 255]$.

Procédure SaisieOctet (val : entier)

Précondition : aucune

Donnée : aucune

Donnée / résultat : val

Description : saisit et retourne un entier entre 0 et 255

Variable locale : aucune

Début

Faire

Afficher("Donnez une valeur entre 0 et 255 ")

Saisir(val)

Tant que (val < 0 ou val > 255)

Fin

2. Ecrire l'algorithme d'un sous-programme `AfficheTab` qui affiche le contenu d'un tableau `T` de 8 entiers.

procédure `AfficheTab`(`T` tableau[8] d'entiers)

Précondition : aucune

Donnée : aucune

Donnée / résultat : `T`

Description : affiche le tableau `T`

Variable locale : `i` : entier

Début

 Pour `i` allant de 0 à 7 par pas de 1 faire

`Afficher(T[i])`

 Fin pour

Fin

3. Ecrire l'algorithme d'un sous-programme `Decompose` qui décompose en binaire un entier `n` compris entre 0 et 255 donné en décimal. Ce sous-programme remplira et "retournera" un tableau `T` de 8 entiers en faisant des divisions successives par les puissances de 2 comme expliqué précédemment. On pourra utiliser la fonction `puissance(a, b)` qui retourne a^b .

procédure `Decompose`(`T` tableau [8] d'entiers, `n` : entier)

Précondition : `n` compris entre 0 et 255

Donnée : aucune

Donnée / résultat : `T`

Description : remplit le tableau `T` avec les 8 entiers correspondant à la traduction binaire de `n`

Variable locale : `i`, `puis` : entiers

Début

 Pour `i` allant de 0 à 7 par pas de 1 faire

`puis` \leftarrow `puissance(2,7-i)`

`T[i]` \leftarrow `n / puis`

`n` \leftarrow `n modulo puis`

 Fin pour

Fin

4. Ecrire l'algorithme du programme principal qui saisit un entier dans [0 ; 255], décompose ce nombre en binaire puis affiche le tableau obtenu. On utilisera les 3 sous-programmes précédents.

Debut

```
Val : entier
T tableau [MAX] d'entiers
SaisieOctet(val)
Decompose(T,val)
AfficheTab(T)
```

Fin

Partie B – Langage C/C++

On souhaite faire des statistiques à l'entrée du parc de la Tête d'Or un dimanche matin pour voir quelle est l'activité préférée des Lyonnais. On va donc recenser les groupes de marcheurs, de coureurs et de cyclistes.

- 1- Déclarer une constante MAX ayant pour valeur 100.

```
const int MAX = 100 ou #define MAX 100
```

- 2- Écrire en C++ une **fonction** `RemplitAlea` qui remplit un tableau T de taille MAX*2 sur les Nb premières colonnes. Ce tableau contiendra sur sa première ligne le nombre de personnes dans le groupe (entre 1 et 5 personnes) et sur la deuxième ligne l'activité pratiquée (1 : Marche, 2 : Course et 3 : Vélo). Nb, le nombre de groupes recensés (nombre de colonnes remplies) sera un entier choisi aléatoirement entre 10 et MAX et sera retourné par la fonction, l'activité sera un entier choisi aléatoirement entre 1 et 3 et le nombre de personnes sera choisi aléatoirement entre 1 et 5.

Exemple de tableau crée

Nb = 8									
Nombre de personnes	5	4	4	5	4	5	3	2	
Activite pratiquée	3	3	2	1	2	1	1	3	

```
int RemplitTabAlea (int T[2][MAX])
{
    int i, NbPers, Activite;
    int nb = rand()%(MAX-10+1) +10;
    for (i=0;i<nb;i++)
    {
        Activite =rand ()%3 + 1;
        NbPers = rand()%5 +1 ;
        T[0][i]=NbPers;
        T[1][i]=Activite;
    }
    return nb;
}
```

- 3- Ecrire en C/C++ une **procédure** Statistiques qui à partir du tableau T rempli à la question précédente et du nombre Nb de groupes compte et "retourne" (sans afficher) le nombre total de marcheurs NbWalk, de coureurs NbRun et de cyclistes NbBike.

Exemple sur le tableau précédent : NbWalk = 13 / NbRun = 8 / NbBike = 11

```
void statistiques (int T[2][MAX], int n, int &nbwalkers, int &nbrunner, int &nbbiker)
{
    nbwalkers = 0;
    nbrunner = 0 ;
    nbbiker = 0;
    int i;
    for (i=0;i<n;i++)
        if (T[1][i]==1)
            nbwalkers+=T[0][i];
        else if (T[1][i]==2)
            nbrunner+=T[0][i];
        else
            nbbiker+=T[0][i];
}
```

- 4- Ecrire en C/C++ le programme principal qui remplit aléatoirement un tableau contenant le nombre de personnes et l'activité pratiquée, puis compte et affiche le nombre de marcheurs, de coureurs et de cyclistes. On utilisera les sous-programmes précédemment écrits.

```
int main (void)
{
    srand(time(NULL)); // pas demandé ici mais pour obtenir des tirages aléatoires différents
    int tab[2][MAX]={0};
    int marcheurs, coureurs, pedaleurs;
    int nb = RemplitTabAlea(tab);
    statistiques(tab,nb,marcheurs, coureurs, pedaleurs);
    cout<<marcheurs<<" marcheurs, "<<coureurs<<" coureurs et "<<pedaleurs<<" pedaleurs"<<endl;
    return 0 ;
}
```