**TP I1101 nº7**

**Les tableaux en C**

Exercice 1

1. Écrire un programme qui saisit un tableau d'entiers (contient obligatoirement des 0 et 1) et de dimension maximale 50 et qui arrête à saisir lorsque -1 est tapé et qui retourne la longueur de la plus longue séquence.

2.Ajouter les instructions nécessaires pour afficher les indices de début et de fin de la sous-séquence.

Exercice 2

Tri par comptage : Etant donné un tableau T de N entiers (sans répétition), il s’agit de trier T en construisant un autre tableau TC dit de comptage de la façon suivante :

Pour chaque élément T[i], TC[i] = nombre d'éléments < T[i] dans T.

Ainsi l’on peut déterminer la position (indice) de chaque élément dans le tableau résultat.

Indice 0

Exemple :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tableau initial T:** | **52** | **10** | **1** | **25** | **62** | **3** | **8** | **55** |
|  |  |  |  |  |  |  |  |  |
| **Tableau comptage TC:** | **5** | **3** | **0** | **4** | **7** | **1** | **2** | **6** |
|  |  |  |  |  |  |  | TRES[5]=T[0] |  |
| **Tableau résultat Res:** | **1** | **3** | **8** | **10** | **25** | **52** | **55** | **62** |

Écrire un programme C permettant de:

*Saisir le tableau T de dimension n;*

*Calculer le tableau de comptage TC;*

*Remplir et afficher le tableau ordonné Res.*

Exercice 3

1. Ecrire un programme qui remplit un tableau T d’une suite de 0 et de 1 d’au plus

MAX éléments, terminée par la valeur -1.

Par ex: 0 0 1 1 1 1 0 1 1 1 0 0 1 0 0 0 0 -1

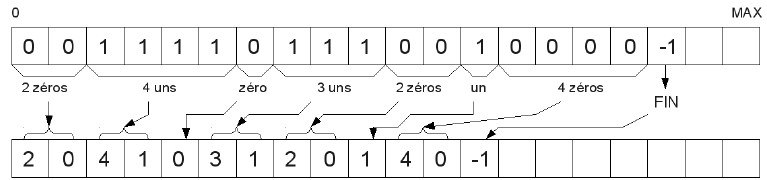
1. Ajouter les instructions nécessaires pour compresser T dans un autre tableau TCOMPRESS.

***Indication :***

L’algorithme de compression :

1. On recherche les groupes d’éléments identiques.
2. Si un groupe d’éléments identiques de T contient un seul élément, il est conservé tel quel dans TCOMPRESS.
3. Si un groupe d’éléments identiques de T contient n éléments avec n .2, il est stocké dans TCOMPRESS sous la forme “ne suivi pas de l’élément répété”.

Par exemple, la séquence ci-dessus sera compressée en : 2 0 4 1 0 3 1 2 0 1 4 0 -1 comme l’illustre le dessin ci-dessous.



**Solution**

Note : Explication selon la méthode de trace du Dr. May Dehayni est a la fin du document de quelque exercice.

Exercice 1

Methode 1 (simple):

#include <stdio.h>

#define N 50

void main()

{

int i=0,n, counter = 1,val, max = 0, init,T[N];

printf("donner les elements de tableau (max=50) de la forme:\n 0 et 1 seulement \n -1 deux fois pour s'arreter\n");

do

{

Remplir d’un tableau conditionné non prelimité (sortie par -1)

scanf("%d", &val);

if (val == 0 || val==1)

{

T[i] = val;

i++;

}

} while (val!=-1);

La limite du tableau remplis aprés est alors n

n = i;

for (i = 0; i < n ; i++)

Affichage d’un tableau avec ses indices correspondant

{

printf("T[%d]=%d \n", i, T[i]);

}

A ajouter pour able to compare avec T[n+1] dans le boucle et entrer évidement dans “else” voir si dessous.

T[n + 1] = -1;

for ( i = 0; i < n; i++)

{

if (T[i]==T[i+1])

{

counter++;

}

Boucle d’algorithme principale.

else

{

if (counter>max)

{

max = counter;

init = i - max +1;

}

counter = 1;

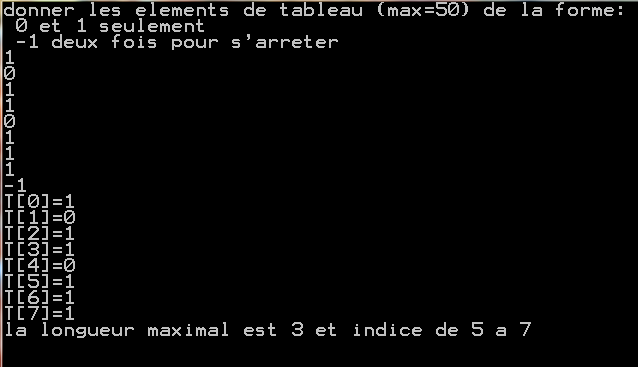
}

}

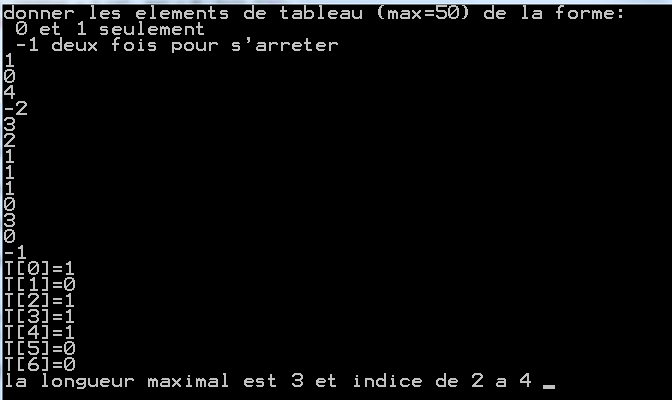
printf("la longueur maximal est %d et indice de %d a %d ",max,init,init+max-1);

}

**Exemple d’exécution :**



Another test case :



Méthode 2 (expert):

Une méthode experte est celle qui prend des cas plus généraux que celle du simple qui sont justes pour donner une réponse à la question.

Dans cet exercice, on a un cas spécial : plusieurs séquences peuvent se répéter avec la même longueur maximal ex : “101110001”. On a “111” et “000” de longueur 3, et évidement de différent indice ce qui fallait un plus avancée algorithme réponse sur la partie 2.

(Note : un second tableau est indispensable dans ce cas)

#include <stdio.h>

#define N 50

void main()

{

int i, counter = 1, max = 0, n=1, T[N], k=0, TMax[N];

printf("donner les elements de tableau (max=50) de la forme:\n 0 et 1 seulement \n -1 deux fois pour s'arreter\n");

for (i = 0; i < n; i++)

{

scanf("%d", &T[i]);

n++;

while (T[i]<0 || T[i]>1)

{

printf("\n donner un valeur valide! ");

scanf("%d", &T[i]);

if (T[i] == -1)

{

break;

}

}

if (T[i] == T[i - 1])

{

counter++;

}

else

{

if (counter > max)

{

max = counter;

k = 0;

TMax[k] = i - counter;

}

else

{

if (counter == max)

{

k++;

TMax[k] = i - counter;

}

}

counter = 1;

}

if (T[i] == -1)

{

break;

}

}

for (i = 0; i < n-2; i++)

{

printf("T[%d]=%d \n", i, T[i]);

}

printf(" la longuer maximal est %d\n", max,k);

for (i = 0; i < k+1; i++)

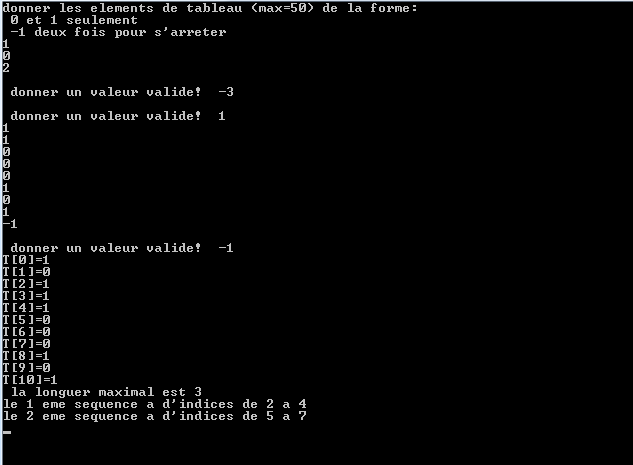
{

printf("le %d eme sequence a d'indices de %d a %d \n",i+1,TMax[i], TMax[i]+max-1);

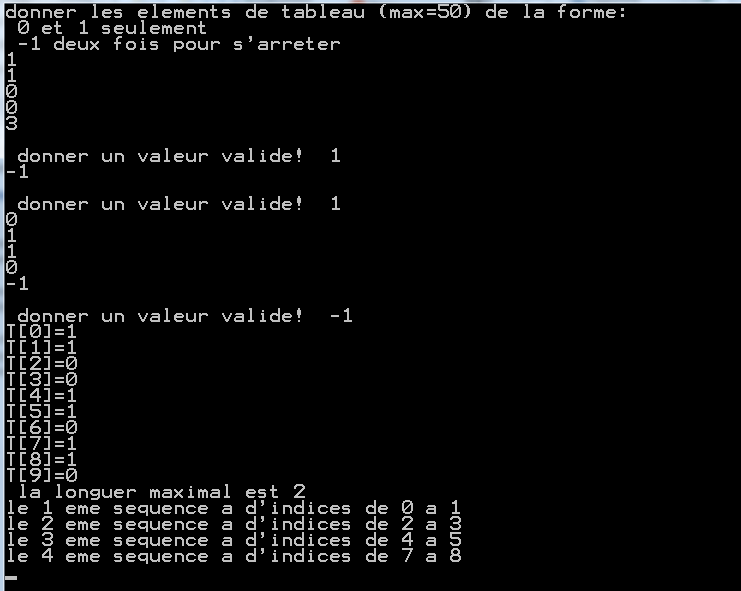
}

}

Exemple d’exécution Méthode 2 :



Another test case: (see below)



Exercice 2

#include <stdio.h>

#define N 50

void main()

{

int n, i, j, T[N], TC[N], TRes[N];

printf("entrer le nombre d'element du tableau");

scanf("%d", &n);

printf("remplir les element du tableau sans repetition");

for (i = 0; i < n; i++)

Remplir le tableau T normal

Et annuler tout élément dans TC afin d’ajouter directement après des “++” ou +1.

{

scanf("%d", &T[i]);

TC[i] = 0;

}

for (i = 0; i < n; i++)

{

for ( j = 0; j < n; j++)

{

if (T[j]<T[i])

{

Boucle d’algorithme principale.

TC[i]++;

}

}

TRes[TC[i]] = T[i];

}

for (i = 0; i < n; i++)

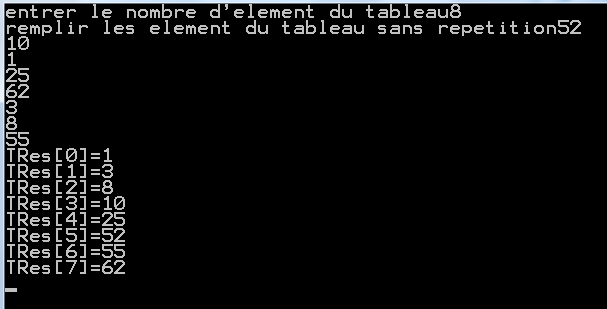
{

printf("TRes[%d]=%d \n", i, TRes[i]);

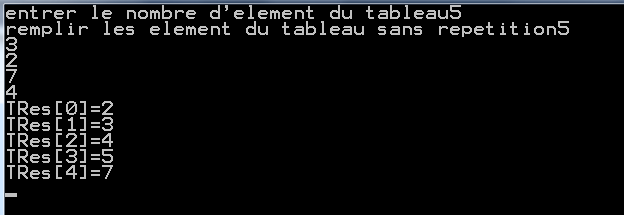
}

}

Exemple d’exécution :



Another test case:



Exercice 3

#include <stdio.h>

#define N 50

void main()

{

int i = 0, n, numgrp=1,k=0, T[N], TCOMPRESS[N],val;

printf("donner les elements de tableau (max=50) de la forme:\n 0 et 1 seulement \n -1 deux fois pour s'arreter\n");

do

{

scanf("%d", &val);

if (val == 0 || val == 1)

{

T[i] = val;

TCOMPRESS[i] = 1;

i++;

}

} while (val != -1);

T[i] = -1;

TCOMPRESS[i] = 1;

n = i + 1;

for (i = 0; i < n-1; i++)

{

if (T[i] == T[i + 1])

{

numgrp++;

}

else

{

if (numgrp==1)

{

TCOMPRESS[k] = T[i];

k++;

}

else

{

TCOMPRESS[k] = numgrp;

k++;

TCOMPRESS[k] = T[i];

k++;

numgrp = 1;

}

}

}

TCOMPRESS[k] = -1;

printf(" Forme compresse: ");

for (i = 0; i < k+1; i++)

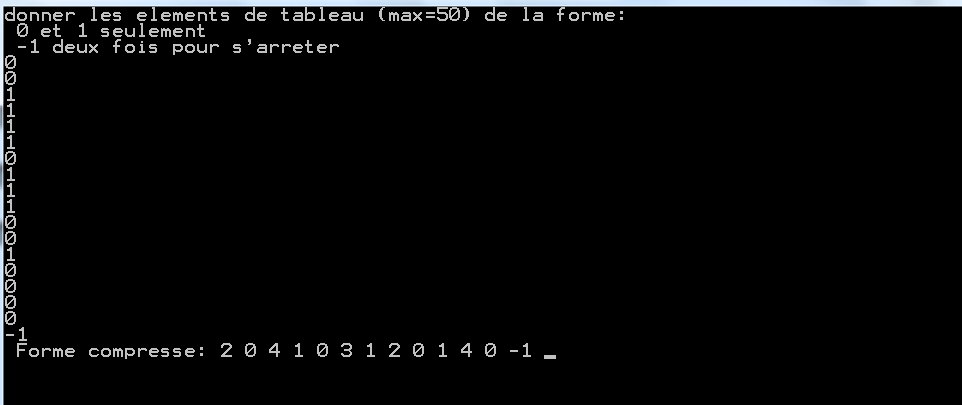
{

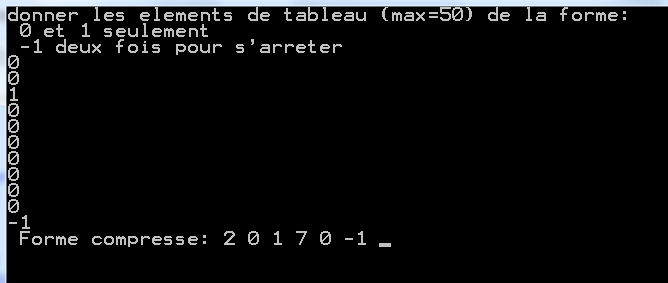
printf("%d ", TCOMPRESS[i]);

}

}

Exemple d’exécution :



Test case: 0010000000-1 >> 2017 0 -1

**Trace ??**

Le trace est de prendre un exemple numérique astucieux sur le programme et voir à chaque instruction les valeurs changes dans le mémoire, pour vérifier à la fin le résultat obtenu avec notre logique de la solution.

**Trace d’ Exercice 2 méthode 2: (image below)**

Cela peut être faite sur des feuilles main levée ! ou se fait juste en a joutant des simples instructions sur la boucle principale dans le programme.

// note écrire un self-program traces n’est pas demande jusqu’a moment

