**7.2 Fibonacci**

**Sub\_Algorithm** *FibR*

**Input**:

n: Integer

**Output**:

Value: Integer

**Variables**: (non 无)

**Instructions**:

If(n=1) Then

Value ← 1

ElseIf(n=2) Then

value ← 2

Else

value ← *FibR*(n1 !) + *FibR*(n)

EndIf

**End** *FibR*

**Algorithm** Main

Variables:

n: Integer

fibn: Integer

Instructions:

Write(“Enter a positive integer n: ” !)

Read(KBD ! n)

*FibR*(n ! fibn)

Write(“The ” ,n, “-th Fibonacci element is: ”, fibn !)

**End** Main

**8.1 Browse a list 1**

**SA Browse\_list1**

**Input parameters:**

Ptr: pointer to MDate\_naiss

**Output parameters:**

//Non

**Variables:**

i: Integer

**Instructions:**

While (Ptr ≠ NULL) Do

i ← 1

While(Ptr↑.nom[i] ≠ ‘\0’) Do

Write (Ptr↑.nom[i] !)

i ← i + 1

EndWhile

Write (Ptr↑.jour, Ptr↑.mois, Ptr↑.annee !)

Ptr ← Ptr↑.next

EndWhile

**End Browse\_list1**

**8.2 Browse a list 2**

**SA *ficherListeRec***

**Input parameters:**

Ptr: pointer to MDate\_naiss

**Output parameters:**

//Non

**Variables:**

i: Integer

**Instructions:**

found ← False

If (Ptr = NULL) Then

Write(“It is the end.” !)

Else

i ← 1

While(Ptr↑.nom[i] ≠ ‘\0’) Do

Write (Ptr↑.nom[i] !)

i ← i + 1

EndWhile

Write (Ptr↑.jour, Ptr↑.mois, Ptr↑.annee !)

Ptr ← Ptr↑.next

*ficherListeRec* (Ptr ! )

EndIf

**End *ficherListeRec***

**8.3 find a name in a list**

**SA *FindNameList***

**Input parameters:**

Ptr: pointer to MDate\_naiss

name: array[1..] of Characters

**Output parameters:**

found: Boolean

dd, mm, yy: Integer

**Variables:**

i: Integer

break: Boolean

**Instructions:**

found ← False

If Ptr ≠ NULL Then

i ← 1 // compare(Ptr↑.nom, name ! found)

break ← False

While(Ptr↑.nom[i] ≠ ‘\0’ AND name[i] ≠ ‘\0’ AND break = False) Do

If (Ptr↑.nom[i] ≠ name[i]) Then

Break ← True

Else

i ← i + 1

EndIf

EndWhile

If (Ptr↑.nom[i] ≠ name[i]) Then

Break ← True

EndIf

If (break = False) Then

found ← True

dd ← Ptr↑.date

mm ← Ptr↑.mois

yy ← Ptr↑.annee

Else

*FindNameList* (Ptr↑.next, name ! found, dd, mm, yy)

EndIf

If (found = False) Then

Write (“The name was not found.” !)

Else

Write (“The name was found, his/her birth-day is: ”, dd, mm, yy !)

EndIf

EndIf

**End *ficherListeRec***

**Instructions**:

found ← False

If (Ptr ≠ NULL) Then

Compare (Ptr↑.nom, name ! found)

If (found = True) Then

dd ← Ptr↑.date

mm ← Ptr↑.mois

yy ← Ptr↑.annee

Write (“The name was found, his/her birth-day is: ”, dd, mm, yy !)

Else

*FindNameList* (Ptr↑.next, name ! found, dd, mm, yy)

EndIf

EndIf

If (found = False) Then

Write (“The name was not found.” !)

EndIf

**8.4 Add**

Type:

*Person* : record (nom: array[1..40] of characters,

prenom: array[1..40] of characters,

next: pointer to a record Person,

MarriedTo: pointer to a record Person)

**SA add\_person**

**Input parameters:**

Ptr : pointer to *Person*

~~per\_nom, per\_prenom, marriedPerson\_nom, marriedPerson\_prenom: array[1..] of characters~~

per: *Person*

marriedPerson\_nom, marriedPerson\_prenom: array[1..] of characters

**Output parameters:**

New\_Ptr: pointer to *Person*

**Variables:**

~~per: record~~ *~~Person~~*  // 局部变量在子函数结束后会被摧毁，主函数中不会得到

**Instructions:**

*~~Copy~~*~~(per\_nom, per.nom !) // copy(string1, string2 ) 是一个子算法（子函数），用来把字符串 string1~~

*~~//~~* ~~复制一份赋值给string2.在这里默认这个子算法已经被写好了。~~

*~~Copy~~*~~(per\_prenom, per.prenom !)~~

~~per↑.next ← NULL //在链表的队尾添加新成员~~

While (Ptr ≠ NULL) Do //add this person to the end of the list

Ptr ← Ptr↑.next

EndWhile

Ptr ←@per

*FindName* (Ptr, marriedPerson\_nom, marriedPerson\_prenom ! found, Ptr\_person)

// *FindName* (Ptr, name, surname ! found) 是一个子算法（子函数）,用来在Ptr 链表里寻找名name，姓surname的人是否存在，返回boolean值 found,以及指向该人地址的指针Ptr\_person。

If (found = True) Then

Ptr\_person↑.marriedTo ← @per

per↑.marriedTo ← Ptr\_person

Else

per↑.marriedTo ← NULL

EndIf

**End Add\_person**