

Final Exam

Exercise 1 (6 points):

1. Write a **recursive** function *TotallyEven* (N: integer) that checks if an integer N is a totally even number or not.

Function *TotallyEven* (N: integer) : boolean (2.5 points)

If (N = 0) then

0,75

Return True;

Else

0,75

If ((N mod 10) mod 2 ≠ 0) then

Return False; 0,5

Else

Return *TotallyEven* (N div 10); 1

Finsi

Fin

2. Write a procedure **display** (A: integer) that displays all totally even numbers exist in a file of integers "Numbers.bin" and are less than A.

Procedure *display* (A: integer) (2.5 points)

F: file of integers; 0,25

X: integer;

Begin

ASSIGN (F, "Number.bin") 0,25

Open (F); 0,25

While (! eof (F)) do 0,5

Read (F, X); 0,5

If (*TotallyEven* (X) and X < A) then 0,5 0,25

Write (x); 0,25

End if

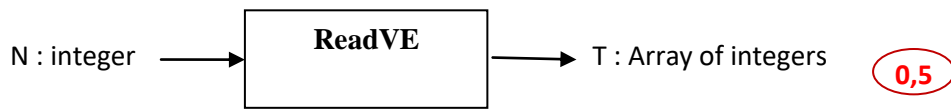
End while

Close (F); 0,25

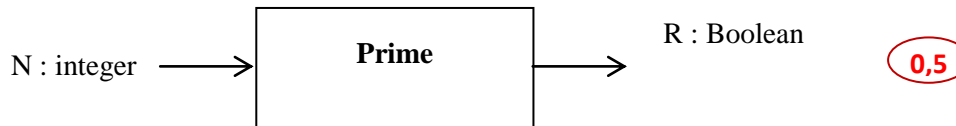
End

Exercise 2 (8 points)

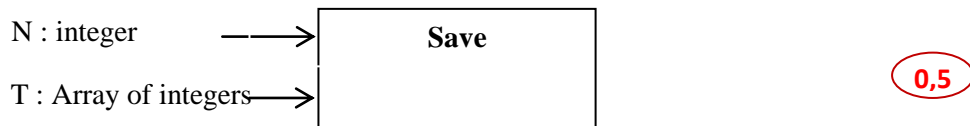
1. Make the modular division of this problem. (1.5 points)



Role : Read an array of N integer



Role : Check if an integer number N is prime or not



Role: Save all prime numbers of an array into a file of integers

2. Write the principal algorithm with all necessary modules.

Algorithm Exo4

(0,25)

Procedure *ReadVE* (Var V []:array of integers, n: integer) (1.5 points)

```
i: integer;  
Begin  
  For i ← 1 to n do  
    Read (V[i]);  
  End For  
END
```

(1.25)

Function *prime* (N: integer): Boolean (2.5 points)

```
i: integer;  
R: boolean;  
Begin  
If (N<2) then  
  Return false  
Else  
  R ← true;
```

(0,5)

```

i ← 2 ;
While (i ≤ Ndiv2 and R=true) do
    If (N mod i = 0) then
        R ← false;
    End if
    i ← i+1 ;
End while
Return (R);

```

END

Procedure Save (V []:array of integers, n: integer) (2 points)

```

F: file of integers
i: integer;

```

Begin

```

ASSIGN (F, "prime.bin");

```

```

Rewrite (F);

```

```

For i ← 1 to n do

```

```

    If (prime(V[i]) then

```

```

        Write (F, V[i]);
    End if

```

```


```

```

End For

```

```

Close (F);

```

End

```

T [100]: array of integers;

```

```

Size: Integer ;

```

Begin (1.5 points)

```

Read (size);

```

```

ReadVE (T, size);

```

```

Save (T, size);

```

END.

Exercise 3 (6 points): write the two following modules on linked list of integers.

1. The function *check* (L) that checks if a list L is a list of digits (contains only digits numbers (1, 2... 9)).

```
// version itérative. (2.5 points)

Function Check (L: List): Boolean // iterative version.

Current : List;

Begin

Current ←L; 0,25

While (Current != NULL) do 0,25

    If (Current -> Ele > = 10) then 0,75

        Return false; 0,5

    End if

    Current ←Current ->next ; 0,25

EndWhile

return (True); 0,5

END
```

2. The function *decimal* (L) that calculates and returns the number represented by a list of digits L (returns the numbers composed by the same digits and with the same order).

```
Function decimal (L: List): integer // iterative version. (2.5 points)

Current : List; S : integer

Begin

Current ←L; S ←0 ; 0,5

While (Current != NULL) do 0,25

    S ←S *10 + Current-> Ele ; 1,25

    Current ←Current ->next ; 0,25

EndWhile

return (S); 0,25

END
```