University of 08 Mai 1945 – Guelma
Faculty of MISM
Department of Mathematics
First year LMD

Date: 12/05/2024
Duration : 2h00
Semester 2

Final Exam – Solution Key

Exercise 1: (5 pts)

1) **The iterative function `sumIt`:** **(2.5 pts)**

0.5	`Function sumIt(A,B:integer):integer;`
0.25	`Var i,s:integer;`
	`Begin`
0.25	`S ← 0;`
0.5	`For i ← A to B do`
0.5	`s ← s + i;`
0.5	`sumIt ← s;`
	`End;`

2) **The recursive function `sumRec`:** **(2.5 pts)**

0.5	`Function sumRec(A,B:integer):integer;`
	`Begin`
1	`If A=B then sumRec ← B`
1	`Else sumRec ← A + sumRec(A+1,B);`
	`End;`

Exercise 2: **(7.5 pts)**

The data structure: **(0.25 pts)**

```
Type  List = ^node;
      node =  Record
              Begin
              val: integer;
              next: List;
              End;
```

1) The procedure `nbSumHeights`: **(2.75 pts)**

0.5	`Procedure nbSumHeights(L:List;Var nb,s:integer);`
0.25	`Var p:List;`
	`Begin`
0.25	`nb ← 0;`
0.25	`s ← 0;`
0.25	`p ← L;`
0.5	`While p ≠ Nil Do`
	`Begin`
0.25	`nb ← nb + 1;`
0.25	`s ← s + p^.val;`
0.25	`p ← p^.next;`
	`End;`
	`End;`

2) The function `isTallDominant`: **(4.5 pts)**

0.25	`Function isTallDominant(L:List):Integer;`
0.5	`Var p:List;nb,s,nbAbove:integer;avg:real;`
	`Begin`
0.5	`nbSumHeights(L,nb,s);`
0.25	`nbAbove ← 0;`
0.25	`If nb≠0 Then`
	`Begin`
0.25	`avg ← s/nb;`
0.25	`p ← L;`
0.5	`While p ≠ Nil Do`
	`Begin`
0. 5	`If p^.val > avg then`
	`nbAbove ← nbAbove + 1;`
0.25	`p ← p^.next;`
	`End;`
	`End;`
0. 5	`If nbAbove > nb/2 Then isTallDominant ← True`
0. 5	`Else isTallDominant ← False;`
	`End;`

Exercise 3: **(7.5 pts)**

1)

The data structure: **(0.25 pts)**

```
Type  Stack = ^node;
      node =  Record
              Begin
              val: integer;
              next: List;
              End;
```

The procedure `binarize`: **(3.5 pts)**

0.25	`Procedure binarize(Var S :Stack;v:integer);`
0.25	`Var e:integer;P:Stack;tr:boolean;`
	`Begin`
0.25	`initializeStack(P);`
0.25	`tr ← False;`
0.25	`While isStackEmpty(S)=False do`
	`Begin`
0.25	`Pop(e,S);`
0.25	`if e=v then tr ← true;`
0.25	`Push(e,P);`
	`End;`
0.25	`While isStackEmpty(P)=False do`
	`Begin`
0.25	`Pop(e,P);`
0.25	`If tr=True then`
0.25	`If e<v Then e ←0`
0.25	`Else e ← 1;`
0.25	`Push(e,S);`
	`End;`
	`End;`

2)

The data structure: **(0.25 pts)**

```
Type  Queue = ^node;
      node =  Record
              Begin
              val: integer;
              next: List;
              End;
```

The procedure `insert`: **(3.5 pts)**

0.25	`Procedure insert(Var Q:Queue,v:integer);`
0.25	`Var e:integer;Q2:Queue;`
	`Begin`
0.25	`initializeQueue(Q2);`
0.5	`While isQueueEmpty(Q)=False and peekQueue(Q)<v Do`
	`Begin`
0.25	`Dequeue(e,Q);`
0.25	`Enqueue(e,Q2);`
	`End;`
0.25	`Enqueue(v,Q2);`
0.25	`While isQueueEmpty(Q)=False Do`
	`Begin`
0.25	`Dequeue(e,Q);`
0.25	`Enqueue(e,Q2);`
	`End;`
0.25	`While isQueueEmpty(Q2)=False Do`
	`Begin`
0.25	`Dequeue(e,Q2);`
0.25	`Enqueue(e,Q);`
	`End;`
	`End;`