Mila University Centre
Institute of mathematics and computer science

2nd year computer science
Year: 2023-2024

First exam
Algorithmic and data structures 3

Exercise 1 (6 points):

1. Write a function `Evaluate (F, x)` that returns the result of evaluating the polynomial represented by the queue *F* for a given value of x. for example if x = 2 the evaluation of the polynomial P(x) represented by F in the previous example is: $P(2) = 6 * \mathbf{2}^4 + 3 * \mathbf{2}^2 + 9 = 117$

```
Function Evaluate (F: Queue, x: real) : real   (2.5)
P: real;
Begin
        P←0;                                   0,25
        While ! is_empty (F) do                0,5
                P ← P + Head (F).Coef * x^ Head (F).Expo;   1
                Dequeue (F);                   0,5
        End While
        Return P;                              0,25
End
```

2. Write a function `Same_poly(F1, F2)` that checks whether two queues *F1* and *F2* represent the same polynomial.

```
Function Same_poly (F1, F2: Queue) : Boolean     (3.5)
Begin
        While ! is_empty (F1) and !  is_empty (F1) do   0,5
                If  Head (F1).Coef ≠  Head (F2).Coef or Head (F1).Expo ≠  Head (F2).Expo then   01
                        Return flase;          0,5
                Endif
                Dequeue (F1);    Dequeue (F2);   0,5
        End While                              0,5
        If (is_empty (F1) and is_empty (F2) then
                Return true;                   0,25
        Else
                Return flase;                  0,25
        Endif
End
```

Exercise 2 (6 points): write the following two procedures for linked lists:

1. The procedure `Split(L, L1, L2).` **(2.25)**

```
Procedure Split(var L, L1, L2: List)
   C: List
Begin
      N ← length (L) div 2;                    0,25
0,25  C ← L;        L1 ←L;                     0,25
      L2 ← Null;                               0,25
0,25  If (N>1 then)
            While N>1 do                       0,25
                  C←C→Nexte;                   0,25
            0,25  N ←N-1;
            End While;
            L2 ← C→next;                       0,25
            C →next ← NULL;                    0,25
      Endif
END
```

2. The procedure `Split2 (L, L1, L2).` **(4 points)**

```
Procedure Split2(var L, L1, L2: List)
   C, C1, C2: List
Begin
      L1←NULL; L2←NULL;                        0,25
      C ← L;
      While C≠NULL do                          0,25
            If C→Ele mod 2=0 then              0,25
                  If L1 =NULL then             0,25
                        L1←C;                  0,5
                        C1←L1;
                  Else
                        C1 ->Next ← C;
                        C1 ← C;                0,5
                  End If
            Else
                  If L2 =NULL then
                        L2←C;                  0,5
                        C2←L2;
                  Else
                        C2 ->Next ← C;         0,5
                        C2 ← C;
                  End If
            Endif
            C←C→Nexte;
      End While;
      If (C1 ≠ NULL) then
            C1 ->Next ← NULL;                  0,5
      End If
      If (C2 ≠ NULL) then
            C2 ->Next ← NULL;                  0,5
      End If
END
```

Exercise 3 (8 points):

1. Provide the necessary declarations for this representation. **(2)**

```
Type Structure Product
        Ref: integer;
        Name: String;                    0,25
        Quantity, UPrice: real;
 End
Type Structure NodeL
        Ele: Product;                    0,25
        Next: * Node;
End
Type List: *NodeL;                       0,25
Type Structure Date
        D, M, Y: Integer;                0,25
End
Type Structure Order
        ID: Integer;
        FirstName, LastName: String;     0,5
        D: Date;
        L: List;
End
Type Structure Node
        Ele: Order;                      0,25
        LC, RC: Node;
End
Type Tree: *Node;                        0,25
```

2. Write the procedure OrdersMonth (a, M, Y).

```
Procedure OrderMonth (a: Tree, M,Y: integer)                    (2.25)
Begin
If a ≠ NULL then 0,25                                           0,5
        If Content (a).Month =M and Content (a).Year = Y) then
                Write (Content (a).ID);  0,5
        Endif
        OrderMonth (LeftChild (a), M,Y);        0,5
        OrderMonth (RightChild(a), M, Y)        0,5
End
```

3. Write a procedure `SearchOrder (a, ID).` **(4)**

Procedure `SearchOrder (a: Tree,ID: Integr)` **2.25**

```
If a = NULL then
        Write ("This order don't exist"); 0,5
Else
        If Content (a).ID =Id the 0,5
                Display_order (Content (a)); 0,5
        Endif
                If ID < Content (a).ID then 0,25
                        OrderMonth (LeftChild (a), ID); 0,25
                Else
                        SearchOrder (RightChild(a), ID) 0,25
                End if
        EndIf
Endif
End
```

Procedure display_order (O: Order) **1.75**

```
Begin
        Write (O.FirstName,  O.LastName) 0,25
        C← O.L; 0,25
        S←0; 0,25
        While (C≠Null) do 0,25
                Write ( C->Ele.name, C->Ele.Quantity); 0,25
                S←S+ C->Ele.Quantity* C->Ele.UPrice); 0,25
        End While
                Write (S); 0,25
End
```