

## Final Exam (Correction)

Exercise 1(8pts):

```
Algorithm ex1; // algorithm structure(algorithm var begin end ) 0.5 pts
Var prev, curr, consumption, tier1, tier2, tier3, tier4: integer;
    Total, vat: real; /* 0.5 pts (Every variable used in the algorithm must be declared.) */
Begin
    do
        write ("Enter previous meter reading: "); // 0.25 pts
        read(prev); // 0.25 pts
        while (prev < 0); // 0.5 pts

    do
        write("Enter current meter reading: "); // 0.25 pts
        read(curr); // 0.25 pts
        while (curr < prev); // 0.5 pts

    consumption ← curr - prev; // 0.5 pts

    tier1 ← 0;
    tier2 ← 0;
    tier3 ← 0;
    tier4 ← 0;

    if (consumption <= 125) then
        tier1 ← consumption;
    else if (consumption <= 250) then
        tier1 ← 125;
        tier2 ← consumption - 125;
    else if (consumption <= 1000) then
        tier1 ← 125;
        tier2 ← 125;
        tier3 ← consumption - 250;
    else
        tier1 ← 125;
        tier2 ← 125;
        tier3 ← 750;
        tier4 ← consumption - 1000;
    end if
    end if
    end if

    total ← (tier1 * 1.7787) + (tier2 * 4.1789) + (tier3 * 4.8120) + (tier4 * 5.4796);
    vat ← (tier1 * 1.7787 * 0.09) + (tier2 * 4.1789 * 0.09) + (tier3 * 4.8120 * 0.19) + (tier4 * 5.4796 * 0.19);
    total ← total + vat;

    write("Total electricity bill = ", total); // 0.5 pts

end.
```

/\* Correctly calculate the amount for one tier  
(1 point each × 4 tiers = 4 pts) \*/

## Exercise 2 (6pts):

Write an algorithm (or a C program) and draw a flowchart that:

1. Asks the user to enter two integers.
2. Calculates the sum of all integers between the smaller and larger numbers that are divisible by 3 or 5.
3. Displays the sum of these numbers.

### Example:

If num1 = 1 and num2 = 15, the integers satisfying the condition are 3, 5, 6, 9, 10, 12, and 15, and their sum is 60.

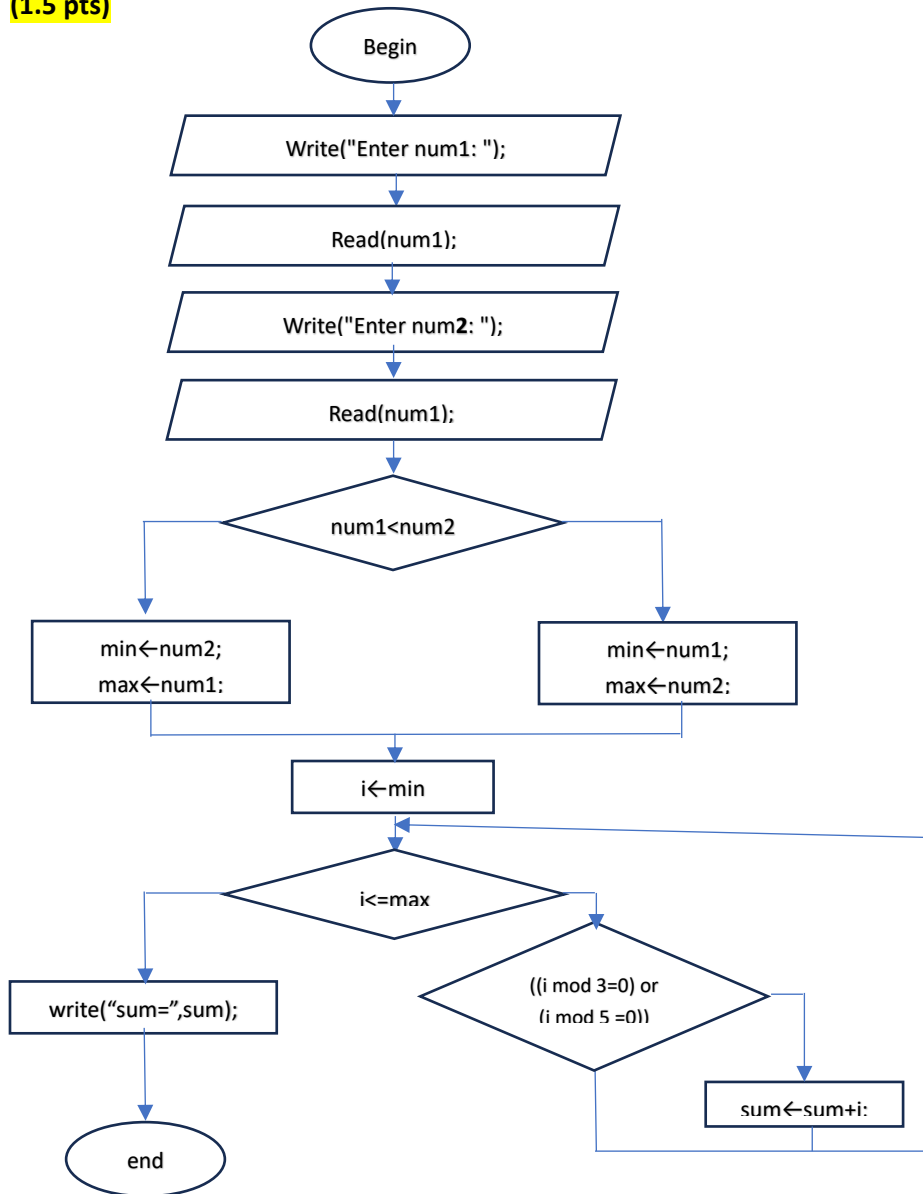
```
Algorithm ex2; // algorithm structure(algorithm var begin end ) 0.5 pts
Var num1, num2, min, max, i: integer; //0.5 pts (Every variable used in the algorithm must be
declared.)
Begin
  Write("Enter num1: "); //0.25 pts
  Read(num1); //0.25 pts
  Write("Enter num2: "); //0.25 pts
  Read(num2); //0.25 pts

  If (num1<num2)then
    min←num1;
    max←num2;
  else
    min←num2;
    max←num1;
  end if //0.5 pts

  for i←min to max do //0.5 pts
    if((i mod 3=0) or (i mod 5 =0)) then //0.5 pts
      sum←sum+i; //0.5 pts
    end if
  end for
  write("sum=",sum); // 0.5 pts
end.
```

Correctly calculate the sum (2 pts)

(1.5 pts)



### Exercise 3(6pts):

Write an algorithm (or a C program) that:

1. Asks the user to enter the elements of a 5x5 matrix A.
2. Replaces all zeros in the matrix with the largest number on the main diagonal.
3. Displays the resulting matrix.

$$A = \begin{bmatrix} 0 & 5 & 0 & 1 & 0 \\ 3 & 0 & 7 & 0 & 8 \\ 0 & 6 & 4 & 0 & 3 \\ 1 & 0 & 0 & 7 & 0 \\ 6 & 3 & 2 & 1 & 0 \end{bmatrix}$$

main diagonal

```
Algorithm ex3; // algorithm structure(algorithm var begin end ) 0.5 pts
Var A:array [5,5] of integer;
    maxDiag,i,j:integer; //0.5 pts (Every variable used in the algorithm must be declared.)
begin
    write("Enter the elements of the 5x5 matrix:");
    for i ← 0 to 4 do
        for j ← 0 to 4 do
            read(A[i,j]);
        end for
    end for

    maxDiag ← A[0][0];
    for i ← 1 to 4 do
        if (A[i][i] > maxDiag) then
            maxDiag ← A[i][i];
        end if
    end for

    for i ← 0 to 4 do
        for j ← 0 to 4 do
            if (A[i][j] = 0) then
                A[i][j] ← maxDiag;
            End if
        End for
    End for

    write("Resulting matrix:");
    for i ← 0 to 4 do
        for j ← 0 to 4 do
            write( A[i][j]);
        End for
    End for
End.
```