> Question 1

> d)

```
> rsolve({T(n)= 3*T(n/2) + c*n, T(1) = d}, T(n));
```

$$d\,n^{\frac{\ln(3)}{\ln(2)}} - 2\,c\,n + 2\,n^{\frac{\ln(3)}{\ln(2)}}\,c \tag{1}$$

```
> rsolve({M(n)= 2*M(n/2) + n/2, M(1) = 0}, M(n));
```

$$\frac{n\ln(n)}{2\ln(2)} \tag{2}$$

```
> rsolve({T(n)= T(n-1) + (n-1)^2, T(1) = 0}, T(n));
```

$$4\,n + 3 + 2\,(n+1)\left(\frac{n}{2}+1\right)\left(\frac{n}{3}+1\right) - 5\,(n+1)\left(\frac{n}{2}+1\right) \tag{3}$$

> e)

```
> my_fft := proc(n, A, p, w)
      local i; local b; local c;
      local B; local C; local T; local W; local FFT_A;
      if n = 1 then
          return A;
      else
          b := [];
          c := [];
          W := Array(1...n/2);
          FFT_A := Array(1...n);
          # split A into even and odd
          for i from 1 to ArrayNumElems(A) do
              if (i - 1) mod 2 = 0 then
                  b := [op(b), A[i]];
              else
                  c := [op(c), A[i]];
              fi;
          od;
          B := my_fft(n/2, Array(b), p, w^2);
          C := my_fft(n/2, Array(c), p, w^2);

          # precompute W
          for i from 1 to n/2 do
              W[i] := w^(i-1);
          od;

          for i from 1 to n/2 do
              T := ((W[i] mod p)* C[i]) mod p;
              FFT_A[i] := (B[i] + T) mod p;
              FFT_A[i + n/2] := (B[i] - T) mod p;
          od;
          return FFT_A;
      fi;
  end;
```

$$my\_fft := \mathbf{proc}(n, A, p, w) \tag{4}$$

```
    local i, b, c, B, C, T, W, FFT_A;
    if n = 1 then
        return A
    else
```

```
        b := [ ];
        c := [ ];
        W := Array(1..1/2*n);
        FFT_A := Array(1..n);
        for i to ArrayNumElems(A) do
            if i - 1 mod 2 = 0 then b := [op(b), A[i]] else c := [op(c), A[i]] end if
        end do;
        B := my_fft(1/2*n, Array(b), p, w^2);
        C := my_fft(1/2*n, Array(c), p, w^2);
        for i to 1/2*n do W[i] := w^(i - 1) end do;
        for i to 1/2*n do
            T := (W[i] mod p) * C[i] mod p;
            FFT_A[i] := B[i] + T mod p;
            FFT_A[i + 1/2*n] := B[i] - T mod p
        end do;
        return FFT_A
    end if
end proc
```

```
> A := Array([1,2,3,4,3,2,1,0]); p := 97; w := 5; n := 8;
```

$$A := \begin{bmatrix} 1 & 2 & 3 & 4 & 3 & 2 & 1 & 0 \end{bmatrix}$$
$$p := 97$$
$$w := 5$$
$$n := 8 \qquad (5)$$

```
> B := my_fft(n, A, p, w);
```

$$B := \begin{bmatrix} 16 & 63 & 0 & 58 & 0 & 33 & 0 & 32 \end{bmatrix} \qquad (6)$$

```
> C := my_fft(n, B, p, w^(-1));
```

$$C := \begin{bmatrix} 8 & 16 & 24 & 32 & 24 & 16 & 8 & 0 \end{bmatrix} \qquad (7)$$

```
> C - (n*A mod p);
```

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (8)$$

> f)

```
> a := Array([1, 3,  0, -1, 0, 0, 0, 0]);
  b := Array([1, 1, -2, -3, 2, 0, 0, 0]);
```

$$a := \begin{bmatrix} 1 & 3 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}$$
$$b := \begin{bmatrix} 1 & 1 & -2 & -3 & 2 & 0 & 0 & 0 \end{bmatrix} \qquad (9)$$

```
> fft_a := my_fft(8, a, 97, 5);
  fft_b := my_fft(8, b, 97, 5);
```

$$fft\_a := \begin{bmatrix} 3 & 85 & 4 & 9 & 96 & 14 & 95 & 90 \end{bmatrix}$$

(10)

$$fft\_b := \begin{bmatrix} 96 & 64 & 8 & 43 & 3 & 28 & 2 & 55 \end{bmatrix} \quad (10)$$

```
> fft_c := (fft_a * fft_b) mod 97;
```

$$fft\_c := \begin{bmatrix} 94 & 8 & 32 & 96 & 94 & 4 & 93 & 3 \end{bmatrix} \quad (11)$$

```
> c := my_fft(8, fft_c, 97, 1/5 mod 97);
```

$$c := \begin{bmatrix} 36 & 93 & 44 & 17 & 8 & 3 & 82 & 81 \end{bmatrix} \quad (12)$$