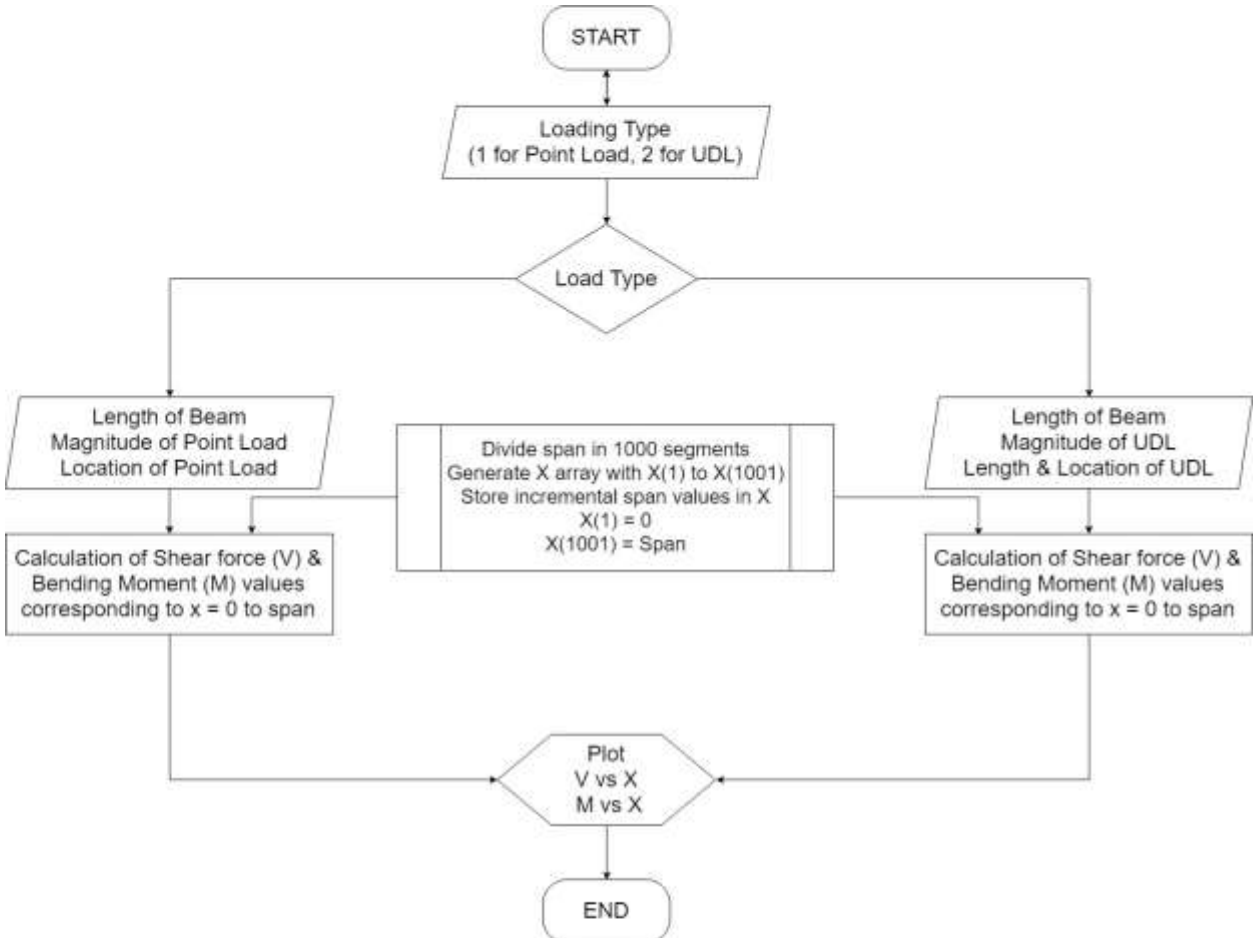
## Program Structure:

Program structure helps us in visualizing the over all lay out of code. This is very helpful in constructing the algorithm. The flow chart/structure of our last week's program is as below;



## Short from of Algorithm:

Create Program Environment

Input for Load Type

**%Code for Input of Data**

**IF** Load Type = Point Load

> Beam Length
>
> Point Load Magnitude  (A)
>
> Point Load Location

**IF** Load Type = UDL

> Beam Length
>
> UDL Magnitude  (B)
>
> UDL Length
>
> UDL Location

**END**

**%Code for Calcualtions**

**IF** Load Type = Point Load  (C)

> Code for Point Load Shear Force (V) & Bending Moment (M)

**IF** Load Type = UDL  (D)

> Code for UDL Shear Force (V) & Bending Moment (M)

**END**

**Plot**

## A. CODE FOR INPUT (POINT LOAD)

```
L    = input('Length of beam in meter =  ');
W = input('Load applied in kN = ');
a = input('Location of Load from left end of the beam in meter = ');
c  = L-a;

R1 = W*(L-a)/L;            % Left Support Reaction.
R2 = W*a/L;               % Right Support Reaction.
```

## B. CODE FOR INPUT (UDL)

```
L    = input('Length of beam in meter =  ');
W  = input('Uniformly distributed load in kN/m = ');
 b  = input('Length of udl in meter = ');
 cg = input('C.G of udl from left end of the beam in meter = ');
 a  = (cg-b/2);
 c  = L-a-b;

  R1 = W*b*(b+2*c)/(2*L); % Left Support Reaction.
  R2 = W*b*(b+2*a)/(2*L); % Right Support Reaction.
```

## C. CODE FOR CALCULATIONS (POINT LOAD)

```
% Discretization of x axis.
x = (0:L/1000:L)';        % Generate column array for x-axis.
V = zeros(size(x, 1), 1);   % Shear force function of x.
M = zeros(size(x, 1), 1);   % Bending moment function of x.
%SFD & BMD Calculation
for i = 1:n+1
   % First portion of the beam, 0 < x < b
   V(i) = R1;
   M(i) = R1*x(i);

   % Second portion of the beam, b < x < L
   if x(i) >= a
      V(i) = R1-W;
      M(i) = R1*x(i)-W*(x(i)-a);
   end
end
```

## D. CODE FOR CALCULATIONS (UDL)

```
% Discretization of x axis.
x = (0:L/1000:L)';        % Generate column array for x-axis.
V = zeros(size(x, 1), 1);   % Shear force function of x.
M = zeros(size(x, 1), 1);   % Bending moment function of x.
%SFD & BMD Calculation
for i = 1:n+1
   % First portion of the beam, 0 < x < a
   if x(i) < a
   V(i) = R1;
   M(i) = R1*x(i);
   elseif a <= x(i) && x(i)< a+b
   % Second portion of the beam, a < x < a+b
      V(i) = R1-W*(x(i)-a);
      M(i) = R1*x(i)-W*((x(i)-a)^2)/2;
   elseif x(i) >= (a+b)
   % Second portion of the beam, a+b < x < L
      V(i) = -R2;
      M(i) = R2*(L-x(i));
   end
  end
```

## E. CODE FOR PLOTING

```
figure
subplot(2,1,1);
plot(x, V, 'r','linewidth',1.5);
grid
line([x(1) x(end)],[0 0],'Color','k');
line([0 0],[0 V(1)],'Color','r','linewidth',1.5);
line([x(end) x(end)],[0 V(end)],'Color','r','linewidth',1.5);
title('Shear Force Diagram','fontsize',16)
text(a/2,V(1),num2str(V(1)),'HorizontalAlignment','center','FontWeight','bold','fontsize',16)
text((L-c/2),V(end),num2str(V(end)),'HorizontalAlignment','center','FontWeight','bold','fontsize',16)
axis off

subplot(2,1,2);
plot(x, M, 'r','linewidth',1.5);
grid
line([x(1) x(end)],[0 0],'Color','k');
line([x1 x1],[0 Mmax],'LineStyle','--','Color','b');
title('Bending Moment Diagram','fontsize',16)
```

```
text(x1+1/L,Mmax/2,num2str(roundn(Mmax,-
2)),'HorizontalAlignment','center','FontWeight','bold','fontsize',16)
text(x1,0,[num2str(roundn(x1,-2)) ' m'],'HorizontalAlignment','center','FontWeight','bold','fontsize',16)
axis off
```