M.V.S.R. Engineering College

```c
# include <stdio.h>
# include <conio.h>
# include <math.h>
main()
{
int  a,b,L,W,x,SF,BM,RA,RB;
clrscr();
Printf("enter the values of a,b,L,W");
Scanf("%d %d %d %d",&a,&b,&L,&W);
Printf("enter the value of x from left");
Scanf("%d",&x);
RB = W(b-a)~~(a+b)~~/2L;
RA = W(b-a)-RB;
If (x<a)
{
SF = RA;
BM = RA*x;
Printf(" the values of SF=%d, BM=%d",SF,BM);
}
else
{
if (x<b)
{
SF= RA-W*(x-a);
BM= RA*x-(W*(x-a)^2)/2;
Printf(" the values of SF=%d, BM=%d",SF,BM);
}
else
{
SF=-RB;
BM= -RB*(L-x);
Printf("the values of SF=%d, BM=%d",SF,BM);
}
```

```
}
getch();
}
```

Output:

```
enter the values of a,b,L,w
   5 15 20 10
enter the value of x from left
      3
  RA=50 , RB=50
  SF=50 , BM=150
```

8/8/19

```
# include <stdio.h>
# include <conio.h>
# include <math.h>
main()
{
float P, ex, ey, X, Y, Q, B, D;
clrscr();
Printf ("enter the values of P, B, D, ex, ey, X, Y");
Scanf ("%f %f %f %f %f %f %f", &P, &B, &D, &ex, &ey, &X, &Y);

Q= (P/(B*D)) + ((P*ex*Y*12)/(B*D*D*D)) + ((P*ey*X*12/D*B*B*B));
Printf ("Stress at P(X,Y) is Q= %f", Q);
getch();
}
```

Output:

```
enter the values of P, B, D, ex, ey, X, Y
50 30 100 10 50 5 20

Stress at P(X,Y) is Q= 0.076222
```

8/8/19

Q: Structural design of an RCC beam section using limit state method, given are the grade of concrete, grade of steel, BM & SF.

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
void main()
{
int   stirrupdia, maindia, fy, fck, cc;
float Ast, b, d, Beta, D, Deff, nb, Mu, SF, k, Q, Tc, Tv, Tcmax, Pt, Sv, Sv2,
      P, B, Vc, Vu, As, Asv, Vus;
Printf("enter the values of In factored Mu, SF in KN-m & KN");
Scanf(" %f %f, &Mu, &SF);
Mu = Mu * 1e6;
SF = SF * 1e3;
Printf("enter the values in N & mm for In fck, fy, cc and dia
        of main reinf");
Scanf("%d %d %d %d", &fck, & fy, &cc, & maindia);
getk:
if (fy==250) k= 0.53;
else if (fy==415) k= 0.48;
else if (fy==500) k= 0.45;
else { printf ("revise fy");
scanf("%d", &fy);
goto getk;
};
Printf("enter breadth of beam");
Scanf(" %f ", &b);
Q = 0.36 * fck * k * (1-0.42 * k);
```

```
Printf ("\n Q= %f \n", Q);
d= sqrt (Mu /(Q*b));
D = (d+cc+ (maindia)/2);
Printf ("min required overall depth of beam is %f \n", D);
Printf(" enter appropriate value of D");
scanf ("%f", &D);
Deff = D-cc- maindia /2 ;
Ast = 0.5 * fck * b * Deff * (1-sqrt(1-4.6* Mu/(fck * b* Deff * Deff)))/fy;
nb = Ast * 4 /(3.14 * main dia * maindia);
Printf ("\n Ast regd = %f mm^2 \n provide %f number of
  %d mm. dia bars. \n", Ast, nb, maindia");
if (fck==20) Tcmax =2.8;
else if (fck==25) Tcmax=3.1;
else if (fck==30) Tcmax=3.5;
else if (fck==35) Tcmax=3.7;
else Tcmax= 4.0;
~~sertav~~ ;
TV= SF/ (b* Deff);
if (Tv > Tcmax)
{ Printf(" D provided (%f) is inadequate for shear.
      \n Enter increased D", D);
scanf ("%f", &D);
Deff= D- CC- maindia;
~~goto sertav;~~
}
Sv=300;
if (Sv> (0.75 * Deff)) Sv= 0.75 * Deff;
Printf ("\n enter the dia of reinforcement for stirrups");
```

```
Scanf ("%d", & stirrupdia);
Asv = 2* 3.14 * stirrupdia * stirrupdia /4;
sv2 = ((Asv * fy) / (0.4 * b));
if (sv> sv2) sv= sv2 ;
P= 100 * Ast / (b* Deff);
Beta= 0.8* fck / (6.89* P) ;
if (Beta<1) Beta=1;
Tc = (0.85* sqrt (0.8 * fck) * (sqrt (1+5* Beta)-1) / (6* Beta));
if (Tc<Tv)
{
Vc = Tc * b* Deff;
Vus = SF-Vc;
sv2 = 0.87* fy * Asv * Deff /(Vus);
};
if (sv> sv2) sv = sv2;
Printf ("\n arrange 2-legged %dmm dia bars at %fmm spacing", stirrupdia, sv);
getch();
}
```

26/8/19

Q: Computation of discharge over a rectangular notch using velocity of approach

```
# include <stdio.h>
# include <conio.h>
# include <math.h>
void main()
{
float length, height, Cd, area, Va, ha, Q, Q1, h2;
Clrscr();
Printf("Enter the values of length, height, Cd, area");
Scanf("%f %f %f %f", &length, &height, &Cd, &area);
Q = 1.705 * Cd * length * height * sqrt(height);
Va = Q/Area;
ha = (Va * Va) /(2x9.81);
h2 = height +ha;
Q1 = 1.705 * Cd * length * ((h2 * sqrt(h2)) - (ha * sqrt(ha)));
Printf("discharge without considering the velocity of approach= %f", Q);
Printf("\n discharge considering the velocity of approach= %f", Q1);

getch();
}
```

Output:

Enter the values of length, height, Cd, area

50 0.50 0.6 50

discharge without considering the velocity of approach= 18.08

discharge considering the velocity of approach= 18.419

Q: Calculation of normal depth & critical depth in a trapezoidal channel.

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
float b,yn,Q,z,s,n,k,yc,g,H,I,p;
clrscr();
printf("Enter the values of b,s,Q,z,ng");
scanf("%f %f %f %f %f %f",&b,&s,&Q,&z,&n,&g);
for (yn=0.1;yn<=100 ;yn+=0.1)
{
k=1/n*yn*(b+z*yn)*pow(yn*(b+z*yn)/b+z*yn*
                    sqrt(1+(z*z)),0.66)*pow(s,0.5);
p=(Q-k)/Q*100;
if (p>=1)
printf("normal depth is %f",yn);
}printf("percentage is %f",p);}
for(yc=0.1; yc<=1; yc+=0.1)
{
I=sqrt(pow(yc*(b+z*yc),3)/(b+z*2*yc));
H=Q-I/Q*(100);
if (H>=1)
printf("critical depth is %f",yc);
getch();
}
}
```

Output:

Enter the values of b, s, Q, z, n, g

30, 1/1500, 30, 2, 0.03, 9.81

Normal depth is 99.99046

Percentage is 99.8999

Critical depth is 0.999

26/8/19

Q: Design an irrigation channel using Kennedy's theory.

FS discharge = $14 m^3/s$.

Bed slope = 1 in 5000

Kutter's $N = 0.0225$

CVR (m) = 1

Side slope = $\frac{1}{2}:1$ (Horizontal : vertical)

Soln Assuming full supply depth = 1.8m.

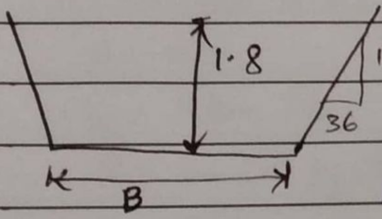
Velocity can be calculated from Kennedy's equation

$$V = 0.545\, m D^{0.64} = 0.546 \times 1 \times 1.8^{0.64} = 0.795\ m/s$$

from eqn, $Q = AV$

$$A = \frac{14}{0.795} = 17.6 m^2$$

$$A = BD + \frac{D^2}{2} = 17.6$$



$$1.8\, B + 3.24/2 = 17.6$$

$$B = 8.8 m.$$

$$R = \frac{A}{P_w} = \frac{17.6}{B + D\sqrt{5}} = \frac{17.6}{8.8 + 1.8\sqrt{5}} = 1.365$$

Chezy's constant can be calculated from Kutter's formula

$$C = \frac{23 + \frac{1}{0.0225} + \frac{0.000155}{0.0002}}{1 + \left(23 + \frac{0.000155}{0.002}\right)\frac{0.00225}{\sqrt{1.365}}} = 46.8$$

Now, using Chezy's formula

$$V = 46.8\sqrt{1.365 \times 0.0002} = 0.772\ m/s.$$

```
# include <stdio.h>
# include <conio.h>
# include <math.h>
void main()
{
float  Q, S, V, N, R, C, Vk, Qk, m, A, P, B, D;
clrscr();
Printf ("Enter the value of Q, S, N, m");
Scanf ("% f %f %f %f", &Q, &S, &N, &m);
For (D=0.1; D<= 100; D+=0.1)
{
V=0.546 * m * pow (D, 0.64);
A= Q/V;
B = A/D- D/2;
P= B+ D* sqrt(5);
R= A/P ;
C=46.8 ;
Vk = 46.8 * sqrt (R*S);
Qk = A*Vk;
If ((Qk /Q) >1)
Printf ("Depth is %f", D);
}
getch();
}
```

Output: Enter the values of Q, S, N, m.

14   1/5000   0.0225   1/2

Depth is 1.8

23/9/18

~~Q:~~ ~~Determination of pre and post jump depths from known specific energy values.~~

$$\frac{y_2}{y_1} = \frac{1}{2}\left(\sqrt{1+8F_1^2}-1\right)$$

$V = Q^2$

Assume the reasonable full supply depth

Kennedy's Theory

$V = 0.546 D^{0.64} \rightarrow (1)$

$Q = AV \rightarrow (2)$

$V = C\sqrt{RS} \rightarrow (3)$

Using eqn. (1) find value 'V'

Substitute 'V' in eqn. (2) & find 'A'

Assume side slope & from A, D find B

Calculate $R = A/P$

Using eqn. (3) find actual velocity 'V'

→ When assumed value of 'D' is correct, ~~find 'V' is correct~~ ~~if not~~ value of 'V'

If not assume some other value for 'D'.

$$E_1 = h_1 + \frac{V_1^2}{2g}$$

$E_1 = 4$

$V_1 = 6\ m/s$

$g = 9.8\ m/s^2$

$$h_1 = 4 - \frac{6^2}{2 \times 9.81} = 2.165\ m.$$

$$h_2 = \frac{h_1}{2} + \sqrt{\frac{h_1^2}{4} + \frac{2h_1V_1^2}{g}}$$

$$h_2 = \frac{2.165}{2} + \sqrt{\frac{(2.165)^2}{4} + \frac{2 \times 2.165 \times 6^2}{9.81}}$$

$$= 1.0825 + \sqrt{1.17 + 16.8}$$

$h_2 = 5.2m.$

Q: Estimation of Specific capacity and maximum pumping rate of a well.

```
# include <stdio.h>
# include <conio.h>
# include< math.h>
void main()
{
Float Q, ho, h, Sc, hw, max D, max P;
Printf (" Enter the values of Q, ho, h, hw");
Scanf(" %f %f %f %f", &Q, &ho, &h, &hw);
Sc = Q/(ho-h);
Max D = (hw-ho);
Max P= (hw-ho)*Sc;
Printf("max. drawdown & pumping rate are %f %f", maxD,
maxP);
getch();
}
```

Output:

Enter the values of Q, ho, h, hw

16, 10, 4, 15

Max. drawdown & pumping rate are 5, 13.33.

Q: Determination of pre & post jump depths from known specific energy values.

```
# include <stdio.h>
# include <conio.h>
# include <math.h>
voidmain()
{
Float e, h1, h2, v, g;
Printf ("enter the values of g,e,v");
Scanf("%f %f %f", &g, &e, &v);
h1= e- [(v*v) / (2*g)];
Printf("max height of pre jump %f", h1);
h2= (h1/2) + (sqrt({((h1*h1)/4) + ((2+h1*v*v)/g)));
Printf ("Post ht. of jump %f", h2);
getch();
}
```

Output:

Enter the values of g,e,v

9.81, 4 , 6

Max. height of pre jump
2.165138

Post ht. of jump 5.213292

Specific Capacity $(S_c) = \dfrac{Q}{h_0 - h}$

Q: Pumping rate

h_0: Depth of water table

h: Depth of static water level

$(h_0 - h)$: drawdown

depth of well: max. available drawdown

$= h_w$

Max. drawdown $= h_w - h_0$

Max. pumping rate $= (h_w - h_0)\, S_c$

Q: Flood routing using Muskingham's method.

x : Weighing factor — 0 to 0.5 usually 0.2

K - Strain constant

$I \rightarrow$ Stage $\rightarrow O$

$O_2 = C_0 I_2 + C_1 I_1 + C_2 O_1$

$O \rightarrow$ Output

$I \rightarrow$ Input

$C_0 = \dfrac{(0.5\Delta t - Kx)}{D}$

I_1 & O_1 at time 1

I_2 & O_2 at time 2

$C_0 + C_1 + C_2 = 0$

$C_1 = (Kx + 0.5\Delta t)/D$

$C_2 = (K - Kx - 0.5\Delta t)/D$

$D = K - Kx + 0.5\Delta t$

If above is not satisfied, adjustment is made for higher 'c' value by subtracting

Time	0	3	6	9	12	15	18
Inflow	1	3	9	15	13	10	6

Time	0	3	6	9	12	15	18
I	1	3	9	15	13	10	6
O	1	1.5	3.7	9.1	13.7	12.6	9.8

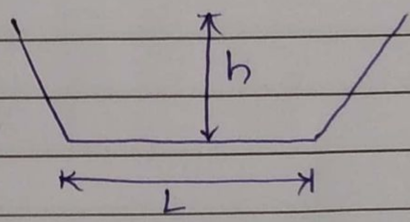
Q: Design a trapezoidal notch canal fall.

Q_F : Full supply discharge = 4 m^3/sec
Bed width = 6 m.
Full supply depth = 1.5 m.
Half supply depth = 1.0 m.

Provide 'n' no. of notches.
Then Q per notch = $\frac{Q_F}{n}$

$$Q = \frac{8}{15} C_d \sqrt{2g} \tan\left(\frac{\theta}{2}\right) h^{5/2} + \frac{2}{3} C_d L \sqrt{2g}\, h^{3/2}$$

$C_d = 0.7$
$g = 9.81$
$h = 1.5m$ (or) $1m$.
$A = ?$



Full supply

$$4 = \frac{8}{15} \times 0.7 \times \sqrt{2\times9.81} \times \tan\left(\frac{\theta}{2}\right) (1.5)^{5/2} + \frac{2}{3} \times 0.7 \times L \times \sqrt{2\times9.81} \times (1.5)^{3/2}$$

Half supply

$$2 = \frac{8}{15} \times 0.7 \times \sqrt{2\times9.81} \times \tan\left(\frac{\theta}{2}\right) (1)^{5/2} + \frac{2}{3} \times 0.7 \times L \times \sqrt{2\times9.81} \times (1)^{3/2}$$

$$4 = 4.56 \tan\left(\frac{\theta}{2}\right) + 2.06 L \rightarrow (1)$$

$$2 = 1.65 \tan\left(\frac{\theta}{2}\right) + 2.06 L \rightarrow (2)$$

$$\boxed{\theta = 69.0}$$

$$\boxed{L = 0.42}$$

Scanned by CamScanner

Q: Compute distribution of increment in vertical stress due to applied point load on a
a) Horizontal plane
b) Vertical plane. Using the computed values, plot the distribution utilizing VC as front end tool.

$$\sigma_z = \frac{3Q}{2\pi} \cdot \frac{1}{Z^2}\left[\frac{1}{\left[1+\left(\frac{r}{z}\right)^2\right]^{5/2}}\right]$$

$$\sigma_z = I_B \cdot \frac{Q}{Z^2}$$

$$I_B = \frac{3}{2\pi} \frac{1}{\left[1+\left(\frac{r}{z}\right)^2\right]^{5/2}}$$

a) Horizontal Plane:
z : Constant
for ex. take 1m or 2m

Input Parameters: Q, Z, r

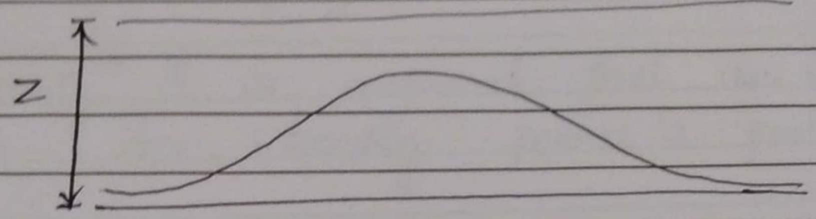
z : constant, = Assume 2m or 3m etc.
r : constant 0, 0.25, 0.5, 0.75, 1, 1.25m, ...

Q: Given value of point load (assume 500kN or 1000kN, etc.)

1) Assume z & Q
2) Take values of r
3) Cal. r/z & then I_B for different 'r' values
4) Cal. σ_z using $\sigma_z = I_B \cdot \frac{Q}{z^2}$

5> Finally, we get different σ_z values for different 'r' values

6> Plot the distribution.



b> <u>Vertical Plane:</u>

<u>Input Parameters:</u> Q, Z, r

1> Assume r & θ_r

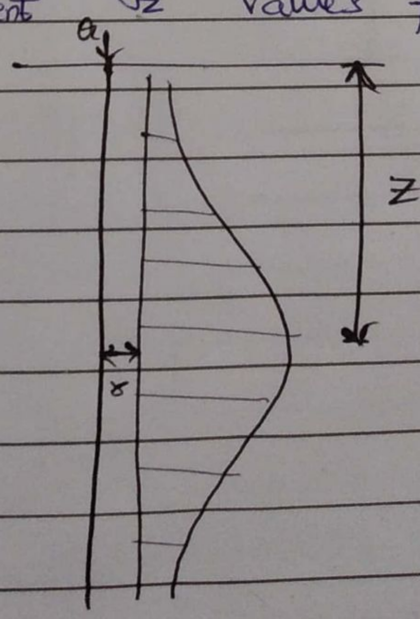
2> Take different values for 'z'

3> Cal. r/z & I_B for different 'z' values

4> Cal. σ_z using $\sigma_z = I_B \cdot \frac{\theta_r}{z^2}$

5> Finally, we get different σ_z values for different 'r' values.

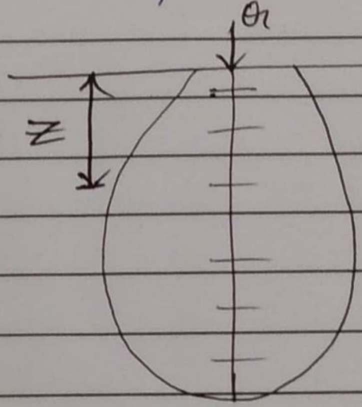
6> Plot the distribution.



Q: Compute the values of a pressure bulb & using the values, plot pressure bulb utilizing VC as front end tool.

Given, σ_z, I_B, $\frac{Q}{Z^2}$

Generally, it is assumed that an isobar of 0.1Q for intensity forms a pressure bulb



Take $\sigma_z = 0.1Q$

$$0.1Q = I_B \cdot \frac{Q}{Z^2}$$

$$I_B = 0.1Z^2$$

Input Parameter: Q, Z

1) Assume 'Q' values & different values for Z (0, 0.25, 0.5, 0.75, --- Z_{max})

2) Calculate I_B using $I_z = 0.1Z^2$

3) Calculate $\frac{r}{Z}$ from $I_B = \frac{3}{2\pi}\left[\frac{1}{1+\left(\frac{r}{Z}\right)^2}\right]^{5/2}$

&

4) Calculate 'r' from $\frac{r}{Z}$ values.

Zmax Calculations

When $r = 0 \Rightarrow Z = Z_{max}$

$$\sigma_z = \frac{3Q}{2\pi} \cdot \frac{1}{Z^2}\left[\frac{1}{1+\left(\frac{r}{Z}\right)^2}\right]^{5/2}$$

Substitute $\sigma_z = 0.1Q$ & $r = 0$

$$\Rightarrow 0.1 = \frac{3}{2\pi} \cdot \frac{1}{(Z_{max})^2} \Rightarrow \boxed{Z_{max} = 2.185\,m.}$$