**PROGRAM ONE ALGORITHM**

Step 1: Start

Step 2: Declare variables db2, rs1, rs2, rs3, j, d, Length and breadth.

Step 3: Read variables Length and breadth.

Step 4: Calculate Initial Area as Ao

Pile =22/7

Ao = Length \* breadth.

Step 5: Calculate Final Area as An

Pile =22/7

An=(pile\*(12^2))/4

Step 6: Calculate Final Stand as X

X = (Log(Ao) - Log(An)) / Log(1.29)

Step 7: Determine the Even and Odd of Final Stand as Y

v = Int(X) / 2

If v <> Int(v) Then

Y = Int(X)

Else

Y = Round(X, 0)

End If

Step 8: Loop Through the Final Stand Y until Y = 1

If y <> 1

Create connection to Database

Save Y as Stand to database

Repeat Step 8

Y= Y-1

Else

Go to Step 9

End If

Step 9: Stop

**PROGRAM TWO AND THREE ALGORITHM**

Step 1: Start

Step 2: Declare variables db2, rs1, rs2, rs3, j, d, b and p.

Step 3: Initialize the loop value j = 19

Step 4: Create connection to Database

Step 5: Start Looping through Records in Database

Step 6: if j = 19

Save Area as 113.47 to database

Else if j = 1

` Save Area as 0 to database

Else if j = 2

Save Area as 0 to database

Else

d = j - 1

b = Area at(d) \* Elongation at (d)

Save Area as b to database

End if

Step 7: Decrement loop value j = j - 1

Step 8: Next loop of Database

Step 9: End loop of Database

Step 10: Stop

**PROGRAM FOUR ALGORITHM**

Step 1: Start

Step 2: Declare variables db2, rs1, rs2, Percentage, rs3, j and p.

Step 3: Initialize the loop value j = 19

Step 4: Create connection to Database

Step 5: Start Looping through Records in Database

Step 6: tt = j - 1

Step 7: if tt = 0

Percentage = ((14400 - Area at (j))/ Area at (j)) \* 100

Save Percentage as Percentage to database

Else

Percentage = ((Area at (tt) - Area at (j))/ Area at (j)) \* 100

Save Percentage as Percentage to database

End if

Step 8: loop value = loop value - 1

Step 9: Next loop of Database

Step 10: End loop of Database

Step 11: Stop

**PROGRAM FIVE ALGORITHM**

Step 1: Start

Step 2: Declare variables db2, rs1, rs2, Length, rs3, j and p.

Step 3: Initialize the loop value j = 1

Step 4: Create connection to Database

Step 5: Start Looping through Records in Database

Step 6: if J <> 1

P = j + 1

Length = Length at (p) \* Elongation at (j)

Save Length as Length to database

Else

Length = 12 \* Elongation at (j)

Save Length as Length to database

End if

Step 7: j = j + 1

Step 8: Next loop of Database

Step 9: End loop of Database

Step 10: Stop

**PROGRAM SIX TO NINE ALGORITHM**

Step 1: Start

Step 2: Declare variables db2, rs1, rs2, Entrance\_h, contact\_friction, frictional\_contact\_area, deltah, actual\_spread, pico, deltah, Exit\_h, Exit\_b, Entrance\_b, BValue, Area, Chnage\_b, k and p.

Step 3: Initialize the loop value p = 1

Step 4: Read variables contact\_friction, entrance\_b and frictional\_contact\_area.

Step 5: Initialize variables Entrance\_h = 120 and Entrance\_b = 120.

Step 6: Create connection to Database

Step 7: Start Looping through Records in Database

Step 8: Determine the Even, Odd Stand, stand one and stand two and Stand 19

v = Int(p) / 2

Check for Stand Equal One

If (p = 1) Then

pico = contact\_friction \* (1.05 - (0.0005 \* (1250)))

pico = pico \* pico

deltah = contact\_friction \* pico \* 250

Exit\_h = Entrance\_h – deltah

Chnage\_b = 0.4 \* (deltah / 120) \* ((250 \* deltah) ^ 0.5)

actual\_spread = 120 + Chnage\_b

Exit\_b = actual\_spread

BValue = actual\_spread + 12

f = frictional\_contact\_area

Area = actual\_spread \* Exit\_h \* f

K = p + 1

Save Entrance\_h as Entrance\_h to database at Stand (k)

Save Entrance\_b as Entrance\_b to database at Stand (k)

Save Exit\_h as Exit\_h to database at Stand (k)

Save Exit\_b as Exit\_b to database at Stand (k)

Save b\_Value as b\_Value to database at Stand (k)

Save h\_Value as h\_Value to database at Stand (k)

Save BValue as BValue to database at Stand (k)

Save Delta\_H as Delta\_H to database at Stand (k)

Save Area as Area to database at Stand (k)

Save Entrance\_h as Entrance\_h to database at Stand (p)

Save Entrance\_b as Entrance\_b to database at Stand (p)

Save Entrance\_Area as Entrance\_Area to database at Stand (p)

Check for Stand Equal Two

Else If (p = 2) Then

pico = contact\_friction \* (1.05 - (0.0005 \* (1225)))

pico = pico \* pico

deltah = contact\_friction \* pico \* 250

Exit\_h = Entrance\_h – deltah

Chnage\_b = 0.4 \* (deltah / 120) \* ((250 \* deltah) ^ 0.5)

actual\_spread = 120 + Chnage\_b

Exit\_b = actual\_spread

BValue = actual\_spread + 12

f = frictional\_contact\_area

Area = actual\_spread \* Exit\_h \* f

K = p + 1

Save Entrance\_h as Entrance\_h to database at Stand (k)

Save Entrance\_b as Entrance\_b to database at Stand (k)

Save Exit\_h as Exit\_h to database at Stand (k)

Save Exit\_b as Exit\_b to database at Stand (k)

Save b\_Value as b\_Value to database at Stand (k)

Save h\_Value as h\_Value to database at Stand (k)

Save BValue as BValue to database at Stand (k)

Save Delta\_H as Delta\_H to database at Stand (k)

Save Area as Area to database at Stand (k)

Save Entrance\_h as Entrance\_h to database at Stand (p)

Save Entrance\_b as Entrance\_b to database at Stand (p)

Save Entrance\_Area as Entrance\_Area to database at Stand (p)

Check for Stand Equal Two

Else If (p = 19) Then

Initialize Variables si = Sin(45), co = Cos(45), b = 12.1, b2 = b \* b, S = 1.6, s2 = S \* S, d19 = entrance\_b, d2 = d \* d, r = 6.01 ^ 2

fu = (b2 + s2 + (4 \* r2)) - (4 \* r \* ((S \* si) + (b \* co)))

fd3 = 8 \* r - (4 \* (S \* si + b \* co))

final = fu / fd3

Save Exit\_h as d19 to database at Stand (p)

Save Entrance\_h as Entrance\_h to database at Stand (p)

Save Exit\_b as d19 to database at Stand (p)

Save Exit\_b as Exit\_b to database at Stand (p)

Save Entrance\_b as d19 to database at Stand (p)

Check for Odd Stand

Else If (v = Int(v) And (P <> 1 And P <> 2 And P <> 19)) Then

J = Area at stand (p)

K2 = constanteven at stand (p)

K18 = constant at stand (p)

h18 = k2 \* d19

m18 = k18 \* h18

ans = ((b18 \* b18) + (h18 - m18) ^ 2)/(4\*(h18 - m18))

finalB = 2 \* [(((h18 - S18) \* ans)-((h18 - S18) / 2)^2)^0.5]

k = p + 1

Save Radius as ans to database at Stand (p)

Save BValue as finalB to database at Stand (p)

Save b\_Value as b18 to database at Stand (p)

Save h\_Value as h18 to database at Stand (p)

Save Exit\_h as h18 to database at Stand (p)

Save Exit\_b as b18 to database at Stand (p)

Save Entrance\_h as h18 to database at Stand (k)

Save Entrance\_b as b18 to database at Stand (k)

Set d19 = 0

Check for Even Stand

ElseIf (v <> Int(v) And (P <> 1 And P <> 2 And P <> 19)) Then

J = Area at stand (p)

K1 = constant at stand (p)

bg1 = 24 \* K1 \* j

d19 = [(24 \* K1 \* j ) /(3 + (16 \* (K1^2))]^0.5

h17 = K1 \* d19

finalr17 = (h17 ^ 2) + (d19 ^ 2) / (4 \* d19)

finalB = d19 + 1

k = p + 1

Save Radius as finalr17 to database at Stand (p)

Save BValue as finalB to database at Stand (p)

Save b\_Value as d19 to database at Stand (p)

Save h\_Value as h17 to database at Stand (p)

Save Exit\_h as h17 to database at Stand (p)

Save Exit\_b as d19 to database at Stand (p)

Save Entrance\_h as h17 to database at Stand (k)

Save Entrance\_b as d19 to database at Stand (k)

Else

End If

Step 9: p = p + 1

Step 10: Next loop of Database

Step 11: End loop of Database

Step 12: Stop

**PROGRAM TEN ALGORITHM**

Step 1: Start

Step 2: Declare variables db2, rs1, rs2, Deduction, rs3, j and p.

Step 3: Initialize the loop value j = 19

Step 4: Create connection to Database

Step 5: Start Looping through Records in Database

Step 6: if exit\_h at (j) > entranhe\_h at (j)

P = j – 1

Deduction = 0

Deduction = b\_Value at (p) - Exit\_h at (j)

Save H as Deduction to database

Else

Deduction = 0

Deduction = Entrance\_h at (j) - Exit\_h at (j)

Save H as Deduction to database

End if

Step 7: j = j + 1

Step 8: Next loop of Database

Step 9: End loop of Database

Step 10: Stop

**PROGRAM ELLEVEN AND TWELVE ALGORITHM**

Step 1: Start

Step 2: Declare variables db2, rs1, rs2, Dw, rs3, total\_angle\_of\_bite\_degree, j, Rw and angle\_of\_bite

Step 3: Initialize the loop value j = 19

Step 4: Create connection to Database

Step 5: Start Looping through Records in Database

Step 6: Dw = (Diameter\_roll at (j) – (Area at (j)/ b\_Value at (j)) + SValue at (j)

Step 7: Rw = Dw / 2

Step 8: angle\_of\_bite = (H at (j) / Rw ) ^ 0.5

Step 9: total\_angle\_of\_bite\_degree = 57.5 \* angle\_of\_bite

Step 10: Save as Dw to database

Step 11: Save Effective\_Radius as angle\_of\_bite to database

Step 12: Save as Dw to database

Step 13: Save as total\_angle\_of\_bite\_degree to database

Step 14: j = j + 1

Step 15: Next loop of Database

Step 16: End loop of Database

Step 17: Stop

**PROGRAM THIRTEEN ALGORITHM**

Step 1: Start

Step 2: Declare variables db2, rs1, rs2, Deduction, rs3, j and p.

Step 3: Initialize the loop value j = 19

Step 4: Create connection to Database

Step 5: Start Looping through Records in Database

Step 6: if J = 19

pile = 22 / 7

Vr = 15

Nr = (Vr \* 60 \* 1000) / (pile \* Effective\_Diameter at (j))

Nm = Nr \* Transmission\_Ratio at (j)

Q\_K = Effective\_Diameter at (j) \* Nr \* Area at (j)

Save moto\_revolution\_Nm as Nm to database

Save revolution\_of\_roll\_Nm as Nr to database

Save rolling\_constant as Q\_K to database

Save speed as Vr to database

Else

pile = 22 / 7

Vr = 15

Nr = Q\_K / (Effective\_Diameter at (j) \* Area at (j))

Nm = Nr \* Transmission\_Ratio at (j)

Vr = (Nr \* pile \* Effective\_Diameter at (j))/ 60000

Save moto\_revolution\_Nm as Nm to database

Save revolution\_of\_roll\_Nm as Nr to database

Save rolling\_constant as Q\_K to database

Save speed as Vr to database

End if

Step 7: j = j + 1

Step 8: Next loop of Database

Step 9: End loop of Database

Step 10: Stop

**PROGRAM FOURTEEN ALGORITHM**

Step 1: Start

Step 2: Declare variables db2, rs1, rs2, K1, Too, Tn, rs3, j and p.

Step 3: Initialize the loop value j = 19, Ho = 120

Step 4: Create connection to Database

Step 5: Start Looping through Records in Database

Step 6: if J = 19

K1 = ((1250 - 800) / (Ho - h\_Value at (j)) \* Ho \* h\_Value at (j)

Too = T1 + (K1 / h\_Value at (1))

Tn = Too - (K1 / h\_Value at (j))

Save Temperature as Tn to database

Else

Tn = Too - (K1 / h\_Value at (j))

Save Temperature as Tn to database

End if

Step 7: j = j - 1

Step 8: Next loop of Database

Step 9: End loop of Database

Step 10: Stop

**PROGRAM FIFTEEN ALGORITHM**

Step 1: Start

Step 2: Declare variables db2, rs1, rs2, K1, Mn, C, rs3, j, step1, step2, Zigma, Acstep1, Acstep2, Pm, Preal, Kf, Ac and p.

Step 3: Initialize the loop value j = 19

Step 4: Read variables Mn and C.

Step 5: Create connection to Database

Step 6: Start Looping through Records in Database

Step 7: Determine the Even, Odd Stand and stand one

v = Int(J) / 2

Check for Odd and Stand not Equal One

If (v <> Int(v) And J<>1) Then

Zigma = (14 - (0.01 \* Temperature at (j))) \* (1.4 + C + Mn)

elongation\_n = 0.01 \* (14 - (0.01 \* Temperature at (j))

coefficient\_friction = (1.05 - (0.0005 \* Temperature at (j)))

step1 = (1.6 \* coefficient\_friction \* (Delta\_H at (j) \* Effective\_Radius at (j)) ^ 0.5) - (1.2 \* Val(RS11!Delta\_H))

step2 = step1 / (h\_Value at (j) + BValue at (j-1))

Kf = 1 + step2

Acstep1 = (Delta\_H at (J) \* Effective\_Radius at (J)) ^ 0.5

Acstep2 = (BValue at (j-1) + BValue at (J)) / 2

Ac = Acstep1 \* Acstep2

Pm = Kf \* (Zigma + (v \* elongation\_n))

Preal = Val(Pm) \* Val(Ac)

P = (Preal \* 9.81) / 1000000

Save Rolling\_Pressure as Pm to database

Save Rolling\_Load\_KgF as Preal to database

Save Rolling\_Load\_MN as P to database

Save Contact\_Area as Ac to database

Check for Even and Stand Equal One

ElseIf (v = Int(v) And J = 1) Then

Zigma = (14 - (0.01 \* Temperature at (j))) \* (1.4 + C + Mn)

elongation\_n = 0.01 \* (14 - (0.01 \* Temperature at (j))

coefficient\_friction = (1.05 - (0.0005 \* Temperature at (j)))

step1 = (1.6 \* coefficient\_friction \* (Delta\_H at (j) \* Effective\_Radius at (j)) ^ 0.5) - (1.2 \* Val(RS11!Delta\_H))

step2 = step1 / (h\_Value at (j) + BValue at (j-1))

Kf = 1 + step2

Acstep1 = (Delta\_H at (J) \* Effective\_Radius at (J)) ^ 0.5

Acstep2 = (BValue at (j-1) + BValue at (J)) / 2

Ac = Acstep1 \* Acstep2

Pm = Kf \* (Zigma + (v \* elongation\_n))

Preal = Val(Pm) \* Val(Ac)

P = (Preal \* 9.81) / 1000000

Save Rolling\_Pressure as Pm to database

Save Rolling\_Load\_KgF as Preal to database

Save Rolling\_Load\_MN as P to database

Save Contact\_Area as Ac to database

Else

End If

Step 8: j = j + 1

Step 9: Next loop of Database

Step 10: End loop of Database

Step 11: Stop

**PROGRAM SIXTEEN ALGORITHM**

Step 1: Start

Step 2: Declare variables db2, rs1, rs3, rs2, pile, Work\_Done, j and p.

Step 3: Initialize the loop value j = 1

Step 4: Create connection to Database

Step 5: Start Looping through Records in Database

Step 6: pile = 22 / 7

Step 6: Work\_Done = 4 \* Rolling\_Load\_MN at (j) \* pile \* 0.5 \* (Effective\_Radius at (j) / 1000)) \* angle\_of\_bite at (j)

Step 7: Save Work\_Done as Work\_Done to database

Step 7: j = j + 1

Step 8: Next loop of Database

Step 9: End loop of Database

Step 10: Stop

**PROGRAM SEVENTEEN ALGORITHM**

Step 1: Start

Step 2: Declare variables db2, rs1, rs3, rs2, pile, Work\_Done, j and p.

Step 3: Initialize the loop value j = 1

Step 4: Create connection to Database

Step 5: Start Looping through Records in Database

Step 6: pile = 22 / 7

Step 7: alpha = 0.5 \* (Effective\_Radius at (j) / 1000) \* angle\_of\_bite at (j)

Step 8: Dynamic\_Torgue = Rolling\_Load\_KgF at (j) \* 2 \* alpha \* 0.001

Step 9: Frictional\_Torgue = Rolling\_Load\_KgF at (j) \* 0.03 \* (D\_Value at (j) / 1000) \* 0.001

Step 10: Rolling\_Torgue = Dynamic\_Torgue at (j) + Frictional\_Torgue at (j)

Step 11: Rolling\_Power\_hp = (Rolling\_Torgue at (j) \* revolution\_of\_roll\_Nm at (j)) / 0.716)

Step 12: Static\_Power = ((Rolling\_Torgue at (j) \* revolution\_of\_roll\_Nm at (j)) / 0.93) \* 1.025

Step 13: Save Dynamic\_Torgue as Dynamic\_Torgue to database

Step 14: Save Frictional\_Torgue as Frictional\_Torgue to database

Step 15: Save Rolling\_Torgue as Rolling\_Torgue to database

Step 16: Save Rolling\_Power\_hp as Rolling\_Power\_hp to database

Step 17: Save Static\_Power as Static\_Power to database

Step 18: j = j + 1

Step 19: Next loop of Database

Step 20: End loop of Database

Step 21: Stop