

```
// ~~~
// Mortgage calculation
// ~~~

in funds sale_value;

in out funds payment_down;
in float percent_down;
percent_down /= 100;

if(!payment_down)
    payment_down = sale_value * percent_down;

out funds financed_value = sale_value - payment_down;

in float irate_y;
irate_y /= 100;
float irate_m = irate_y / 12;

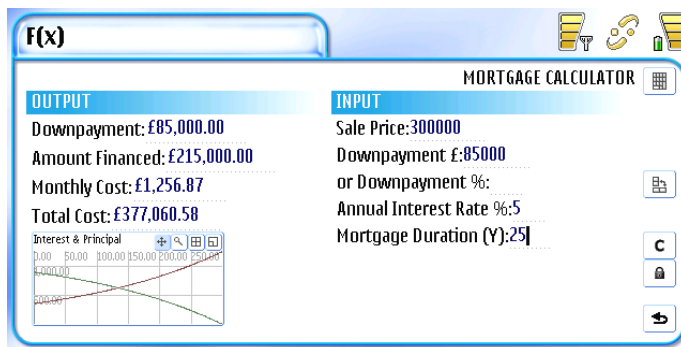
in float term_y;
float term_m = term_y * 12;

funds principal_paid[term_m],
    interest_paid[term_m],
    balance[term_m];

float denom = (1+irate_m)**-term_m;
out funds payment_m = financed_value * (irate_m/(1-denom));
out funds total = payment_m * term_m;
funds principal = financed_value;
for(int i = 0; i < term_m; i++)
{
    interest_paid[i] = principal * irate_m;
    principal_paid[i] = payment_m - interest_paid[i];
    balance[i] = principal - principal_paid[i];
    principal = balance[i];
}

out funds g[2][];
g[0] = interest_paid;
g[1] = principal_paid;
@g graph1d 0 term_m min(g) max(g) "Interest & Principal";

@sale_value "Sale Price";
@percent_down "or Downpayment %";
@payment_down "Downpayment";
@financed_value "Amount Financed";
@irate_y "Annual Interest Rate %";
@term_y "Mortgage Duration (Y)";
@term_m "Mortgage Duration (M)";
@total "Total Cost";
@payment_m "Monthly Cost";
```



```
// ~~~
// Family history questionnaire
// ~~~

int iCount = 6;
in str client;
in bool q1,q2,q3,q4,q5,q6;

@client string "Client Name";

@q1 check "Diabetes";
@q2 check "Heart Disease";
@q3 check "Mental Illness";
@q4 check "High BP";
@q5 check "Cancer";
@q6 check "Epilepsy/Seisure";

out str msg = "Family History for: "+client+"\n";

for(int i = 1; i <= iCount; i++)
{
    msg += vtitle("q"+i)+": "
        +(vvalue("q"+i) ? "YES" : "NO")+"\n";
}

@msg send "Send History Info";
```



```
// ~~~
// Head loss in pipe
// ~~~
// PREPARED By Eng. ERHAN ZORER
```

```
real g=9.8;
in real Q,k,D,L,vis;
in int NoBend22,NoBend45,NoBend90,NoCV,NoV,NoIn,NoOut;

real Qs,Area,Re,a=1e-7,b=1,fa,fb,fλ,tol=1e-10,K22=0.1,
K45=0.17,K90=0.29,KCV=1.5,KV=0.3,KIn=0.5,KOut=1,Ktot;

out real V,λ,Hf,Hm,Htot;

@Q unit flowvolume MCuDay "Flow";
@k unit length mm "Absolute roughness";
@D unit length m "Diameter of pipe";
@L unit length m "Length of pipe";
@vis "kinematic viscosity";
@NoBend22 "No of 22.5 Bend";
@NoBend45 "No of 45 Bend";
@NoBend90 "No of 90 Bend";
@NoCV "No of Check Valves";
@NoV "No of valves";
@NoIn "No of Inlet";
@NoOut "No of Outlet";

Qs=Q*MCuDay2MCuSec;
Area=PI*(D)**2/4;
V=Qs/Area;
Re=D*V/vis;

fa=2*log10(k/(3.71*(D*1000))+2.51/(Re*sqrt(a)))+1/sqrt(a);
fb=2*log10(k/(3.71*(D*1000))+2.51/(Re*sqrt(b)))+1/sqrt(b);

for(int itr = 0; itr < 100; itr++)
{
    λ=(a+b)/2;
    fλ=2*log10(k/(3.7*D*1000)+2.51/(Re*sqrt(λ)))+1/sqrt(λ);

    if (fa*fλ<=0) { b=λ; fb=fλ;}
    else {a=λ; fa=fλ;}
}

if((b-a)>tol && abs(fλ)>tol) error(E_RANGE);

Hf=λ*L/D*(V**2/(2*g));
Ktot=K22*NoBend22+K45*NoBend45+K90*NoBend90+KCV
*NoCV+KV*NoV+KIn*NoIn+KOut*NoOut;
Hm=Ktot*(V**2/(2*g));
Htot=Hf+Hm;

@V unit speed MSec "velocity in pipe";
@λ "Friction Factor";
@Hf unit length m "Frictional Head Loss";
@Hm unit length m "Minor Head Loss";
@Htot unit length m "Total Head Loss";
```