//

// main.cpp

// Xcode

//

// Created by Vinod Myll Mylsamy on 5/1/15.

// Copyright (c) 2015 Vinod Myll Mylsamy. All rights reserved.

//

#include <iostream>

#include <stdio.h>

#include <math.h>

#include <sys/time.h>

using namespace std;

int main(int argc, const char \* argv[]) {

// insert code here...

std::cout << "Hello, World!\n";

int i,j,k;

double x = -1.2,y = 1;

double xnew[2];

double x[2]= {-1.2,1};

double xdiff[2];

double gradient[2];

double hessian[2][2];

double hessianInverse[2][2];

double hessianInv[2][2];

double s[2];

double f;

for(k=0;k<20;k++)

{

if (k == 0)

{

x = -1.2;

y = 1;

xold[0]= -1.2;

xold[1] =1;

}

else

{ x =0;

y =0;

xold [0] = 0;

xold[1] = 0;

x = xnew[0];

y = xnew[1];

xold[0]= xnew[0] ;

xold[1] = xnew[1];

}

f = 0;

f = ((1- x)\*(1- x)) + (100\*(y-(x\*x))\*(y-(x\*x)));

printf("The value of the function is %f\n",f);

double g1 = (double)((-2.0\*(1.0-x))-((400.0\*x)\*(y-(x\*x))));

double g2 =(double)(200\*(y-(x\*x)));

double h1 = (double)(2-(400\*y)+(1200\*x\*x));

double h2 = (double)(-400 \*x);

double h3 = (double)(-400 \* x);

double h4 = (double)(200);

double determinant = 1/((h1 \* h4)-(h2\*h3));

gradient[0] = g1;

gradient[1] = g2;

hessian[0][0]=h1;

hessian[0][1]=h2;

hessian[1][0]= h3;

hessian[1][1]=h4;

hessianInv[0][0]=h4;

hessianInv[0][1]=-h3;

hessianInv[1][0]=-h2;

hessianInv[1][1]=h1;

printf("Gradient\n");

for (i=0; i<=1;i++)

{

printf("%f\n",gradient[i]);

}

printf("HessianMatrix\n");

for(i=0; i<=1; i++)

{

for(j=0;j<=1;j++)

{

printf("%f\t",hessian[i][j]);

}

printf("\n");

}

printf("HessianInv\n");

for(i=0; i<=1; i++)

{

for(j=0;j<=1;j++)

{

printf("%f\t",hessianInv[i][j]);

}

printf("\n");

}

printf("HessianInverse\n");

for(i=0; i<=1; i++)

{

for(j=0;j<=1;j++)

{

hessianInverse[i][j] = determinant \* hessianInv[i][j];

printf("%f\t",hessianInverse[i][j]);

}

printf("\n");

}

for(i=0; i<=1; i++)

{ s[i] = 0 ;

for(j=0;j<=1;j++)

{

s[i] = s[i]+ (hessianInverse[i][j] \* (- gradient[j]));

}

printf("\n");

}

printf("The solution S \n");

for(i=0; i<=1; i++)

{

printf("%f\n", s[i]);

}

printf("The Value of X is after %d iteration \n", k);

for(i=0; i<=1; i++)

{

xnew[i] = 0;

xnew [i] = s[i] + xold [i];

}

for(i=0; i<=1; i++)

{

printf("%f\n", xnew[i]);

}

for(i=0; i<=1; i++)

{

xdiff[i] = xnew[i] – x[i];

}

for(i=0; i<=1; i++)

{

x[i] = xnew[i];

}

}

return 0;

}