#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <stdbool.h>

struct point {

double x;

double y;

};

double

f(struct point z)

{

//return (z.x\*z.x + 4\*z.y\*z.y + 10);

//return (10\*z.x\*z.x - z.y\*z.y\*z.y);

return (exp(z.x\*z.x + z.y\*z.y) + 2\*z.x - 3.5\*z.y);

}

struct point grad(struct point z)

{

struct point t;

//t.x = 2\*z.x;

//t.y = 8\*z.y;

//t.x = 20\*z.x;

//t.y = -3\*z.y\*z.y;

t.x = 2\*z.x\*exp(z.x\*z.x + z.y\*z.y) + 2;

t.y = 2\*z.y\*exp(z.x\*z.x + z.y\*z.y) - 3.5;

return t;

}

int

main(void)

{

double eps;

double step = 10;

struct point old;

struct point new;

struct point cur;

int i = 1;

bool flag = true;

printf("Введите начальное приближение\n");

scanf("%lf %lf",&old.x,&old.y);

printf("Введите точность\n");

scanf("%lf",&eps);

for(i = 1; i < 1001 && flag; i++){

cur = grad(old);

new.x = old.x - step\*cur.x;

new.y = old.y - step\*cur.y;

flag = fabs(f(old) - f(new)) > eps;

while(f(new) > f(old) && flag && step){

step/=5;

new.x = old.x - step\*cur.x;

new.y = old.y - step\*cur.y;

flag = fabs(f(old) - f(new)) > eps;

}

flag = (f(old) - f(new) > eps);

old.x = new.x;

old.y = new.y;

}

printf("Примерная точка минимума (%lf , %lf),\n значение функции в этой точке: %lf\n количество итераций : %d\n",old.x,old.y,f(new),i);

return 0;

}