**Покоординатный спуск:**

*#include <stdio.h>*

*#include <stdlib.h>*

*#include <math.h>*

*#define sqr(x) ((x)\*(x))*

*double f(double\*);*

*int main(){*

*double \*x, \*df, h, eps = 0.0001, f0, df1, s, grad, dx = 0.0001; //Fixed eps to 0.0001*

*int i, m, t, n = 2; // Fixed n to 2*

*x = (double\*) malloc(sizeof(double)\*n);*

*df = (double\*) malloc(sizeof(double)\*n);*

*printf("Величина точности eps установлена равной = %lf\n",eps);*

*printf("Введите величину шага h = ");*

*scanf("%lf",&h);*

*printf("Введите предельное число итераций m = ");*

*scanf("%d",&m);*

*puts("Введите начальное значение переменных");*

*for(i=0;i<n;i++){*

*printf("x[%d] = ",i+1);*

*scanf("%lf",&x[i]);*

*}*

*t = 0;*

*do{*

*f0 = f(x);*

*s = 0;*

*t++;*

*for(i=0;i<n;i++){*

*x[i] += dx;*

*df[i] = (f(x)-f0)/dx;*

*x[i] -= dx;*

*}*

*for(i=0;i<n;i++){*

*x[i] -= h\*df[i];*

*}*

*df1 = f(x)-f0;*

*if(df1 > 0) h /= 2;*

*for(i=0;i<n;i++){*

*s += sqr(df[i]);*

*}*

*grad = sqrt(s);*

*} while(grad > eps && t <= m);*

*if(grad < eps && t <= m){*

*puts("Результаты оптимизации");*

*for(i=0;i<n;i++){*

*printf("\nx[%d] = %5.10lf",i+1,x[i]);*

*}*

*printf("\nзначение функции цели = %5.10lf",f0);*

*}*

*else{*

*puts("Решение не найдено");*

*puts("Уменьшите/увеличите величину шага или увеличьте предельное число итераций");*

*}*

*fflush(stdin);*

*getchar();*

*return 0;*

*}*

*double f(double \* x){*

*return 4 \* pow(x[0], 2) + 2 \* x[0] \* x[1] + 5 \* pow(x[1], 2) + 9 \* x[0];*

*}*

**Ньютон-рафсон:**

***#include <stdio.h>***

***#include <stdlib.h>***

***#include <math.h>***

***#define sqr(x) ((x)\*(x))***

***const double dx = 0.0001;***

***double \*df,\*\*d2f;***

***double f(double \* x){***

***return 4 \* pow(x[0], 2) + 2 \* x[0] \* x[1] + 5 \* pow(x[1], 2) + 9 \* x[0];***

***}***

***void invert(int n, int &q);***

***void for\_mat\_d2f(int n, double &grad, double &h, double \*x);***

***int main(){***

***int i,j,n = 2, q;***

***double eps = 0.0001, grad, s, h, \*x;***

***d2f = (double\*\*)malloc(sizeof(double\*)\*n);***

***for(i=0;i<n;i++)***

***d2f[i] = (double\*)malloc(sizeof(double)\*n);***

***x = (double\*) malloc(n\*sizeof(double));***

***df = (double\*) malloc(n\*sizeof(double));***

***// Инициализируем начальное значение переменных. Это может быть любые ваши числа.***

***x[0] = 1.0;***

***x[1] = 1.0;***

***do{***

***for\_mat\_d2f(n, grad, h,x);***

***invert(n,q);***

***if(q)***

***{***

***puts("Определитель равен нулю");***

***fflush(stdin);***

***getchar();***

***return 0;***

***}***

***for(i=0;i<n;i++)***

***{***

***s=0;***

***for(j=0;j<n;j++)***

***s+= d2f[i][j]\*df[j];***

***x[i] -= h\*s;***

***}***

***} while (grad >= eps);***

***puts("\nРезультаты оптимизации:\n");***

***for(i=0;i<n;i++)***

***printf("x[%d]=%5.10lf\n",i+1,x[i]);***

***printf("Значение функции цели = %5.10lf",f(x));***

***for(i=0;i<n;i++)***

***free(d2f[i]);***

***free(d2f);***

***free(x);***

***free(df);***

***fflush(stdin);***

***getchar();***

***return 0;***

***}***

***void invert(int n, int &q)***

***{***

***double \*\*a,t;***

***int i,j,k,m;***

***m = 2\*n;***

***a = (double\*\*)malloc(sizeof(double\*)\*n);***

***for(i=0;i<n;i++)***

***a[i] = (double\*)malloc(sizeof(double)\*m);***

***q = 0;***

***for(i=0;i<n;i++)***

***for(j=0;j<m;j++)***

***if (j<n) a[i][j] = d2f[i][j];***

***else***

***if (j ==n+i) a[i][j]=1;***

***else a[i][j]=0;***

***for(i=0;i<n;i++){***

***k=i;***

***do{***

***if(!a[k][i])***

***{***

***q=1;***

***if(k < n-1) k++;***

***else return;***

***}***

***} while(q);***

***if(q)***

***for(j=0;j<m;j++)***

***{***

***t=a[k][j];***

***a[k][j]=a[i][j];***

***a[i][j]=t;***

***}***

***for(j=m-1;j>=i;j--) a[i][j] /= a[i][i];***

***for(k=0;k<n;k++)***

***if(k!=i)***

***for(j=m-1;j>=0;j--)***

***a[k][j] -= a[i][j]\*a[k][j];***

***}***

***q=0;***

***for(i=0;i<n;i++)***

***for(j=0;j<n;j++)***

***d2f[i][j]=a[i][j+n];***

***for(i=0;i<n;i++)***

***free(a[i]);***

***free(a);***

***}***

***void for\_mat\_d2f(int n,double &grad, double &h, double\*x)***

***{***

***double s,s1,f0;***

***int i,j;***

***//расчет первых производных***

***f0=f(x); //s=0;***

***for(i=0;i<n;i++)***

***{***

***x[i] += dx;***

***df[i]=(f(x)-f0)/dx;***

***s = sqr(df[i]);***

***x[i] -=dx;***

***}***

***grad = sqrt(s);***

***//расчет вторых производных***

***for(i=0; i<n;i++)***

***{***

***s=-2\*f(x);***

***x[i] += dx;***

***s += f(x);***

***x[i] -=2\*dx;***

***s += f(x);***

***x[i] += dx;***

***d2f[i][i] = s/sqr(dx);***

***}***

***//расчет смешанных производных***

***for(i=0;i<n-1;i++)***

***for(j=i+1;j < n; j++)***

***{***

***s=f(x);***

***x[i] -= dx; x[j] -=dx;***

***s += f(x);***

***x[j] += dx;***

***s -= f(x);***

***x[i] += dx; x[j]-= dx;***

***s-= f(x);***

***x[j] += dx;***

***d2f[i][j] = s/sqr(dx);***

***d2f[j][i] = d2f[i][j];***

***}***

***s=0; s1= 0;***

***for(i=0;i<n;i++)***

***s+=sqr(df[i]);***

***for(i=0;i<n;i++)***

***for(j=0;j<n;j++)***

***s1+= d2f[i][j]\*df[i]\*df[j];***

***h=s/s1; grad = sqrt(s);***

***}***