```
#include<stdlib.h>
#include<stdio.h>
#include<time.h>
#include<math.h>
#include<conio.h>

#define gen_max 2000
#define D 10
#define NP 100
#define F 0.3
#define CR 0.8
#define N 2
#define M 3
#define H 6000.0
#define alpha 250.0
#define beta 0.6
/*#define inibound_l 0.0
#define inibound_h 6000.0  */

/*----Constant for rnd_uni()-------------------*/
#define IM1 2147483563
#define IM2 2147483399
#define AM (1.0/IM1)
#define IMM1 (IM1-1)
#define IA1 40014
#define IA2 40692
#define IQ1 53668
#define IQ2 52774
#define IR1 12211
#define IR2 3791
#define NTAB 32
#define NDIV (1+IMM1/NTAB)
#define EPS1 1.2e-7
#define RNMX (1.0-EPS1)

double evaluate(double [],long *);
double evaluate(double tmp[],long *nfe)
 {     double cost; int j;  (*nfe)++;

   cost=alpha*((int)tmp[0])*pow((tmp[3]*2500.0), beta)+((int)tmp[1]*pow((tmp[4]*2500), beta)+((in
t)tmp[2])*pow((tmp[5]*2500), beta));
         return(cost);

 } /************** end of evaluate() ******************/

float rnd_uni(long *);
float rnd_uni(long *idum)
       {
      long j; long k;
      static long idum2=123456789;
      static long iy=0;static long iv[NTAB]; float temp;
      if(*idum<=0)
      {
         if(-(*idum)<1)    *idum=1;  else  *idum=-(*idum);   idum2=(*idum);
         for(j=NTAB+7;j>=0;j--)
         {
           k=(*idum)/IQ1;
           *idum=IA1*(*idum-k*IQ1)-k*IR1;
           if(*idum<0)      *idum+=IM1;
           if(j<NTAB)       iv[j]=*idum;
         }
         iy=iv[0];
       }
       k=(*idum)/IQ1;
       *idum=IA1*(*idum-k*IQ1)-k*IR1;
       if(*idum<0)    *idum+=IM1;
       k=idum2/IQ2;
       idum2=IA2*(idum2-k*IQ2)-k*IR2;
       if(idum2<0)         idum2+=IM2;
       j=iy/NDIV;      iy=iv[j]-idum2;   iv[j]=*idum;
       if(iy<1)    iy+=IMM1;
       if((temp=AM*iy)>RNMX)     return RNMX;
       else                      return temp;
     }

void main()
```

```
{
   int i,j,k,a,b,c,good,count=0,seed; long nfe=0;
   double x1[NP][D],x2[NP][D],cost[NP],trial[D],cost_trial,pen,Sf[N][M],tp[N][M];
   double lhs2,lhs3,lhs4,costmin,lhs1,costmax,Q[2];Q[0]=40000.0;Q[1]=20000.0;
   Sf[0][0]=2.0,Sf[0][1]=3.0,Sf[0][2]=4.0,Sf[1][0]=4.0;Sf[1][1]=6.0,Sf[1][2]=3.0;
   tp[0][0]=8.0,tp[0][1]=20.0,tp[0][2]=8.0,tp[1][0]=16.0,tp[1][1]=4.0,tp[1][2]=4.0;
   double lhs5,lhs6,lhs7,lhs8,lhs9,lhs10,lhs11,lhs12,lhs13;
   clock_t start, end;
 printf("\nseed=");
 scanf("%d",&seed);
 long  rnd_uni_init= -(long)seed;   start = clock();

for (i=0;i<NP;i++)
 {
      for (j=0;j<D;j++)
       {
      if(j==0 ||j==1 ||j==2)
      x1[i][j]= (int)(1.0+rnd_uni(&rnd_uni_init)*(3.0-1.0));
      if(j==3 || j==4 || j==5)
      x1[i][j]=0.1 + rnd_uni(&rnd_uni_init)*(1.0-0.1);
      if(j==6 || j==7)
      x1[i][j]=0.33 + rnd_uni(&rnd_uni_init)*(1.0-0.33);
      if(j==8)
      x1[i][j]=(20.0*x1[i][6]/(625.0*3)) + rnd_uni(&rnd_uni_init)*(1.0-(20.0*x1[i][6]/(625.0*3)))
;
      if(j==9)
      x1[i][j]=1.0*x1[i][7]/125.0 + rnd_uni(&rnd_uni_init)*(1.0-(1.0*x1[i][7]/125.0));
       }

    pen=0.0;

    lhs4=((x1[i][5]*2500)/(x1[i][8]*625));
      if(lhs4<Sf[0][2])
       {
      pen=lhs4*1e20;
      cost[i]=evaluate(x1[i], &nfe);
      cost[i]=cost[i]+pen;
      continue;
       }
    lhs6=((x1[i][4]*2500)/(x1[i][9]*1250/3));
      if(lhs6<Sf[1][1])
       {
      pen=lhs6*1e20;
      cost[i]=evaluate(x1[i], &nfe);
      cost[i]=cost[i]+pen;
      continue;
       }
    lhs1=(Q[0]*(x1[i][6]*20)/(H*(x1[i][8]*625)))+(Q[1]*(x1[i][7]*16)/(H*(x1[i][9]*1250/3.0)))-1;
     if(lhs1>0.0)
      {
      pen=lhs1*1e20;
      cost[i]=evaluate(x1[i], &nfe);
      cost[i]=cost[i]+pen;
      continue;
      }

    lhs2=((x1[i][3]*2500)/(x1[i][8]*625));
      if(lhs2<Sf[0][0])
       {
      pen=lhs2*1e20;
      cost[i]=evaluate(x1[i], &nfe);
      cost[i]=cost[i]+pen;
      continue;
       }
    lhs3=((x1[i][4]*2500)/(x1[i][8]*625));
      if(lhs3<Sf[0][1])
       {
      pen=lhs3*1e20;
      cost[i]=evaluate(x1[i], &nfe);
      cost[i]=cost[i]+pen;
      continue;
       }

    lhs5=((x1[i][3]*2500)/(x1[i][9]*1250/3));
      if(lhs5<Sf[1][0])
```

```
       {
      pen=lhs5*1e20;
      cost[i]=evaluate(x1[i], &nfe);
      cost[i]=cost[i]+pen;
      continue;
       }

    lhs7=((x1[i][5]*2500)/(x1[i][9]*1250/3));
      if(lhs7<Sf[1][2])
       {
      pen=lhs7*1e20;
      cost[i]=evaluate(x1[i], &nfe);
      cost[i]=cost[i]+pen;
      continue;
       }
    lhs8=((int)x1[i][0]*(x1[i][6]*20));
      if(lhs8<tp[0][0])
       {
      pen=lhs8*1e20;
      cost[i]=evaluate(x1[i], &nfe);
      cost[i]=cost[i]+pen;
      continue;
       }
    lhs9=((int)x1[i][1]*x1[i][6]*20);
      if(lhs9<tp[0][1])
       {
      pen=lhs9*1e20;
      cost[i]=evaluate(x1[i], &nfe);
      cost[i]=cost[i]+pen;
      continue;
       }
    lhs10=((int)x1[i][2]*(x1[i][6]*20));
      if(lhs10<tp[0][2])
       {
      pen=lhs10*1e20;
      cost[i]=evaluate(x1[i], &nfe);
      cost[i]=cost[i]+pen;
      continue;
       }
    lhs11=((int)x1[i][0]*(x1[i][7]*16));
      if(lhs11<tp[1][0])
       {
      pen=lhs11*1e20;
      cost[i]=evaluate(x1[i], &nfe);
      cost[i]=cost[i]+pen;
      continue;
       }
    lhs12=((int)x1[i][1]*(x1[i][7]*16));
      if(lhs12<tp[1][1])
       {
      pen=lhs12*1e20;
      cost[i]=evaluate(x1[i], &nfe);
      cost[i]=cost[i]+pen;
      continue;
       }
    lhs13=((int)x1[i][2]*(x1[i][7]*16));
      if(lhs13<tp[1][2])
       {
      pen=lhs13*1e20;
      cost[i]=evaluate(x1[i], &nfe);
      cost[i]=cost[i]+pen;
      continue;
       }

if(lhs1<=0.0 && lhs2>=2.0 && lhs3>=3.0 && lhs4>=4.0 && lhs5>=4.0 && lhs6>=6.0 && lhs7>=3.0 && lhs
8>=8.0 && lhs9>=20.0 && lhs10>=8.0 && lhs11>=16.0 && lhs12>=4.0 && lhs13>=4.0)
       cost[i]=evaluate(x1[i], &nfe);
}
    costmin=cost[0];
      for(i=1;i<NP;i++)
       {  if(cost[i]<costmin)
      costmin=cost[i];
       }

while (count<gen_max)
 {
       for (i=0;i<NP;i++)
    {
        do a=int ((rnd_uni(&rnd_uni_init))*NP); while (a==i);

        do b=int (rnd_uni(&rnd_uni_init)*NP); while (b==i || b==a);

        do c=int (rnd_uni(&rnd_uni_init)*NP); while (c==i || c==a || c==b);

        j=int (rnd_uni(&rnd_uni_init)*D);

        for (k=1;k<=D;k++)
          {
           if(rnd_uni(&rnd_uni_init)<CR  || k==D)
              {
             trial[j]=x1[c][j]+F*(x1[a][j]-x1[b][j]);
              }
           else trial[j]=x1[i][j];

     if(j==0 ||j==1 || j==2)
     {      if(trial[j]<1.0)    trial[j]=1.0;
        if(trial[j]>3.0)    trial[j]=3.0;
        /*  trial[j]=(int)(1.0+rnd_uni(&rnd_uni_init)*(3.0-1.0)); */
     }
     if(j==3 || j==4 || j==5)
     {  if(trial[j]<0.1)    trial[j]=0.1;
        if(trial[j]>1.0)        trial[j]=1.0;
          /*trial[j]=(0.1+rnd_uni(&rnd_uni_init)*(1.0-0.1));  */
     }
     if(j==6 || j==7)
     {  if(trial[j]<0.33)   trial[j]=0.33;
        if(trial[j]>1.0)        trial[j]=1.0;
         /* trial[j]=(0.33+rnd_uni(&rnd_uni_init)*(1.0-0.33));*/
     }
     if(j==8)
     {      if(trial[j]<(20.0*trial[j-2]/(625.0*3))) trial[j]=(20.0*trial[j-2]/(625.0*3));
        if(trial[j]>1.0)      trial[j]=1.0;
          /*trial[j]=((20.0*trial[j-2]/(625.0*3))+rnd_uni(&rnd_uni_init)*(1.0-(20.0*trial[6]/(625
.0*3)))); */
     }
     if(j==9)
     {      if(trial[j]<(1.0*trial[j-2]/125.0))  trial[j]=(1.0*trial[j-2]/125.0);
        if(trial[j]>1.0)            trial[j]=1.0;
        /*  trial[j]=((1.0*trial[j-2]/125.0)+rnd_uni(&rnd_uni_init)*(1.0-(1.0*trial[6]/125.0)));
*/
     }
         j=(j+1)%D;
          }

  pen=0.0;

    lhs4=(trial[5]*2500)/(trial[8]*625);
    if(lhs4<Sf[0][2])
      {
      pen=lhs4*1e25;
      cost_trial=evaluate(trial, &nfe);
      cost_trial=cost_trial+pen;
      if(cost_trial<cost[i])
        {
            for (j=0;j<D;j++)
            x2[i][j]=trial[j];
            cost[i]=cost_trial;
          if(cost_trial<costmin)
           {
            costmin=cost_trial;
              /* imin=i;
            assignd(best,trial); */
           }
        }
        else for (j=0;j<D;j++)
            x2[i][j]=x1[i][j];
      continue;
      }
   lhs6=(trial[4]*2500)/(trial[9]*1250/3);
    if(lhs6<Sf[1][1])
      {
      pen=lhs6*1e25;
      cost_trial=evaluate(trial, &nfe);
      cost_trial=cost_trial+pen;
      if(cost_trial<cost[i])
        {
            for (j=0;j<D;j++)
            x2[i][j]=trial[j];
            cost[i]=cost_trial;
          if(cost_trial<costmin)
           {
            costmin=cost_trial;
              /* imin=i;
            assignd(best,trial); */
           }
        }
        else for (j=0;j<D;j++)
            x2[i][j]=x1[i][j];
      continue;
      }
   lhs1=(Q[0]*(trial[6]*20)/(H*(trial[8]*625)))+(Q[1]*(trial[7]*16)/(H*(trial[9]*1250/3)))-1;
    if(lhs1>0.0)
      {
      pen=lhs1*1e25;
      cost_trial=evaluate(trial, &nfe);
      cost_trial=cost_trial+pen;
      if(cost_trial<cost[i])
        {
            for (j=0;j<D;j++)
            x2[i][j]=trial[j];
            cost[i]=cost_trial;
          if(cost_trial<costmin)
           {
            costmin=cost_trial;
              /* imin=i;
            assignd(best,trial); */
           }
        }
        else for (j=0;j<D;j++)
            x2[i][j]=x1[i][j];
      continue;
      }
    lhs2=(trial[3]*2500)/(trial[8]*625);
    if(lhs2<Sf[0][0])
      {
      pen=lhs2*1e25;
      cost_trial=evaluate(trial, &nfe);
      cost_trial=cost_trial+pen;
      if(cost_trial<cost[i])
        {
            for (j=0;j<D;j++)
            x2[i][j]=trial[j];
            cost[i]=cost_trial;
          if(cost_trial<costmin)
           {
            costmin=cost_trial;
              /* imin=i;
            assignd(best,trial); */
           }
        }
        else for (j=0;j<D;j++)
            x2[i][j]=x1[i][j];
      continue;
      }
    lhs3=(trial[4]*2500)/(trial[8]*625);
    if(lhs3<Sf[0][1])
      {
      pen=lhs3*1e25;
      cost_trial=evaluate(trial, &nfe);
      cost_trial=cost_trial+pen;
      if(cost_trial<cost[i])
        {
            for (j=0;j<D;j++)
            x2[i][j]=trial[j];
            cost[i]=cost_trial;
          if(cost_trial<costmin)
           {
            costmin=cost_trial;
```

```
          /* imin=i;
        assignd(best,trial); */
       }
    }
    else for (j=0;j<D;j++)
        x2[i][j]=x1[i][j];
  continue;
  }

lhs5=(trial[3]*2500)/(trial[9]*1250/3);
if(lhs5<Sf[1][0])
  {
  pen=lhs5*1e25;
  cost_trial=evaluate(trial, &nfe);
  cost_trial=cost_trial+pen;
  if(cost_trial<cost[i])
    {
        for (j=0;j<D;j++)
        x2[i][j]=trial[j];
        cost[i]=cost_trial;
      if(cost_trial<costmin)
       {
        costmin=cost_trial;
          /* imin=i;
        assignd(best,trial); */
       }
    }
    else for (j=0;j<D;j++)
        x2[i][j]=x1[i][j];
  continue;
  }

lhs7=(trial[5]*2500)/(trial[9]*1250/3);
if(lhs7<Sf[1][2])
  {
  pen=lhs7*1e25;
  cost_trial=evaluate(trial, &nfe);
  cost_trial=cost_trial+pen;
  if(cost_trial<cost[i])
    {
        for (j=0;j<D;j++)
        x2[i][j]=trial[j];
        cost[i]=cost_trial;
      if(cost_trial<costmin)
       {
        costmin=cost_trial;
          /* imin=i;
        assignd(best,trial); */
       }
    }
    else for (j=0;j<D;j++)
        x2[i][j]=x1[i][j];
  continue;
  }

lhs8=((int)trial[0]*(trial[6]*20));
if(lhs8<tp[0][0])
  {
  pen=lhs8*1e25;
  cost_trial=evaluate(trial, &nfe);
  cost_trial=cost_trial+pen;
  if(cost_trial<cost[i])
    {
        for (j=0;j<D;j++)
        x2[i][j]=trial[j];
        cost[i]=cost_trial;
      if(cost_trial<costmin)
       {
        costmin=cost_trial;
          /* imin=i;
        assignd(best,trial); */
       }
    }
    else for (j=0;j<D;j++)
        x2[i][j]=x1[i][j];
  continue;
  }

lhs9=((int)trial[1]*(trial[6]*20));
if(lhs9<tp[0][1])
  {
  pen=lhs9*1e25;
  cost_trial=evaluate(trial, &nfe);
  cost_trial=cost_trial+pen;
  if(cost_trial<cost[i])
    {
        for (j=0;j<D;j++)
        x2[i][j]=trial[j];
        cost[i]=cost_trial;
      if(cost_trial<costmin)
       {
        costmin=cost_trial;
          /* imin=i;
        assignd(best,trial); */
       }
    }
    else for (j=0;j<D;j++)
        x2[i][j]=x1[i][j];
  continue;
  }
lhs10=((int)trial[2]*(trial[6]*20));
if(lhs10<tp[0][2])
   {
  pen=(lhs10)*1e25;
  cost_trial=evaluate(trial, &nfe);
  cost_trial=cost_trial+pen;
  if(cost_trial<cost[i])
    {
        for (j=0;j<D;j++)
        x2[i][j]=trial[j];
        cost[i]=cost_trial;
      if(cost_trial<costmin)
       {
        costmin=cost_trial;
          /* imin=i;
        assignd(best,trial); */
       }
    }
    else for (j=0;j<D;j++)
        x2[i][j]=x1[i][j];
  continue;
  }
lhs11=((int)trial[0]*(trial[7]*16));
if(lhs11<tp[1][0])
  {
  pen=lhs11*1e25;
  cost_trial=evaluate(trial, &nfe);
  cost_trial=cost_trial+pen;
  if(cost_trial<cost[i])
    {
        for (j=0;j<D;j++)
        x2[i][j]=trial[j];
        cost[i]=cost_trial;
      if(cost_trial<costmin)
       {
        costmin=cost_trial;
          /* imin=i;
        assignd(best,trial); */
       }
    }
    else for (j=0;j<D;j++)
        x2[i][j]=x1[i][j];
  continue;
  }
lhs12=((int)trial[1]*(trial[7]*16));
if(lhs12<tp[1][1])
   {
  pen=lhs12*1e25;
  cost_trial=evaluate(trial, &nfe);
  cost_trial=cost_trial+pen;
  if(cost_trial<cost[i])
    {
            for (j=0;j<D;j++)
            x2[i][j]=trial[j];
            cost[i]=cost_trial;
          if(cost_trial<costmin)
           {
            costmin=cost_trial;
              /* imin=i;
            assignd(best,trial); */
           }
        }
        else for (j=0;j<D;j++)
            x2[i][j]=x1[i][j];
      continue;
      }
    lhs13=((int)trial[2]*(trial[7]*16));
    if(lhs13<tp[1][2])

      {
      pen=lhs13*1e25;
      cost_trial=evaluate(trial, &nfe);
      cost_trial=cost_trial+pen;
      if(cost_trial<cost[i])
        {
            for (j=0;j<D;j++)
            x2[i][j]=trial[j];
            cost[i]=cost_trial;
          if(cost_trial<costmin)
           {
            costmin=cost_trial;
              /* imin=i;
            assignd(best,trial); */
           }
        }
        else for (j=0;j<D;j++)
            x2[i][j]=x1[i][j];
      continue;
      }

  if(lhs1<=0.0 && lhs2>=2.0 && lhs3>=3.0 && lhs4>=4.0 && lhs5>=4.0 && lhs6>=6.0 && lhs7>=3.0 && l
hs8>=8.0 && lhs9>=20.0 && lhs10>=8.0 && lhs11>=16.0 && lhs12>=4.0 && lhs13>=4.0)
     {   cost_trial=evaluate(trial, &nfe);
      if(cost_trial<=cost[i])
        {
            for (j=0;j<D;j++)
            x2[i][j]=trial[j];
            cost[i]=cost_trial;
          if(cost_trial<costmin)
           {
            costmin=cost_trial;
              /* imin=i;
            assignd(best,trial); */
           }
        }
        else for (j=0;j<D;j++)
            x2[i][j]=x1[i][j];
     continue;
      }
   } /************* end of for loop ****************/

      for (i=0;i<NP;i++)
    {
        for (j=0;j<D;j++)
        x1[i][j]=x2[i][j];
    }

    costmax=cost[0];
      for(i=1;i<NP;i++)
       {  if(costmax<cost[i])
      costmax=cost[i];
       }
       costmin=cost[0];
     for(i=1;i<NP;i++)
       {  if(costmin>cost[i])
      costmin=cost[i];
       }
   if((costmax-costmin)<0.00001)
       break;
    count++;

}  /*************  end of while loop *****************/

        end = clock();
for(i=0;i<NP;i++)
 {
   for(j=0;j<D;j++)
    { if(j==3 || j==4 || j==5)
     printf("u[%d]=%lf  ",j,(x1[i][j]*2500));
     if(j==6)
     printf("u[%d]=%lf  ",j,(x1[i][j]*20));
     if(j==7)
     printf("u[%d]=%lf  ",j,(x1[i][j]*16));
     if(j==8)
     printf("u[%d]=%lf  ",j,(x1[i][j]*625));
     if(j==9)
     printf("u[%d]=%lf  ",j,(x1[i][j]*1250/3));
     if(j==0 || j==1 || j==2)
     printf("u[%d]=%lf    ",j,(x1[i][j]));
    }
     printf("cost[%d]=%lf     ",i,cost[i]);
 }
 printf("NFE=%ld\n",nfe);
 printf("The time was: %f\n", (end - start) / CLK_TCK);
 printf("costmax is=%lf",costmax);
 printf("lhs1=%lf  lhs2=%lf  lhs3=%lf lhs4=%lf  lhs5=%lf lhs6=%lf",lhs1,lhs2,lhs3,lhs4,lhs5,lhs6)
;
 printf("lhs7=%lf  lhs8=%lf  lhs9=%lf lhs10=%lf  lhs11=%lf lhs12=%lf",lhs7,lhs8,lhs9,lhs10,lhs11,
lhs12);
 printf("lhs13=%lf",lhs13);printf("\ncostmin=%lf     seed=%d\n",costmin,seed);

} /************* end of main() ******************/
```