

```

/*******
/*          スプライン関数の決定（1次関数法） spline.c          */
/*          スプライン関数を求める。さらに等分刻みのxの値に      */
/*          対する関数値を求めて結果を数表の形で出力する。        */
/*******
#include <stdio.h>
#include <math.h>
#define N 10
int main(void)
{
    int i,j,k,m,n,kz,z,lp;
    double A[N],B[N],C[N+2],D[N],h[N],x[N];
    double y[N], u[N], p[N+2][N+3];
    double g,q,xw,s,f;
    char qq,zz;
    /* h[]:hj  u[]:uj  A[],B[],C[],D[]:Aj,Bj,Cj,Dj */
    /* p[][]: 連立方程式の係数配列 */
    /* x,y, に関するデータの入力 */
    while(1)
    {
        printf("スプライン関数の決定（1次関数法） \n\n");
        printf("データの個数は何個ですか？（2<m<10） m = ");
        scanf("%d%c",&m,&zz);
        if( (m <= 2) || (10<=m) ) continue;
        n = m-1;
        for(i=0; i<=n; i++)
        {
            printf("x(%d)= ",i); scanf("%lf%c",&x[i],&zz);
            printf("y(%d)= ",i); scanf("%lf%c",&y[i],&zz);
        }

        printf("全区間の左端点における1次微分係数は？ ");
        scanf("%lf%c",&C[1],&zz);

        printf("全区間の右端点における1次微分係数は？ ");
        scanf("%lf%c",&C[n+1],&zz);

        printf("\n 正しく入力しましたか？（y/n）");
        scanf("%c%c",&qq,&zz);
        if( qq=='y' ) break;
    }
    for(i=1; i<=n; i++)
    {
        D[i] = y[i-1];
        h[i] = x[i] - x[i-1];
        u[i] = (y[i] - y[i-1]) / h[i];
    }
    for(i=1;i<=n+1; i++)
    {
        for( j=1; j<=n+2; j++)
        {
            p[i][j] = 0.0;
        }
    }
}

```

```

/* 漸化式から得られる連立方程式の係数を*/
/* 配列pに入れる */
p[1][1] = 1.0;
p[n+1][n+1] = 1.0;
p[1][n+2] = C[1];
p[n+1][n+2] = C[n+1];
for(j=2; j<=n; j++)
{
    p[j][j-1] = h[j];
    p[j][j] = 2*(h[j-1] + h[j]);
    p[j][j+1] = h[j-1];
    p[j][n+2] = 3*(h[j]*u[j-1] + h[j-1]*u[j]);
}

/* ガウス・ジョルダン(掃き出し) 法により*/
/* 連立方程式を解く */
for(i=1; i<=n+1; i++)
{
    q = p[i][i];
    for(j=1; j<=n+2; j++)
    {
        p[i][j] = p[i][j] / q;
    }
    for(k=1; k<=n+1; k++)
    {
        g = p[k][i];
        if(k != i){
            for(j=1; j<=n+2; j++)
            {
                p[k][j] = p[k][j] - g * p[i][j];
            }
        }
    }
}

/*上で得られた解を配列cに入れる*/
for(j=1; j<=n; j++)
{
    C[j] = p[j][n+2];
}

/*B1,B2,...,Bnを求める*/
for(j=1; j<=n-1; j++)
{
    B[j] = (3*u[j]-2*C[j]-C[j+1]) / h[j];
}
B[n] = -(3*u[n-1]-C[n-1]-2*C[n]) / h[n-1];

/*A1,A2,...,Anを求める*/
for(j=1; j<=n-1; j++)
{
    A[j] = (B[j+1] - B[j]) / (3* h[j]);
}
A[n] = (u[n] - h[n] * B[n] -C[n]) / (h[n] * h[n]);

```

```

/*結果を出力する*/
printf("名区間のスプライン関数の係数を出力します\n");
printf("(X-Xj)の降べきの順(係数Aj,Bj,Cj,Djの値) \n");
for(i=1; i<=n; i++)
{
    printf("S%d(x)=",i);
    printf("(x-%6.31f)^3+",A[i],x[i-1]);
    printf("(x-%6.31f)^2+",B[i],x[i-1]);
    printf("(x-%6.31f)+",C[i],x[i-1]);
    printf("(x-%6.31f)\n",D[i]);
}
/*求めたスプライン関数を使って補間値を求める*/
printf("x座標の範囲を何等分して補間値を求めますか? \n");
scanf("%d",&kz);
xw = (x[n] - x[0]) / kz;
for (z=1,s=x[0],lp=0;lp<=kz;lp++)
{
    if(z >= n) z--;
    f = A[z] * pow((s-x[z-1]),3.0) + B[z] *
        pow((s-x[z-1]),2.0)+C[z]*(s-x[z-1])+D[z];
    printf("%10.61f %10.61f\n",s,f);
    s = s + xw;
    if(s > x[z]) z++;
}
return 0;
}

```