

```

/* 基本関数fiのxでの関数値yを求める */
double ffv(int fi, double x)
{
    double y;
    switch(fi) {
        case 1: y = 1.0;      break;
        case 2: y = x;        break;
        case 3: y = pow(x,2);  break;
        case 4: y = pow(x,3);  break;
        case 5: y = 1.0/x;     break;
        case 6: y = exp(x);    break;

        default: y = x;       break;
    }
    return y;
}

/** データの入力 */
int data_input(char *fname, double x[N], double y[N] )
{
    int n=-1, i;
    FILE* fp;

    fp = fopen(fname, "r");
    if (fp == NULL){
        printf("ファイル「%s」を開くことができません", fname);
        exit(0);
    }

    /* データの個数は何個ですか? (1< n < %d) n = ", N); */
    fscanf(fp, "%d", &n);
    if ( (n <= 1) || (N <= n) )
        printf("データの個数は(1< n < %d) でなければなりません, n = %d", N, n);

    /* printf("\nデータxの値は小から大の順に入力する. \n"); */
    for(i=1; i<=n; i++) {
        fscanf(fp, "%lf", &x[i]);
        fscanf(fp, "%lf", &y[i]);
    }

    fclose(fp);
    return n;
}

/* モデルを読み取る */
int read_model(FILE *fp_models, int f_id[N])
{
    int k = 0;
    char line[81], *token;

    /* 基本関数fk(x)を1～kの番号を読み取る */
    /* ファイルの 1 行から文字列としてモデルの仕様を読み取る */
    fgets(line, sizeof (line), fp_models);

```

```

/* 空白で区切られた文字列に分割 */
token = strtok(line, " ");
while ( token != NULL)
{
    k++;
    /* 関数のIDを文字列から整数に変換する */
    f_id[k] = atoi(token);
    /* 次の関数のIDを取得します */
    token = strtok(NULL, " ");
}
/* 基本関数の数を返す */
return k;
}

```

```

/* A[n][k], tA[k][n]とb[k]を設定する */
void set_A_tA_b(int n, int k, int f_id[N], double x[N], double y[N], double
A[N][N], double tA[N][N], double b[N])
{
    int i, j;

```

```

}

```

```

/* tA[k][n]・A[n][k]を計算して配列tAA[k][k]に入れる */
void seki_tA_A(int n, int k, double tA[N][N], double A[N][N], double
tAA[N][N])
{
    int i, j, jj;
    double s;

```

```

}

```

```

/* tA[k][n]・b[k]を計算してtAb[k]配列に入れる */
void seki_tA_b(int n, int k, double tA[N][N], double b[N], double tAb[N])
{
    double s;
    int i, j;

```

```

}

```

```

/* モデルの係数を求める */
void compute_model(int n, int k, double x[N], double y[N], int f_id[N],
double sol[N])
{
    double A[N][N], tA[N][N], b[N], tAA[N][N], tAb[N];
    int i;

    for (i =0; i<=k ; i++)
        sol[i] = 0; /* 求めたa1,...akの係数を扱う*/

    /* A[n][k],tA[k][n]とb[k]を設定する */
    set_A_tA_b(n, k, f_id, x, y, A, tA, b);
    /* tA[k][n]・A[n][k]を計算して配列tAA[k][k]に入れる */
    seki_tA_A(n, k, tA, A, tAA);
    /* tA[k][n]・b[k]を計算してtAb[k]配列に入れる */
    seki_tA_b(n, k, tA, b, tAb);
    /* LU分解法で解く */
    lu_solve(k, tAA, tAb, sol);
}

```

```

/* モデルの誤差を求める */
double model_error(int n, int k , double x[N], double y[N], double sol[N],
int f_id[N])
{
    int i, j;
    double err =0.0, yy;

    for (i=1; i <= n; i++){
        yy = 0.0;
        for (j=1; j<= k; j++){
            yy += sol[j] * ffv(f_id[j], x[i]);
        }
        err += (yy-y[i])*(yy-y[i]);
    }
    return err/n;
}

```

```

/* モデルの誤差, モデルの基本関数と求めた係数を出力 */
void model_error_output(FILE* fp_out, int i, double err, int k, double
sol[N], int f_id[N])
{
    int j;
    fprintf(fp_out, "%d\t%f", i, err);
    for (j=1; j<=k; j++) {
        fprintf(fp_out, "\t%f*f%d", sol[j], f_id[j]);
    }
    fprintf(fp_out, "\n");
}

```

```

/* 補間結果ファイル : グラフを描くための準備 (数表を出力) */
void data_output(char *fname, int n, int k, double x[N], double y[N],
double sol[N], int f_id[N])
{
    double h, xx, yy;
    int i, j;
    FILE* fp;

    fp = fopen(fname, "w");
    if (fp == NULL){
        printf("ファイル「%s」を開くことができません", fname);
        exit(0);
    }

    h = (x[n] - x[1]) / 500.0;
    xx = x[1] ;
    for(i=0; i<=500; i++) {
        yy = 0.0;
        for (j=1; j<= k; j++)
            yy += sol[j] * ffv(f_id[j], xx) ;
        fprintf(fp, "%lf\t%lf\n", xx, yy);
        xx = xx + h;
    }
    fclose(fp);
}

```