

ภาคผนวก จ

โปรแกรมภาษาซี ที่ใช้สำหรับคำนวณค่าสถิติทดสอบทั้ง 5 สถิติ และคำนวณ
ค่าวิกฤติของสถิติทดสอบคราเมอร์-ฟอนมิส แบบถ่วงน้ำหนัก

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

#define TRUE 1
#define FALSE 0
#define MAX 60
#define MAXDATA 500

float power(float value,int pow);
double factorial(double n);
float w(unsigned n,unsigned i,unsigned k);
float t(unsigned n,unsigned k);
void bubbleSort(float numbers[],int s_order);
void find_z_a(void);
void random_e(void);
void create_data(void);
void process_w(int loop);
float t_w(unsigned n,unsigned k,float value_num3,float w_e[]);
void show_w();
void write_output(char out[],float ans_w1,float ans_w2,float ans_w3);
void keep_date();
```

```

float
table[310],inputs[MAX],e_sort[MAX],e_1[MAX],e_random[MAX],x_data[MAX],f_w1[100],f_
w2[100],f_w3[100],average,num1,num2,num3,all,ans_z_a,ans_z_c;
static int first_time = TRUE;
int main (int argc,char **argv)
{
    char filename[ ] = "./input/data1_1.dat";
    char tablename[ ] = "./table.txt";
    char output[] = "./output/output1_1.dat";
    float g,g1,g2,g_arg,d,data,sum,total_sum=0,r1,r2,test,test2;
    float w1=0,w2=0,w3=0;
    float ans[MAX],w_array[MAX],value_t,value_t2;
    float r1_up,r_down,r2_up;
    int loop,i,k,n,order = 0,tcount = 0,sort_order = 0,a,b;
    int debug = 0;
    char ch;
    FILE *fp,*ft;
    time_t t1;
    struct tm *tm_ptr;
    if ( argc != 4 )
    {
        printf("Usage : ./one <value>\nvalue in [1, 2, ..., 8]\n");
        exit();
    }

/* --- Start Read Table --- */
    tcount = -1;
    ft = fopen(tablename,"r");
    while(fscanf(ft,"%f",&data) != EOF)

```

```

    {
        table[++tcount] = data;
    }
    fclose(ft);
/* --- End Read Table --- */

a = 1;
/* for(b=1; b<=500; b++)*/
for(b=atoi(argv[2]); b<=atoi(argv[3]); b++)
{
    keep_date();
    order = 0;
    all = 0;
    w1 = 0;
    w2 = 0;
    w3 = 0;
    sprintf(filename, "./input/d%s_%d.dat", argv[1], b);
    sprintf(output, "./output/o%s_%d.dat", argv[1], b);
    printf("filename : %s\n", filename);
    printf("output : %s\n\n", output);

/* --- Open file for read data --- */
    if( (fp = fopen(filename, "r")) == NULL)
    {
        fprintf(stderr, "Can't Open %s\n", filename);
        exit(1);
    }
    while(fscanf(fp, "%f", &data) != EOF)
    {
        inputs[order++] = data;
    }
}

```

```

    }

    fclose(fp);

    all = order;

/* --- End Read Input File --- */

/* Search Average */
    sum = 0;

    for(i=0; i< order; i++)
    {
        sum += inputs[i];
    }

    average = sum / order;

    printf (" Average : %6.3f / %d = %6.3f\n\n",sum,order,average);

/* Find r1 */
    r1 = 0;
    r1_up = 0;
    for(i=1; i<order; i++)
    {
        r1_up = r1_up + ( ( inputs[i] - average )*( inputs[i-1] - average ) );
    }

    r_down = 0;
    for(i=0; i < order; i++)
    {
        r_down = r_down + ( ( inputs[i] - average )*( inputs[i] - average ) );
    }

    r1 = r1_up / r_down;

    printf (" R1 : %6.3f / %6.3f = %6.3f\n\n",r1_up,r_down,r1);

```

```

/* Find r2 */
r2 = 0;
r2_up = 0;
for(i=2; i<order; i++)
{
    r2_up = r2_up + ( ( inputs[i] - average )*( inputs[i-2] - average ) );
}
r2 = r2_up / r_down;
printf (" R2 : %6.3f / %6.3f = %6.3f\n",r2_up,r_down,r2);

/* Find num1 */
num1 = ( r1 * ( 1 - r2 ) ) / ( 1 - (r1 * r1) );
printf (" Num1 : %6.3f\n",num1);

/* Find num2 */
num2 = ( ( r2 - (r1 * r1) ) / ( 1 - (r1 * r1) ) );
printf (" Num2 : %6.3f\n",num2);

/* Find Num3 */
num3 = 0;
sort_order = 0;
for(i=2; i< order; i++)
{
    sum = 0;
    sum = ( ( inputs[i]-average ) - (num1 * (inputs[i-1] - average)) - (num2 * (inputs[i-2] -
average)) );
    e_sort[sort_order++] = sum;
    e_1[sort_order] = sum;
}

```

```

    num3 += power( sum, 2);
}
sum = num3 / (order-2);
num3 = sqrt(sum);
printf (" Num3 : %6.3f\n\n",num3);

bubbleSort(e_sort,sort_order);

/* Start Find W */

n = order;
for(k=1;k<=n-1; k++)
{
    g = ( 1 / sqrt( ((float)k/(float)n) * (1-((float)k/(float)n)) ) );
    d = (float)k/(float)n;
    g1 = power( d, 2 );
    g2 = ( power( d, 2) / (float)(d+1) );
    sum = 0;
    value_t = t(n,k);

/*
    printf(" G1 : %6.3f\n",g);
    printf(" G2 : %6.3f\n",g1);
    printf(" G3 : %6.3f\n",g2);
    printf(" T : %6.3f\n",value_t);
    printf(" D : %6.3f\n\n",d);
*/
    value_t2 = power( (value_t - d), 2);
    sum = power( g, 2) * value_t2;

```

```
w1 += sum;

sum = power( g1, 2) * value_t2;
w2 += sum;

sum = power( g2, 2) * value_t2;
w3 += sum;

if ( debug == 1)
{
    if (k != n-1)
    {
        printf ("(%6.3f * power( %6.3f-%6.3f, 2) + ", power( g, 2),t(n,k),d);
    }
    else
    {
        printf ("(%6.3f * power( %6.3f-%6.3f,2 )\n", power( g, 2),t(n,k),d );
    }
}

/* total_sum += sum;*/

/* printf(" power(%6.3f) * power(%6.3f - %6.3f) = ",g,value_t,d);*/
if ( debug == 2)
{
    printf(" Sum : %6.3fn",sum);
}
}
```

```
printf ("\nW1 : %6.3f\n",w1);
printf ("W2 : %6.3f\n",w2);
printf ("W3 : %6.3f\n\n",w3);

/* Find Z */
find_z_a();

/* Start Random */
e_random[0] = 0;
e_random[1] = 0;
printf(" Start Find W100 \n");
for(loop=0;loop<100;loop++)
{
    if ( debug == 1 )
    {
        printf(" +++ Loop : %d +++\n", (loop+1));
    }

    random_e();
    create_data();
    process_w(loop);
}

bubbleSort(f_w1,100);
bubbleSort(f_w2,100);
bubbleSort(f_w3,100);
show_w();
write_output(output,w1,w2,w3);
```

```
    }  
}  
  
double factorial(double n)  
{  
    double fac = 1;  
    if (n > 1)  
    {  
        fac = n * factorial(n-1);  
    }  
    else  
    {  
        fac = 1;  
    }  
    return fac;  
}  
  
float power(float value, int pow)  
{  
    float return_value = 1.0;  
    unsigned i;  
  
    for(i=1; i<= abs(pow); i++)  
    {  
        return_value *= value;  
    }  
    if ( pow >= 0 )  
    {  
        return(return_value);  
    }  
}
```

```

else
{
    return(1.0/return_value);
}
}

float t(unsigned n,unsigned k)
{
    float sum=0,w_ans=0,e=0,f=0,r_ans,f_value,f_100,test,t_ans=0;
    int i,r;
    int debug = 0;
    unsigned int abs_r;

    if ( debug == 2 )
    {
        printf("\n == Start T %d ==\n\n",k);
    }

    for(i=1; i<=n-2; i++)
    {
        /* printf("I : %d\n",i); */
        e = e_sort[i-1];
        f = e / num3;

        /* Round floating Point */

        f_100 = f * 100;
        r = f * 100;

        if ( ((f_100 - r) >= 0.5)||((f_100 - r) < -0.5) )

```

```

{
    if (r >= 0)
    {
        ++r;
    }
    else
    {
        --r;
    }
}

/* printf ("f,r = %f,%d\n",f,r);*/
/* printf (" E / v = %6.3f / %6.3f = %6.3f\n",e,num3,f);*/

abs_r = abs(r);
if ( ( f >= -3.09)&&(f <= 3.09) )
{
    if ( f >= 0 )
    {
        f_value = table[abs_r] ;
/*      printf("%f : table[%d] = %f\n",f,r,table[r]);*/
/*      f_value = table[abs_r] * w_ans;*/
    }
    else
    {
        r_ans = 1 - table[abs_r];
        f_value = ( 1 - table[abs_r] ) ;
/*      printf("%f : table[%d] = %f\n",f,r,r_ans);*/
/*      f_value = ( 1 - table[abs_r] ) * w_ans;*/
    }
}

```

```

}
else
{
    if ( f >= 0 )
    {
        f_value = 1;
    }
    else
    {
        f_value = 0; /* 1 - 1 */
    }
}
if ( debug == 2 )
{
    printf("Start w\n");
}
w_ans = w(n,i,k);

sum = ( f_value * w_ans );
t_ans += sum;
if ( debug == 2 )
{
    printf(" F = %6.3f , w = %6.3f , Sum : %6.3f\n",f_value,w_ans,sum);
    printf(" T_ans : %6.3f\n",t_ans);
}
}
/* printf("W(%d,%d)(X%d) = Fo(%6.3f/%6.3f) * %6.3f\n",n,k,n,e,num3,w(n,i,k));*/
return ( t_ans);

```

```
}

```

```
float w(unsigned n,unsigned i,unsigned k)

```

```
{

```

```
    unsigned l,debug=0;

```

```
    float  sum_g,test,g1=0,g2=0,c0,c1,c2,c3,c4;

```

```
    double  c00,c01,c02;

```

```
/*

```

```
    printf("n = %d , i = %d , k = %d\n",n,i,k);

```

```
*/

```

```
    for(l=0; l<=k-1; l++)

```

```
    {

```

```
        /* c0 = ( (float)factorial(n-2) / ( (float)factorial(l) * (float)factorial(n-2-l) ) );*/

```

```
/*    c00 = factorial((double)(l));

```

```
    printf("C00 : %ld\n",c00);

```

```
    c01 = factorial((double)(n-2-l));

```

```
    printf("C01 : %ld\n",c01); */

```

```
        c0 = (float)( factorial((double)(n-2)) / ( factorial( (double)(l) ) * factorial( (double)(n-2-
l) ) ) );

```

```
        c1 = ( ( (float)(i-1)/(float)(n-2)) );

```

```
        c2 = ( 1 - ( ( (float)(i-1)/(float)(n-2)) ) );

```

```
/*    printf("G1 += (%6.3f) * (%6.3f)%d * (%6.3f)%d\n",c0,c1,l,c2,n-2-l);*/

```

```
/*    printf("g1 += %6.3f * %6.3f * %6.3f\n",c0,power(c1,l),power(c2,n-2-l));

```

```
/*    test = ( c0 * power( c1, l ) * power( c2, n-2-l ) );*/

```

```
/*    printf("Test : %6.3f\n",test);*/

```

```
        g1 += ( c0 * power( c1, l ) * power( c2, n-2-l ) );

```

```
        if ( debug == 1)

```

```

    {
        if ( l != k-1 )
        {
/*      printf ("%6.3f * %6.3f * %6.3f) + ",c0,c1,c2);*/
            printf ("%6.3f * %6.3f * %6.3f) + ",c0,power(c1,l),power(c2,n-2-l));
        }
        else
        {
/*      printf ("%6.3f * %6.3f * %6.3f) \n",c0,c1,c2);*/
            printf ("%6.3f * %6.3f * %6.3f) \n",c0,power(c1,l),power(c2,n-2-l));
        }
    }
}
if ( debug == 1)
{
    printf("G1 : %6.3f\n",g1);
}
for(l=0; l<=k-1; l++)
{
/*    c0 = ( (float)factorial(n-2) / ( (float)factorial(l) * (float)factorial(n-2-l) ) );*/
    c0 = (float)( factorial((double)(n-2)) / ( factorial( (double)(l) ) * factorial( (double)(n-2-
l) ) ) );
    c3 = ( (float)i/(float)(n-2) );
    c4 = ( 1 - ( (float)i/(float)(n-2) ) );
/*    printf("G2 += (%6.3f) * (%6.3f)%d * (%6.3f)%d\n",c0,c3,l,c4,n-2-l);*/
    g2 += ( c0 * power( c3,l ) * power( c4, n-2-l ) );
    if ( debug == 1 )
    {
        if ( l != k-1 )

```

```

    {
/*      printf ("%6.3f * %6.3f * %6.3f) + ",c0,c3,c4);*/
        printf ("%6.3f * %6.3f * %6.3f) + ",c0,power(c3,l),power(c4,n-2-l));
    }
    else
    {
/*      printf ("%6.3f * %6.3f * %6.3f) \n",c0,c3,c4);*/
        printf ("%6.3f * %6.3f * %6.3f) \n",c0,power(c3,l),power(c4,n-2-l));
    }
    }
}

if ( debug == 1 )
{
    printf("G2 : %6.3f\n",g2);
}

sum_g = g1 - g2;
/* printf("Sum G1-G2 : %6.3f-%6.3f = %6.3f\n",g1,g2,sum_g);*/

return ( sum_g );

}

void bubbleSort(float numbers[], int s_order)
{
    int i, j;
    float temp;

    for (i = (s_order - 1); i >= 0; i--)
    {

```

```
for (j = 1; j <= i; j++)
{
    if (numbers[j-1] > numbers[j])
    {
        temp = numbers[j-1];
        numbers[j-1] = numbers[j];
        numbers[j] = temp;
    }
}
}

void find_z_a(void)
{
    int i,n,r_z;
    float f_z=0,one=0,two=0,sum_z=0,z=0,f_z_100,ans_f,ans_z=0;
    float z_c=0,sum_z_c=0,z_c_1,z_c_2,z_c_3,z_c_4;
    float test_a,test_b;
    int debug = 0;
    unsigned abs_r_z;

    n = all-2;
    for(i=1; i<=n; i++)
    {
        one = 0;
        two = 0;
        z = 0;
        r_z = 0;
        f_z = e_sort[i-1] ;
```

```
f_z_100 = f_z * 100;
r_z = f_z * 100;
if ( ((f_z_100 - r_z) >= 0.5) || ((f_z_100 - r_z) <= -0.5) )
{
    if (r_z >= 0)
    {
        ++r_z;
    }
    else
    {
        --r_z;
    }
}
abs_r_z = abs(r_z);
if ( debug != 0 )
{
    printf("f : %6.3f | R_z = %d\n",f_z,abs_r_z);
}
if ( ( f_z >= -3.09) && (f_z <= 3.09) )

{
    if ( f_z >= 0 )
    {
        ans_f = table[abs_r_z];
        if ( debug == 1 )
        {
            test_a = log(ans_f);
            test_b = (n - i + 0.5);
            printf("%6.3f / %6.3f\n",test_a,test_b);
        }
    }
}
```

```
}  
one = log(ans_f) / ( n - i + 0.5 );  
if ( debug == 1 )  
{  
    test_a = log(1-ans_f);  
    test_b = i - 0.5;  
    printf("%6.3f / %6.3f\n",test_a,test_b);  
}  
two = log(1-ans_f) / ( i - 0.5 );  
z_c_1 = pow( ans_f, -1) - 1;  
}  
else  
{  
    ans_f = 1 - table[abs_r_z];  
    if ( debug == 1 )  
    {  
        test_a = log(ans_f);  
        test_b = (n - i + 0.5);  
        printf("%6.3f / %6.3f\n",test_a,test_b);  
    }  
    one = log(ans_f) / ( n - i + 0.5 );  
    if ( debug == 1 )  
    {  
        test_a = log(1-ans_f);  
        test_b = i - 0.5;  
        printf("%6.3f / %6.3f\n",test_a,test_b);  
    }  
    two = log(1 - ans_f) / ( i - 0.5 );
```

```

if ( debug == 2 )
{
    test_a = pow( ans_f, -1) -1;
    printf("FoZc : %6.3f\n",test_a);
}
    z_c_1 = pow( ans_f, -1) - 1;
}
}
else
{
    if ( f_z >= 0 )
    {
        /* f > 3.09 */
        ans_f = 0.999;
        if ( debug == 1 )
        {
            test_a = log(ans_f);
            test_b = (n - i + 0.5);
            printf("%6.3f / %6.3f\n",test_a,test_b);
        }
        one = log(ans_f) / ( n - i + 0.5 );
        if ( debug == 1 )
        {
            test_a = log(1-ans_f);
            test_b = i - 0.5;
            printf("%6.3f / %6.3f\n",test_a,test_b);
        }
        two = log(1 - ans_f) / ( i - 0.5 );
        z_c_1 = pow( ans_f, -1) - 1;
    }
}

```

```
    }  
    else  
    {  
        /* f < -3.09 */  
        ans_f = 0.001;  
        if ( debug == 1 )  
        {  
            test_a = log(ans_f);  
            test_b = (n - i + 0.5);  
            printf("%6.3f / %6.3f\n",test_a,test_b);  
        }  
        one = log(ans_f) / ( n - i + 0.5 );  
        if ( debug == 1 )  
        {  
            test_a = log(1-ans_f);  
            test_b = i - 0.5;  
            printf("%6.3f / %6.3f\n",test_a,test_b);  
        }  
        two = log(1 - ans_f) / ( i - 0.5 );  
        z_c_1 = pow( ans_f, -1) - 1;  
    }  
}  
if ( debug != 0 )  
{  
    printf(" Fo = %6.3f\n",ans_f);  
}  
z = one + two;  
if ( debug == 1 )  
{
```

```

    printf("%6.3f + %6.3f = %6.3fn",one,two,z);
}
sum_z += z;

z_c_2 = ( (n - 0.5) / (i-0.75) ) - 1;
z_c_3 = z_c_1 / z_c_2;
z_c_4 = log(z_c_3);
z_c = pow( z_c_4, 2);
sum_z_c += z_c;
if ( debug == 2 )
{
    printf("log(Fo) = log(%6.3f / %6.3f) = %6.3fn",z_c_1,z_c_2,z_c_4);
}
}
ans_z = -(sum_z);
ans_z_a = ans_z;
ans_z_c = sum_z_c;
printf("Z_a : %6.3fn",ans_z_a);
printf("Z_c : %6.3fn",ans_z_c);
}

void random_e(void)
{
    int value,i,min = 3,max ;
    int n;

    if( first_time )
    {
        first_time = FALSE;

```

```
    srand( (unsigned int)time( NULL ) );
}
n = all - 1;
max = all - 2;
for(i=2;i<all;i++)
{
    value = (min + rand() % max);
    e_random[i] = e_1[value-2];
}
}

void create_data(void)
{
    int i,n;

    n = all;
    x_data[0] = average;
    x_data[1] = average;
    for(i=2; i<n; i++)
    {
        x_data[i] = ( (num1*(inputs[i-1]-average)) + (num2*(inputs[i-2]-average)) + e_random[i]
+ average );
    }
}

void process_w(int loop)
{
```

```

float
w_sum,w_avr=0,w_num1=0,w_num2=0,w_num3=0,w_r1,w_r1_up,w_r_down,w_r2,w_r2_u
p;
float w_g,w_g1,w_g2,w_d,w_value_t,w_value_t2,w_w1=0,w_w2=0,w_w3=0;
float w_e_sort[MAX];
int debug = 0;
int i,k,n,w_sort_order=0;

n = all;

/* Find W average */
w_sum = 0;
for(i=0; i < n; i++)
{
    w_sum += x_data[i];
}
w_avr = w_sum / n;
if ( debug == 1 )
{
    printf (" W_avr : %6.3f / %d = %6.3f\n\n",w_sum,n,w_avr);
}

/* Find W r1 */
w_r1 = 0;
w_r1_up = 0;
for(i=1; i<n; i++)
{
    w_r1_up += ( ( x_data[i] - w_avr )*( x_data[i-1] - w_avr ) );
}

```

```

w_r_down = 0;
for(i=0; i < n; i++)
{
    w_r_down += ( ( x_data[i] - w_avr )*( x_data[i] - w_avr ) );
}

w_r1 = w_r1_up / w_r_down;
if ( debug == 1 )
{
    printf (" W_R1 : %6.3f / %6.3f = %6.3f\n\n",w_r1_up,w_r_down,w_r1);
}

/* Find W r2 */
w_r2 = 0;
w_r2_up = 0;
for(i=2; i<n; i++)
{
    w_r2_up += ( ( x_data[i] - w_avr )*( x_data[i-2] - w_avr ) );
}
w_r2 = w_r2_up / w_r_down;
if ( debug == 1 )
{
    printf (" W_R2 : %6.3f / %6.3f = %6.3f\n\n",w_r2_up,w_r_down,w_r2);
}

/* Find W num1 */
w_num1 = ( w_r1 * (1 - w_r2) ) / ( 1 - (w_r1 * w_r1) );
if ( debug == 1 )

```

```

{
    printf (" W_num1 : %6.3f\n",w_num1);
}

/* Find W num2 */
w_num2 = ( ( w_r2 - (w_r1 * w_r1) ) / ( 1 - (w_r1 * w_r1) ) );
if ( debug == 1 )
{
    printf (" W_num2 : %6.3f\n",w_num2);
}

/* Find Num3 */
w_num3 = 0;
w_sort_order = 0;
for(i=2; i< n; i++)
{
    w_sum = 0;
    w_sum = ( ( x_data[i]-w_avr ) - (w_num1 * (x_data[i-1] - w_avr)) - (w_num2 * (x_data[i-
2] - w_avr)) );
    w_e_sort[w_sort_order++] = w_sum;
    w_num3 += power( w_sum, 2);
}
w_sum = w_num3 / (n-2);
w_num3 = sqrt(w_sum);
if ( debug == 1 )
{
    printf (" W_num3 : %6.3f\n\n",w_num3);
}

```

```

bubbleSort(w_e_sort,w_sort_order);

/*
for(i=0; i< w_sort_order; i+ +)
{
    printf("w_e_sort[%d] : %6.3f\n",i,w_e_sort[i]);
}
*/

for(k=1;k<n; k++)
{
    w_g = ( 1 / sqrt( ((float)k/(float)n) * (1-((float)k/(float)n)) ) );

    w_d = (float)k/(float)n;

    w_g1 = power( w_d, 2 );

    w_g2 = ( power( w_d, 2) / (float)(w_d+1) );

    w_sum = 0;
    w_value_t = t_w(n,k,w_num3,w_e_sort);
    /*    printf("w_value_t : %6.3f\n",w_e_sort);*/

/*
    printf(" G1 : %6.3f\n",w_g);
    printf(" G2 : %6.3f\n",w_g1);
    printf(" G3 : %6.3f\n",w_g2);
    printf(" T : %6.3f\n",w_value_t);
    printf(" D : %6.3f\n\n",w_d);
*/
}

```

```

w_value_t2 = power( (w_value_t - w_d), 2);

w_sum = power( w_g, 2) * w_value_t2;
w_w1 += w_sum;

w_sum = power( w_g1, 2) * w_value_t2;
w_w2 += w_sum;

w_sum = power( w_g2, 2) * w_value_t2;
w_w3 += w_sum;
}
if ( debug == 1 )
{
printf ("\nW1 : %6.3f\n",w_w1);
printf ("W2 : %6.3f\n",w_w2);
printf ("W3 : %6.3f\n\n",w_w3);
}
f_w1[loop] = w_w1;
f_w2[loop] = w_w2;
f_w3[loop] = w_w3;
}

float t_w(unsigned n,unsigned k,float value_num3,float w_e[])
{
float sum=0,w_ans=0,e=0,f=0,r_ans,f_value,f_100,test,t_ans=0;
int i,r;
int debug = 0;
unsigned int abs_r;

```

```

if ( debug == 2 )
{
    printf("\n == Start T %d ==\n\n",k);
}
for(i=1; i<=n-2; i++)
{
    e = w_e[i-1];
    f = e / value_num3;
/*    printf("e / value_num3 = %6.3f / %6.3f = %6.3f\n",e,value_num3,f);*/

/* Round floating Point */

    f_100 = f * 100;
    r = f * 100;

    if ( ((f_100 - r) >= 0.5)||((f_100 - r) <= -0.5) )
    {
        if (r >= 0)
        {
            ++r;
        }
        else
        {
            --r;
        }
    }
/*    printf ("f,r = %f,%d\n",f,r);*/

/*    printf (" E / v = %6.3f / %6.3f = %6.3f\n",e,num3,f);*/

```

```

abs_r = abs(r);
if ( ( f >= -3.09) && ( f <= 3.09 )
{
    if ( f >= 0 )
    {
        f_value = table[abs_r] ;
/*    printf("%f : table[%d] = %f\n",f,r,table[r]);*/
/*    f_value = table[abs_r] * w_ans;*/
    }
    else
    {
        r_ans = 1 - table[abs_r];
        f_value = ( 1 - table[abs_r] ) ;
/*    printf("%f : table[%d] = %f\n",f,r,r_ans);*/
/*    f_value = ( 1 - table[abs_r] ) * w_ans;*/
    }
}
else
{
    if ( f >= 0 )
    {
        f_value = 1;
    }
    else
    {
        f_value = 0; /* 1 - 1 */
    }
}
if ( debug == 2 )

```

```

    {
        printf("Start w\n");
    }
    w_ans = w(n,i,k);

    sum = ( f_value * w_ans );
    t_ans += sum;
    if ( debug == 2)
    {
        printf(" F = %6.3f , w = %6.3f , Sum : %6.3f\n",f_value,w_ans,sum);
        printf(" T_ans : %6.3f\n",t_ans);
    }
}
/* printf("W(%d,%d)(X%d) = Fo(%6.3f/%6.3f) * %6.3f\n",n,k,n,e,num3,w(n,i,k));*/
return ( t_ans);

}

void show_w()
{
    int i=94;
    printf("W1[%d] : %6.3f\n",(i+1),f_w1[i]);
    printf("W2[%d] : %6.3f\n",(i+1),f_w2[i]);
    printf("W3[%d] : %6.3f\n",(i+1),f_w3[i]);
}

void write_output(char out[],float ans_w1,float ans_w2,float ans_w3)
{
    FILE *fout;

```

```
/* --- Open file for read data --- */
if( (fout = fopen(out,"w")) == NULL)
{
    fprintf(stderr,"Can't Open %s\n",out);
    exit(1);
}

fprintf(fout,"W1   : %6.3f\n",ans_w1);
fprintf(fout,"W2   : %6.3f\n",ans_w2);
fprintf(fout,"W3   : %6.3f\n",ans_w3);
fprintf(fout,"Za   : %6.3f\n",ans_z_a);
fprintf(fout,"Zc   : %6.3f\n",ans_z_c);
fprintf(fout,"W1[95] : %6.3f\n",f_w1[94]);
fprintf(fout,"W2[95] : %6.3f\n",f_w2[94]);
fprintf(fout,"W3[95] : %6.3f\n",f_w3[94]);
fclose(fout);
}

void keep_date()
{
    time_t t1;
    char logfile[] = "./log";
    struct tm *tm_ptr;
    FILE *ftime;
    if( (ftime = fopen(logfile,"a")) == NULL)
    {
        fprintf(stderr,"Can't Open %s\n",logfile);
        exit(1);
    }
}
```

```
(void) time(&t1);  
tm_ptr = gmtime(&t1);  
fprintf(ftime,"Time : %02d:%02d:%02d\n",(tm_ptr->tm_hour+7),tm_ptr->tm_min,tm_ptr->tm_sec );  
fclose(ftime);  
  
}
```