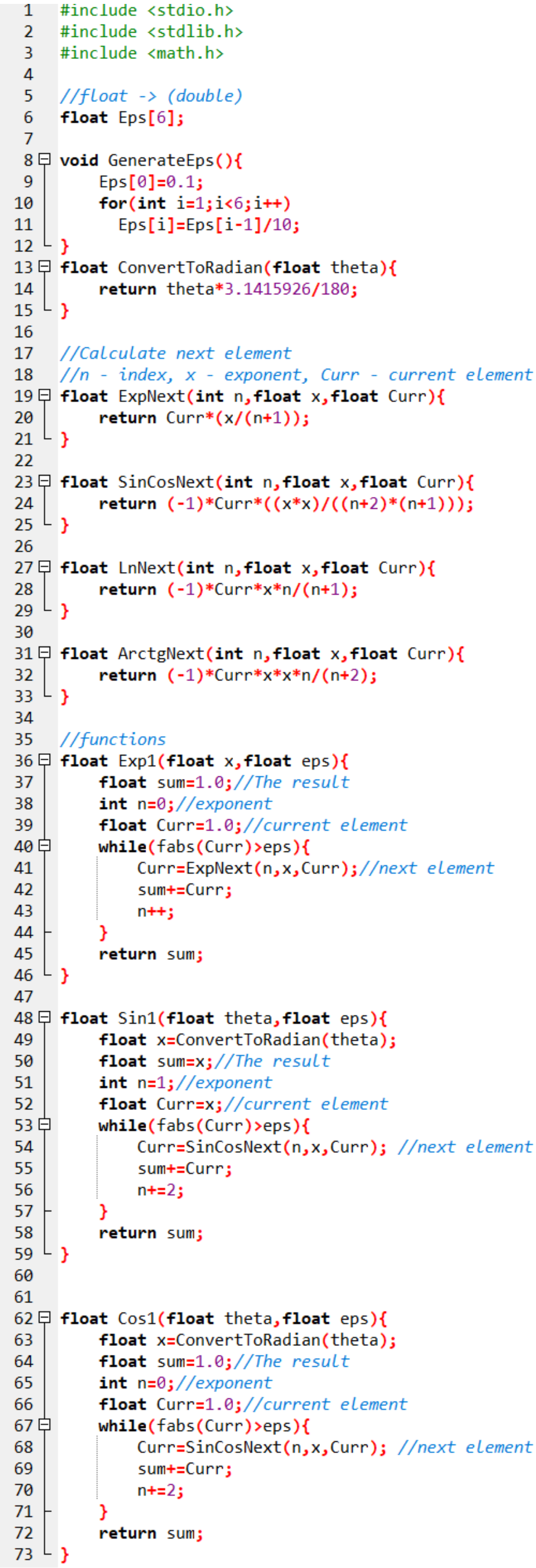
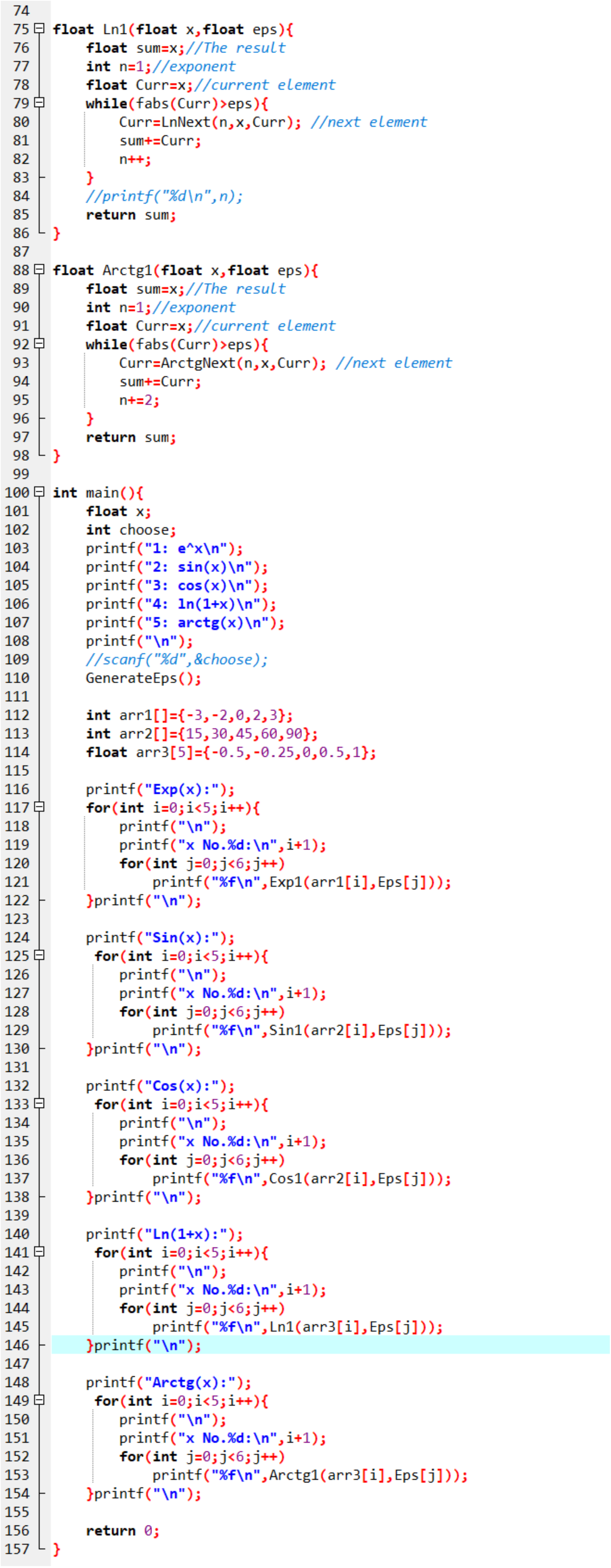
Calculation with Talyor’s formula

Code by Xumingchuan

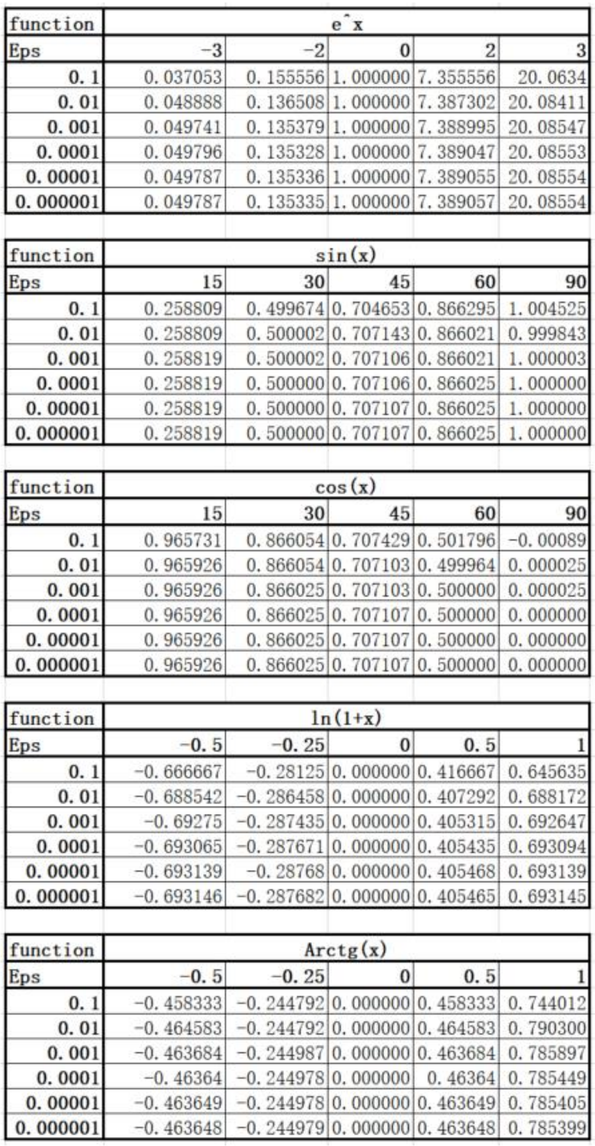


```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

//float -> (double)
float Eps[6];

void GenerateEps(){
    Eps[0]=0.1;
    for(int i=1;i<6;i++)
      Eps[i]=Eps[i-1]/10;
}
float ConvertToRadian(float theta){
    return theta*3.1415926/180;
}

//Calculate next element
//n - index, x - exponent, Curr - current element
float ExpNext(int n,float x,float Curr){
    return Curr*(x/(n+1));
}

float SinCosNext(int n,float x,float Curr){
    return (-1)*Curr*((x*x)/((n+2)*(n+1)));
}

float LnNext(int n,float x,float Curr){
    return (-1)*Curr*x*n/(n+1);
}

float ArctgNext(int n,float x,float Curr){
    return (-1)*Curr*x*x*n/(n+2);
}

//functions
float Exp1(float x,float eps){
    float sum=1.0;//The result
    int n=0;//exponent
    float Curr=1.0;//current element
    while(fabs(Curr)>eps){
        Curr=ExpNext(n,x,Curr);//next element
        sum+=Curr;
        n++;
    }
    return sum;
}

float Sin1(float theta,float eps){
    float x=ConvertToRadian(theta);
    float sum=x;//The result
    int n=1;//exponent
    float Curr=x;//current element
    while(fabs(Curr)>eps){
        Curr=SinCosNext(n,x,Curr); //next element
        sum+=Curr;
        n+=2;
    }
    return sum;
}


float Cos1(float theta,float eps){
    float x=ConvertToRadian(theta);
    float sum=1.0;//The result
    int n=0;//exponent
    float Curr=1.0;//current element
    while(fabs(Curr)>eps){
        Curr=SinCosNext(n,x,Curr); //next element
        sum+=Curr;
        n+=2;
    }
    return sum;
}
```



```c

float Ln1(float x,float eps){
    float sum=x;//The result
    int n=1;//exponent
    float Curr=x;//current element
    while(fabs(Curr)>eps){
        Curr=LnNext(n,x,Curr); //next element
        sum+=Curr;
        n++;
    }
    //printf("%d\n",n);
    return sum;
}

float Arctg1(float x,float eps){
    float sum=x;//The result
    int n=1;//exponent
    float Curr=x;//current element
    while(fabs(Curr)>eps){
        Curr=ArctgNext(n,x,Curr); //next element
        sum+=Curr;
        n+=2;
    }
    return sum;
}

int main(){
    float x;
    int choose;
    printf("1: e^x\n");
    printf("2: sin(x)\n");
    printf("3: cos(x)\n");
    printf("4: ln(1+x)\n");
    printf("5: arctg(x)\n");
    printf("\n");
    //scanf("%d",&choose);
    GenerateEps();

    int arr1[]={-3,-2,0,2,3};
    int arr2[]={15,30,45,60,90};
    float arr3[5]={-0.5,-0.25,0,0.5,1};

    printf("Exp(x):");
    for(int i=0;i<5;i++){
        printf("\n");
        printf("x No.%d:\n",i+1);
        for(int j=0;j<6;j++)
            printf("%f\n",Exp1(arr1[i],Eps[j]));
    }printf("\n");

    printf("Sin(x):");
     for(int i=0;i<5;i++){
        printf("\n");
        printf("x No.%d:\n",i+1);
        for(int j=0;j<6;j++)
            printf("%f\n",Sin1(arr2[i],Eps[j]));
    }printf("\n");

    printf("Cos(x):");
     for(int i=0;i<5;i++){
        printf("\n");
        printf("x No.%d:\n",i+1);
        for(int j=0;j<6;j++)
            printf("%f\n",Cos1(arr2[i],Eps[j]));
    }printf("\n");

    printf("Ln(1+x):");
     for(int i=0;i<5;i++){
        printf("\n");
        printf("x No.%d:\n",i+1);
        for(int j=0;j<6;j++)
            printf("%f\n",Ln1(arr3[i],Eps[j]));
    }printf("\n");

    printf("Arctg(x):");
     for(int i=0;i<5;i++){
        printf("\n");
        printf("x No.%d:\n",i+1);
        for(int j=0;j<6;j++)
            printf("%f\n",Arctg1(arr3[i],Eps[j]));
    }printf("\n");

    return 0;
}
```

Tables:



function	e^x				
Eps	-3	-2	0	2	3
0.1	0.037053	0.155556	1.000000	7.355556	20.0634
0.01	0.048888	0.136508	1.000000	7.387302	20.08411
0.001	0.049741	0.135379	1.000000	7.388995	20.08547
0.0001	0.049796	0.135328	1.000000	7.389047	20.08553
0.00001	0.049787	0.135336	1.000000	7.389055	20.08554
0.000001	0.049787	0.135335	1.000000	7.389057	20.08554

function	sin(x)				
Eps	15	30	45	60	90
0.1	0.258809	0.499674	0.704653	0.866295	1.004525
0.01	0.258809	0.500002	0.707143	0.866021	0.999843
0.001	0.258819	0.500002	0.707106	0.866021	1.000003
0.0001	0.258819	0.500000	0.707106	0.866025	1.000000
0.00001	0.258819	0.500000	0.707107	0.866025	1.000000
0.000001	0.258819	0.500000	0.707107	0.866025	1.000000

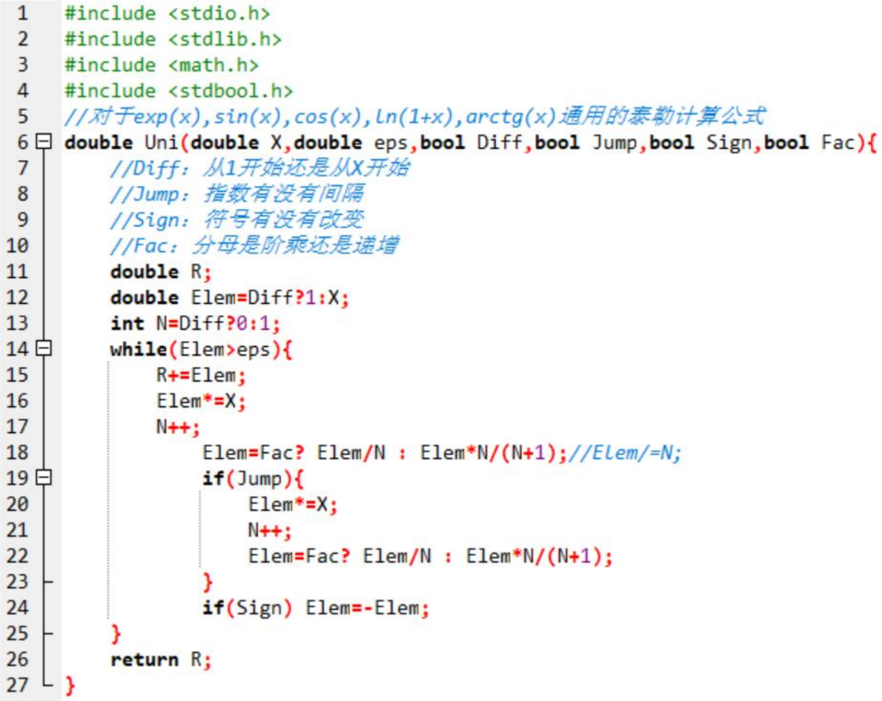
function	cos(x)				
Eps	15	30	45	60	90
0.1	0.965731	0.866054	0.707429	0.501796	-0.00089
0.01	0.965926	0.866054	0.707103	0.499964	0.000025
0.001	0.965926	0.866025	0.707103	0.500000	0.000025
0.0001	0.965926	0.866025	0.707107	0.500000	0.000000
0.00001	0.965926	0.866025	0.707107	0.500000	0.000000
0.000001	0.965926	0.866025	0.707107	0.500000	0.000000

function	ln(1+x)				
Eps	-0.5	-0.25	0	0.5	1
0.1	-0.666667	-0.28125	0.000000	0.416667	0.645635
0.01	-0.688542	-0.286458	0.000000	0.407292	0.688172
0.001	-0.69275	-0.287435	0.000000	0.405315	0.692647
0.0001	-0.693065	-0.287671	0.000000	0.405435	0.693094
0.00001	-0.693139	-0.28768	0.000000	0.405468	0.693139
0.000001	-0.693146	-0.287682	0.000000	0.405465	0.693145

function	Arctg(x)				
Eps	-0.5	-0.25	0	0.5	1
0.1	-0.458333	-0.244792	0.000000	0.458333	0.744012
0.01	-0.464583	-0.244792	0.000000	0.464583	0.790300
0.001	-0.463684	-0.244987	0.000000	0.463684	0.785897
0.0001	-0.46364	-0.244978	0.000000	0.46364	0.785449
0.00001	-0.463649	-0.244978	0.000000	0.463649	0.785405
0.000001	-0.463648	-0.244979	0.000000	0.463648	0.785399

Универсальная формула Тейлора

Для exp(x), sin(x), cos(x), ln(1+x), и arctg(x)



```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <stdbool.h>
//对于exp(x),sin(x),cos(x),ln(1+x),arctg(x)通用的泰勒计算公式
double Uni(double X,double eps,bool Diff,bool Jump,bool Sign,bool Fac){
    //Diff: 从1开始还是从X开始
    //Jump: 指数有没有间隔
    //Sign: 符号有没有改变
    //Fac: 分母是阶乘还是递增
    double R;
    double Elem=Diff?1:X;
    int N=Diff?0:1;
    while(Elem>eps){
        R+=Elem;
        Elem*=X;
        N++;
            Elem=Fac? Elem/N : Elem*N/(N+1);//Elem/=N;
            if(Jump){
                Elem*=X;
                N++;
                Elem=Fac? Elem/N : Elem*N/(N+1);
            }
            if(Sign) Elem=-Elem;
    }
    return R;
}
```

Комментарии для параметров:

1.double X : Входное значение

2.double eps: Точность

3.bool Diff: Ряд начинается с 1 **(true)**, или X **(false)**

4.bool Jump: Пропустить ли степень. ЧЕРЕЗ степень – **true**; ПОСЛЕДОВАТЕЛЬНЫЙ степень – **false.**

5.bool Sign:Меняет ли знак. Менять –**true;** НЕ менять - **false.**

6.bool Fac: Быстро ли растёт знаменатель . Быстро(*) – **true;** Медленно(+) - **false.**

	Diff	**Jump**	**Sign**	**Fac**
Exp(x)	true	false	false	true
Sin(x)	false	true	true	true
Cos(x)	true	true	true	true
Ln(1+x)	false	false	true	false
Arctg(x)	false	true	true	false