

Calculation with Talyor's formula

Code by Xumingchuan

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  //float -> (double)
6  float Eps[6];
7
8  void GenerateEps(){
9      Eps[0]=0.1;
10     for(int i=1;i<6;i++)
11         Eps[i]=Eps[i-1]/10;
12 }
13 float ConvertToRadian(float theta){
14     return theta*3.1415926/180;
15 }
16
17 //Calculate next element
18 //n - index, x - exponent, Curr - current element
19 float ExpNext(int n,float x,float Curr){
20     return Curr*(x/(n+1));
21 }
22
23 float SinCosNext(int n,float x,float Curr){
24     return (-1)*Curr*((x*x)/((n+2)*(n+1)));
25 }
26
27 float LnNext(int n,float x,float Curr){
28     return (-1)*Curr*x*n/(n+1);
29 }
30
31 float ArctgNext(int n,float x,float Curr){
32     return (-1)*Curr*x*x*n/(n+2);
33 }
34
35 //functions
36 float Exp1(float x,float eps){
37     float sum=1.0;//The result
38     int n=0;//exponent
39     float Curr=1.0;//current element
40     while(fabs(Curr)>eps){
41         Curr=ExpNext(n,x,Curr);//next element
42         sum+=Curr;
43         n++;
44     }
45     return sum;
46 }
47
48 float Sin1(float theta,float eps){
49     float x=ConvertToRadian(theta);
50     float sum=x;//The result
51     int n=1;//exponent
52     float Curr=x;//current element
53     while(fabs(Curr)>eps){
54         Curr=SinCosNext(n,x,Curr); //next element
55         sum+=Curr;
56         n+=2;
57     }
58     return sum;
59 }
60
61
62 float Cos1(float theta,float eps){
63     float x=ConvertToRadian(theta);
64     float sum=1.0;//The result
65     int n=0;//exponent
66     float Curr=1.0;//current element
67     while(fabs(Curr)>eps){
68         Curr=SinCosNext(n,x,Curr); //next element
69         sum+=Curr;
70         n+=2;
71     }
72     return sum;
73 }
```

```

74
75 float Ln1(float x, float eps){
76     float sum=x; //The result
77     int n=1; //exponent
78     float Curr=x; //current element
79     while(fabs(Curr)>eps){
80         Curr=LnNext(n,x,Curr); //next element
81         sum+=Curr;
82         n++;
83     }
84     //printf("%d\n",n);
85     return sum;
86 }
87
88 float Arctg1(float x, float eps){
89     float sum=x; //The result
90     int n=1; //exponent
91     float Curr=x; //current element
92     while(fabs(Curr)>eps){
93         Curr=ArctgNext(n,x,Curr); //next element
94         sum+=Curr;
95         n+=2;
96     }
97     return sum;
98 }
99
100 int main(){
101     float x;
102     int choose;
103     printf("1: e^x\n");
104     printf("2: sin(x)\n");
105     printf("3: cos(x)\n");
106     printf("4: ln(1+x)\n");
107     printf("5: arctg(x)\n");
108     printf("\n");
109     //scanf("%d",&choose);
110     GenerateEps();
111
112     int arr1[]={-3,-2,0,2,3};
113     int arr2[]={15,30,45,60,90};
114     float arr3[5]={-0.5,-0.25,0,0.5,1};
115
116     printf("Exp(x):");
117     for(int i=0;i<5;i++){
118         printf("\n");
119         printf("x No.%d:\n",i+1);
120         for(int j=0;j<6;j++){
121             printf("%f\n",Exp1(arr1[i],Eps[j]));
122         }printf("\n");
123
124     printf("Sin(x):");
125     for(int i=0;i<5;i++){
126         printf("\n");
127         printf("x No.%d:\n",i+1);
128         for(int j=0;j<6;j++){
129             printf("%f\n",Sin1(arr2[i],Eps[j]));
130         }printf("\n");
131
132     printf("Cos(x):");
133     for(int i=0;i<5;i++){
134         printf("\n");
135         printf("x No.%d:\n",i+1);
136         for(int j=0;j<6;j++){
137             printf("%f\n",Cos1(arr2[i],Eps[j]));
138         }printf("\n");
139
140     printf("Ln(1+x):");
141     for(int i=0;i<5;i++){
142         printf("\n");
143         printf("x No.%d:\n",i+1);
144         for(int j=0;j<6;j++){
145             printf("%f\n",Ln1(arr3[i],Eps[j]));
146         }printf("\n");
147
148     printf("Arctg(x):");
149     for(int i=0;i<5;i++){
150         printf("\n");
151         printf("x No.%d:\n",i+1);
152         for(int j=0;j<6;j++){
153             printf("%f\n",Arctg1(arr3[i],Eps[j]));
154         }printf("\n");
155
156     return 0;
157 }

```

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	$\text{Arctg}(x)$				
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 $\text{arctg}(x)$

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4  #include <stdbool.h>
5  //对于exp(x),sin(x),cos(x),ln(1+x),arctg(x)通用的泰勒计算公式
6  double Uni(double X,double eps,bool Diff,bool Jump,bool Sign,bool Fac){
7      //Diff: 从1开始还是从X开始
8      //Jump: 指数有没有间隔
9      //Sign: 符号有没有改变
10     //Fac: 分母是阶乘还是递增
11     double R;
12     double Elem=Diff?1:X;
13     int N=Diff?0:1;
14     while(Elem>eps){
15         R+=Elem;
16         Elem*=X;
17         N++;
18         Elem=Fac? Elem/N : Elem*N/(N+1); //Elem/=N;
19         if(Jump){
20             Elem*=X;
21             N++;
22             Elem=Fac? Elem/N : Elem*N/(N+1);
23         }
24         if(Sign) Elem=-Elem;
25     }
26     return R;
27 }
```

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