

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №2.1
з дисципліни
«Алгоритми і структури даних»

Виконав:
Студент групи ІМ-34
Никифоров Артем Михайлович
Номер у списку групи: 16

Перевірила:
Молчанова А. А.

Постановка задачі

Дане натуральне число n . Знайти суму перших n членів ряду чисел, заданого

рекурентною формулою. Розв'язати задачу *трьома способами*:

- 1) у програмі використати рекурсивну функцію, яка виконує обчислення і членів ряду, і суми на рекурсивному спуску;
- 2) у програмі використати рекурсивну функцію, яка виконує обчислення і членів ряду, і суми на рекурсивному поверненні;
- 3) у програмі використати рекурсивну функцію, яка виконує обчислення членів ряду на рекурсивному спуску, а обчислення суми на рекурсивному поверненні.

Варіант № 16

$$F_1 = x; \quad F_{i+1} = F_i \cdot (2i - 1)^2 \cdot x^2 / (4i^2 + 2i), \quad i > 0;$$
$$\sum_{i=1}^n F_i = \arcsin x, \quad |x| < 1.$$

1) Обчислення членів ряду і суми на рекурсивному спуску:

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

```
double x;
```

```
int n;
```

```
double expression(int n){
```

```
    return (2 * n - 1) * (2 * n - 1) * (x * x) / (4 * n * n + 2 * n);
```

```
}
```

```
double rec(double prev, int curr, double sum){
```

```
    double elem = (curr != 1) ? prev * expression( curr- 1) : x;
```

```
    sum += elem;
```

```
    if (curr == n){
```

```
        return sum;
```

```
    }
```

```
    else{
```

```
        return rec(elem, curr + 1, sum);
```

```
    }
```

```
}
```

```
double cpi(double arg, int totup){
```

```
    x = arg;
```

```
    n = totup;
```

```
    return rec(0, 1, 0);
```

```
}
```

```
int main(){
```

```
    double x = -1;
```

```
while (x >= 1 || x <= -1){  
    printf("Input x:\n");  
    scanf("%lf", &x);  
}  
printf("Input n:\n");  
scanf("%d", &n);  
double sum = cpi(x, n);  
printf("%.30lf", sum);  
return 0;  
}
```

2) Обчислення членів ряду і суми на рекурсивному поверненні:

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

```
double x;
```

```
double expression(int n){
```

```
    return (2 * n - 1) * (2 * n - 1) * (x * x) / (4 * n * n + 2 * n);
```

```
}
```

```
double rec(int n, double *sum){
```

```
    if (n == 1){
```

```
        *sum += x;
```

```
        return x;
```

```
    }
```

```
    double elem = rec(n - 1, sum) * expression(n - 1);
```

```
    *sum += elem;
```

```
    return elem;
```

```
}
```

```
void cpi(double arg, int totup, double *sum){
```

```
    *sum = 0;
```

```
    x = arg;
```

```
    rec(totup, sum);
```

```
}
```

```
int main(){
```

```
    double arg = -1;
```

```
    while (arg >= 1 || arg <= -1){
```

```
        printf("Input x:\n");
```

```
    scanf("%lf", &arg);  
}  
unsigned int totup;  
printf("Input n:\n");  
scanf("%u", &totup);  
double sum;  
cpi(arg, totup, &sum);  
printf("%.30lf", sum);  
return 0;  
}
```

3) Обчислення членів ряду на рекурсивному спуску, а обчислення суми на рекурсивному поверненні:

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

```
double x;
```

```
int n;
```

```
double expression(int n){
```

```
    return (2 * n - 1) * (2 * n - 1) * (x * x) / (4 * n * n + 2 * n);
```

```
}
```

```
double rec(double prev, int curr){
```

```
    double elem = (curr != 1) ? prev * expression(curr - 1) : x;
```

```
    if (curr == n){
```

```
        return elem;
```

```
    }
```

```
    else{
```

```
        double sum = elem + rec(elem, curr + 1);
```

```
        return sum;
```

```
    }
```

```
}
```

```
double cpi(double arg, int count){
```

```
    x = arg;
```

```
    n = count;
```

```
    return rec(0, 1);
```

```
}
```

```
int main(){
```

```
double arg = -1;
while (arg >= 1 || arg <= -1){
    printf("Input x:\n");
    scanf("%lf", &arg);
}
unsigned int count;
printf("Input n:\n");
scanf("%d", &count);
double sum = cpi(arg, count);
printf("%.30lf", sum);
return 0;
}
```


- Циклічний варіант рішення задачі

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

```
double x;
```

```
double expression(int n){  
    return (2 * n - 1) * (2 * n - 1) * (x * x) / (4 * n * n + 2 * n);  
}
```

```
double cpi(double arg, int totup){  
    x = arg;  
    double sum = arg;  
    double elem = arg;  
    for (int i = 2; i <= totup; i++) {  
        elem *= expression(i - 1);  
        sum += elem;  
    }  
    return sum;  
}
```

```
int main(){  
    double arg = -1;  
    while (arg >= 1 || arg <= -1){  
        printf("Input x:\n");  
        scanf("%lf", &arg);  
    }  
}
```

```
    unsigned int totup;  
    printf("Input n:\n");  
    scanf("%d", &totup);  
    double sum = cpi(arg, totup);  
    printf("%.30lf", sum);  
    return 0;  
}
```

Результати тестування програм:

При $x = 0.555$ та $n = 5$

1) Обчислення членів ряду і суми на рекурсивному спуску:

```
Input x:
0.555
Input n:
5
0.588317567981034739155177248904
Process finished with exit code 0
```

2) Обчислення членів ряду і суми на рекурсивному поверненні:

```
Input x:
0.555
Input n:
5
0.588317567981034739155177248904
Process finished with exit code 0
```

3) Обчислення членів ряду на рекурсивному спуску, а обчислення суми на рекурсивному поверненні:

```
Input x:
0.555
Input n:
5
0.588317567981034739155177248904
Process finished with exit code 0
```

4) Циклічний варіант рішення задачі

```
C:\Users\User\CLION\Projects\untitled27\  
Input x:  
0.555  
Input n:  
5  
0.588317567981034739155177248904  
Process finished with exit code 0
```











Результат обчислень на калькуляторі:

Input

$$\sin^{-1}(0.555)$$

Result

0.5883629661366031469009222131370445642989165247107760699360020369

1	 $a = 0.555$ 
	 -10  10
2	$f(x,i) = \frac{x \cdot (2i-1)^2 \cdot a^2}{(4i^2 + 2i)}$ 
3	$B_2 = f(0.555,1)$  $B_2 = 0.0284923125$
4	$B_3 = f(B_2,2)$  $B_3 = 0.00394935505102$
5	$B_4 = f(B_3,3)$  $B_4 = 0.000724107196184$
6	$B_5 = f(B_4,4)$  $B_5 = 0.000151793233835$
7	$S = a + B_2 + B_3 + B_4 + B_5$  $S = 0.588317567981$

При $x = -0.143$ та $n = 5$

1) Обчислення членів ряду і суми на рекурсивному спуску:

```
Input x:
-0.143
Input n:
5
-0.143491907965087678311633112571
Process finished with exit code 0
```

2) Обчислення членів ряду і суми на рекурсивному поверненні:

```
C:\Users\user\OneDrive\Documents\oneDrive
Input x:
-0.143
Input n:
5
-0.143491907965087678311633112571
Process finished with exit code 0
```

3) Обчислення членів ряду на рекурсивному спуску, а обчислення суми на рекурсивному поверненні:

```
Input x:
-0.143
Input n:
5
-0.143491907965087678311633112571
Process finished with exit code 0
|
```

4) Циклічний варіант рішення задачі

```
Input x:
-0.143
Input n:
5
-0.143491907965087678311633112571
Process finished with exit code 0
```











Результат обчислень на калькуляторі:

Input

$\sin^{-1}(-0.143)$

Result

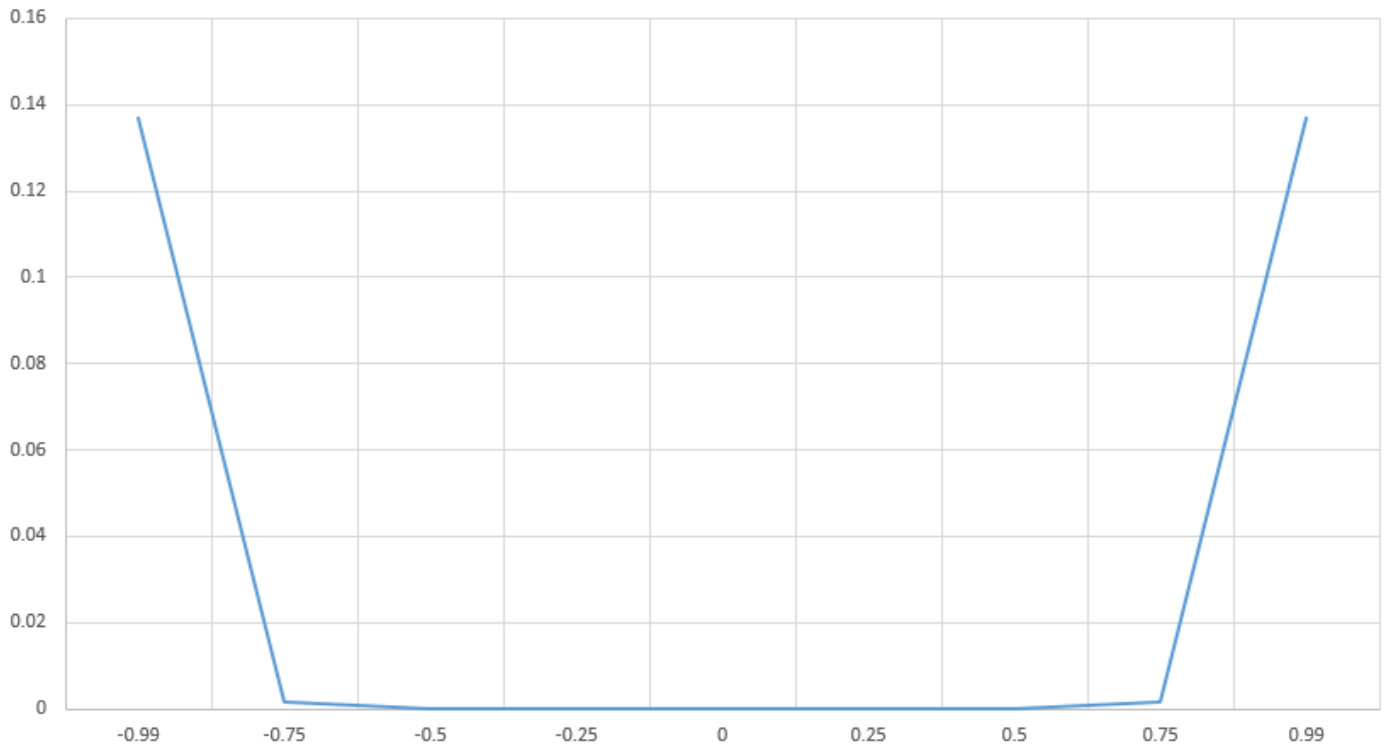
-0.143491907976711609730169504223632156461954684081904421180036084
...

1	 $a = -0.143$ 
	 -10  10
2	$f(x,i) = \frac{x \cdot (2i-1)^2 \cdot a^2}{(4i^2 + 2i)}$ 
3	$B_2 = f(-0.143, 1)$  $B_2 = -0.000487367833333$
4	$B_3 = f(B_2, 2)$  $B_3 = -0.00000448478317072$
5	$B_4 = f(B_3, 3)$  $B_4 = -5.4588887535 \times 10^{-8}$
6	$B_5 = f(B_4, 4)$  $B_5 = -7.596961097 \times 10^{-10}$
7	$S = a + B_2 + B_3 + B_4 + B_5$  $S = -0.143491907965$

Залежність похибки від x

-0.99	-0.75	-0.5	-0.25	0	0.25	0.5	0.75	0.99
0.13685	0.00171	0.00001	$6 \cdot 10^{-9}$	0	$6 \cdot 10^{-9}$	0.00001	0.00171	0.13685

Залежність похибки від x



Висновок:

Після виконання Лабораторної роботи №1 я засвоїв матеріал та набув практичного досвіду створення рекурсивних алгоритмів та відповідних їм програм.