

Национальный исследовательский ядерный университет «МИФИ»

Институт интеллектуальных кибернетических систем

Кафедра №12 «Компьютерные системы и технологии»



ОТЧЁТ

О выполнении лабораторной работы № 2

"Вычисление значений числовых рядов и функций с заданной точностью"

Студент: Ким В.А.

Группа: Б22-703

Преподаватель: Овчаренко Е.С.

Москва - 2022

1. Формулировка индивидуального задания

Вариант № 5.

Вычислить значение функции в точке при помощи разложения в ряд:

$$\ln(1 + x + x^2 + x^3) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1} + [1 + (-1)^n](-1)^{\frac{n}{2}+1}}{n} x^n$$

где $-1 < x \leq 1$.

2. Описание использованных типов данных

В данной работе использовались тип данных «float» для работы с числами с плавающей запятой, «int» для работы с целыми числами

3. Описание использованного алгоритма

3.1 Алгоритм вычисления с определенным количеством итераций

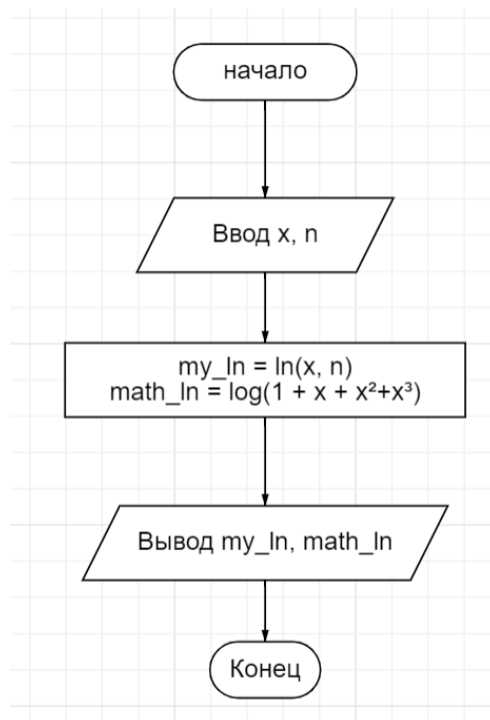


Рис. 1: Блок-схема алгоритма работы функции main(for)

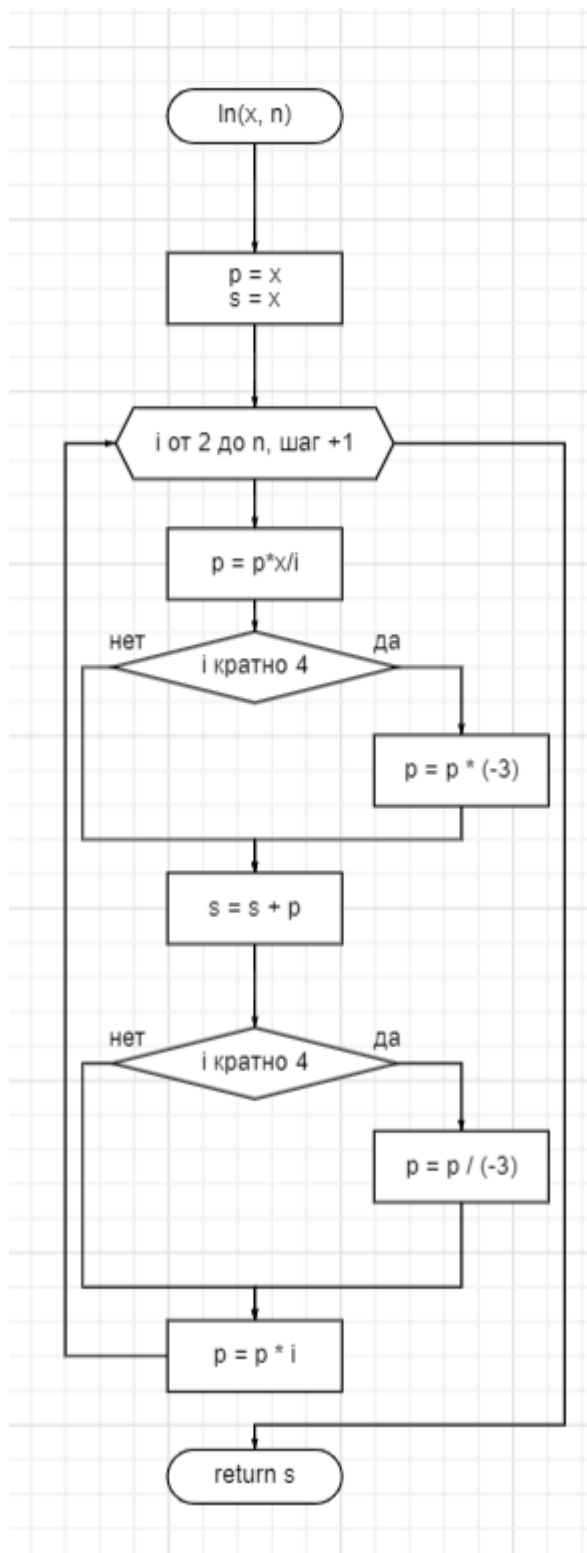


Рис. 2: Блок-схема алгоритма работы функции $\ln(x, n)$

3.2 Алгоритм вычисления с определенной точностью

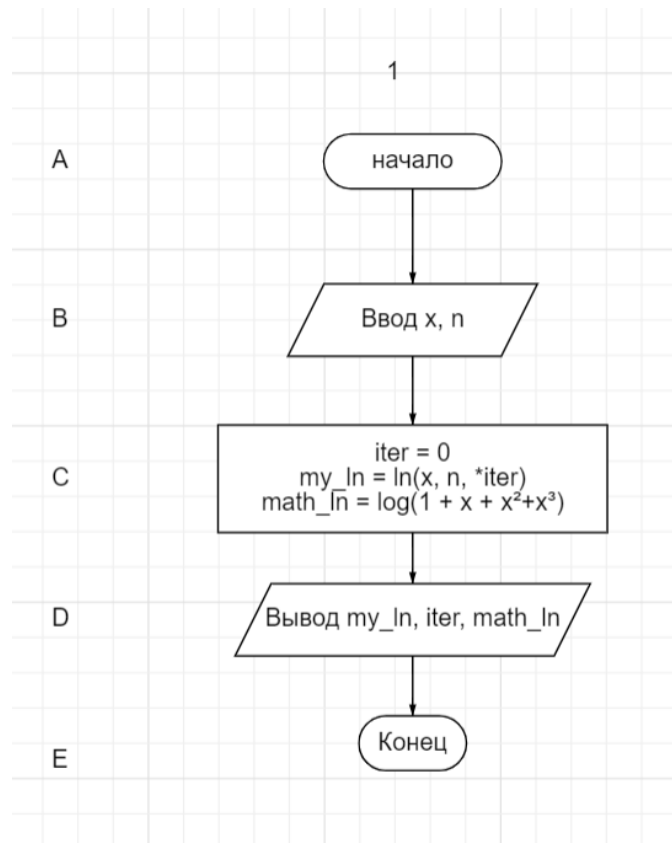


Рис. 3: Блок-схема алгоритма работы функции main(while)

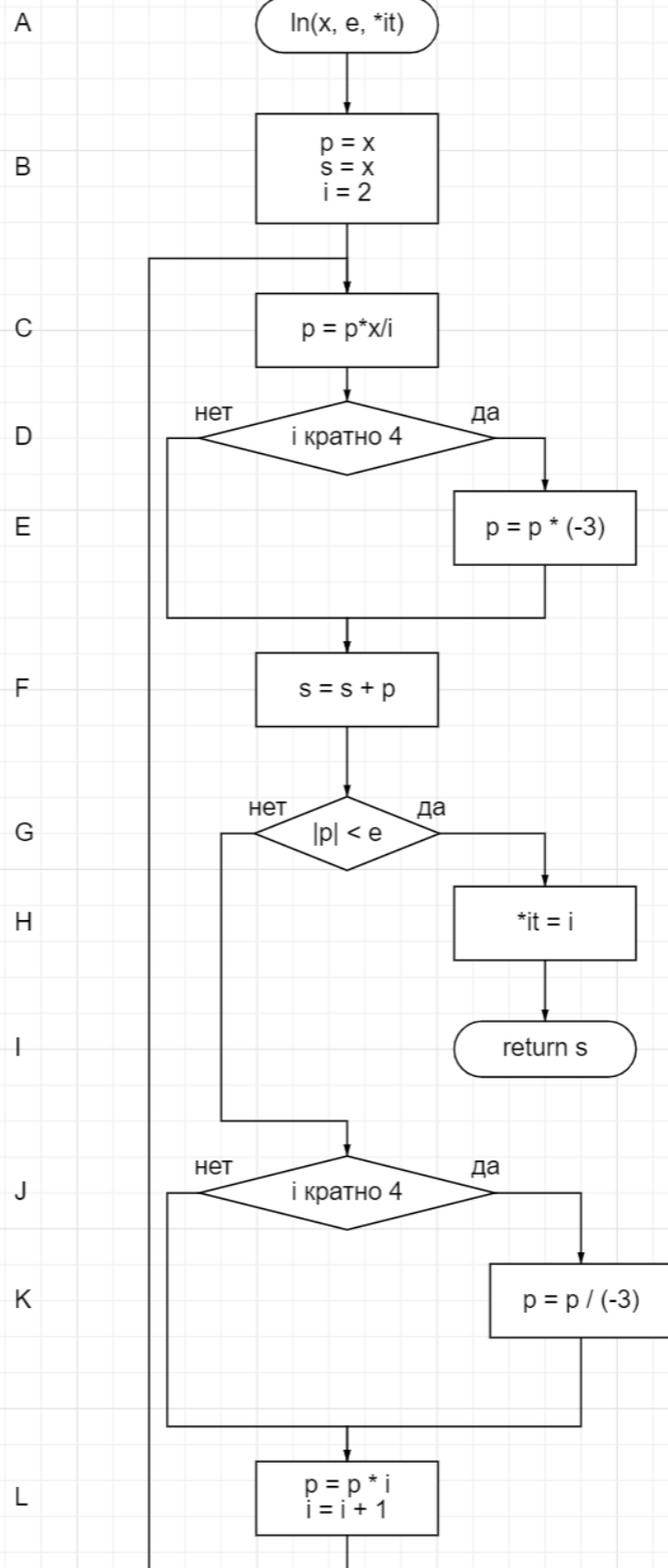


Рис. 3: Блок-схема алгоритма работы функции $\ln(x, e, *it)$

4. Исходные коды разработанных программ

Листинг 1: Исходные код программы main, файл: lab_02.c (цикл с определенным количеством итераций)

```
1  #include <stdio.h>
2  #include <math.h>
3
4  float get_float();
5  int get_num();
6  float ln(float x, int n);
7
8  int main() {
9      float x = get_float();
10     int n = get_num();
11
12     printf("%f", ln(x, n) );
13     printf("\n");
14     printf("%f", log(1 + x + x*x + x*x*x) );
15
16     return 0;
17 }
18
19 float get_float() {
20     float input;
21
```

```
22     printf("Input x (-1; 1]: ");
23     int flag = scanf("%f", &input);
24     while (getchar() != '\n') {}
25
26     if (!flag || !(input <= 1.f && input > -1.f) ) {
27         printf("Error...\n");
28         return get_float();
29     }
30     return input;
31 }
32
33 int get_num() {
34     float input;
35
36     printf("Input natural N: ");
37     int flag = scanf("%f", &input);
38     while (getchar() != '\n') {}
39
40     if (!flag || input <= 0 || ( input - (int)input > 0 ) ) {
41         printf("Error...\n");
42         return get_num();
43     }
44     return (int)input;
45 }
46
47 float ln(float x, int n) {
48     float p = x, s = x;
49
50     for (int i = 2; i <= n; ++i) {
51         p *= x / i;
52         if (i % 4 == 0) p *= (-3);
53
54         s += p;
```

```
55     if (i % 4 == 0) p /= (-3);
56     p *= i;
57 }
58
59 return s;
60 }
```

Листинг 2: Исходные код программы main, файл: lab_02_.c (цикл с определенной точностью)

```
1  #include <stdio.h>
2  #include <math.h>
3
4  float get_float();
5  float get_e();
6  float ln(float x, float n, int* it);
7  float absolut(float num);
8
9  int main() {
10     float x = get_float();
11     float e = get_e();
12     int count_it;
13
14     printf("%f", ln(x, e, &count_it) );
15     printf("\n");
16     printf("Iterations: %d", count_it);
17     printf("\n");
18     printf("%f", log(1 + x + x*x + x*x*x) );
19     printf("\n");
20
21     return 0;
22 }
23 float get_float() {
```



```
24     float input;
25
26     printf("Input x (-1; 1]: ");
27     int flag = scanf("%f", &input);
28     while (getchar() != '\n') {}
29
30     if (!flag || !(input <= 1.f && input > -1.f) ) {
31         printf("Error...\n");
32         return get_float();
33     }
34     return input;
35 }
36
37 float get_e() {
38     float input;
39
40     printf("Input positive E: ");
41     int flag = scanf("%f", &input);
42     while (getchar() != '\n') {}
43
44     if (!flag || input <= 0) {
45         printf("Error...\n");
46         return get_e();
47     }
48     return input;
49 }
50
51 float ln(float x, float e, int* it) {
52     float p = x, s = x;
53     int i;
54
55     for (i = 2;; ++i) {
```

```

56     p *= x / i;
57     if (i % 4 == 0) p *= (-3);
58
59
60     s += p;
61     if (absolut(p) < e) break;
62     if (i % 4 == 0) p /= (-3);
63     p *= i;
64 }
65
66 *it = i;
67
68 return s;
69 }
70
71 float absolut(float num) {
72     return num >= 0? num : -num;
73 }
74
75

```

5. Описание текстовых примеров

Программа №1

Значение x, n	Ожидаемое значение В	Полученное значение В
1, 1000	1.386294	1.384795
0.05, 100	0.051287	0.051287
-0.75, 10	-0.940007	-0.923800
0, 1	0.000000	0.0000000

Программа №1

Значение x , e	Ожидаемое значение B	Полученное значение B (кол-во итераций)
1, 0.01	1.386294	1.381321 (101)
0.05, 0.001	0.051292	0.051292 (3)
-0.9, 0.000001	-1.709258	-1.709258(89)
0, 1	0.000000	0.000000(2)

6. Скриншоты

6.1 Программа №1

```
[kim.va@unix:~/labs/lab_02]$ ./lab_02
Input x (-1; 1]: 1
Input natural N: 1000
1.384795
1.386294
[kim.va@unix:~/labs/lab_02]$ ./lab_02
Input x (-1; 1]: 0.05
Input natural N: 1000
0.051287
0.051287
[kim.va@unix:~/labs/lab_02]$ ./lab_02
Input x (-1; 1]: 0.05
Input natural N: 100
0.051287
0.051287
[kim.va@unix:~/labs/lab_02]$ ./lab_02
Input x (-1; 1]: -0.75
Input natural N: 10
-0.923800
-0.940007
[kim.va@unix:~/labs/lab_02]$ ./lab_02
Input x (-1; 1]: 0
Input natural N: 1
0.000000
0.000000
```

6.2 Программа №2

```

[kim.va@unix:~/labs/lab_02]$ ./lab_02_
Input x (-1; 1]: 1
Input positive E: 0.01
1.381321
Iterations: 101
1.386294

[kim.va@unix:~/labs/lab_02]$ ./lab_02_
Input x (-1; 1]: 0.05
Input positive E: 0.001
0.051292
Iterations: 3
0.051287

[kim.va@unix:~/labs/lab_02]$ ./lab_02_
Input x (-1; 1]: -0.9
Input positive E: 0.000001
-1.709252
Iterations: 89
-1.709258

[kim.va@unix:~/labs/lab_02]$ ./lab_02_
Input x (-1; 1]: 0
Input positive E: 1
0.000000
Iterations: 2
0.000000

```

7. Выводы

В ходе выполнения данной работы на примере программ, выполняющих приближительные вычисления математических функций с помощью разложения функций на ряд Тейлора. Мы научились:

1. Вычислять функции при помощи разложения ее в ряд с заданной точностью
2. Работать с указателями
3. Обрабатывать ввод пользователя