

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

**Лабораторна робота
№2**

з дисципліни
«Алгоритмізації та програмування»

Виконав:
студент групи КН-108
Жеребецький Олег

Львів – 2018 р.

5. Зміст звіту

1. Постановка завдання.
2. Текст програми.
3. Результат розв'язку конкретного варіанту.

Постановка завдання

- Знайти рекурентну формулу для знаходження наступного члену ряду,

$$a_n = \frac{2^n n!}{n^n}.$$

загальний члени якого

- Обчислити наступний член ряду .
- Знайти суму ряду з точністю $\epsilon=0.0001$.
- Пояснити отримані результати.

Програма розв'язання завдання №1

Варіант 1 (з використанням функції Факторіал)

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

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

```
int fact(int i)
```

```
{
```

```
    int j=1;
```

```
    for(;i>0;j=j*i--);
```

```
    return j;
```

```
}
```

```
int main(void)
```

```
{
```

```
    double x,a = 0.0, eps=0.0001;
```

```

int n=0;

do
{
    x = (pow(2,n)*fact(n))/pow(n,n);
    a+= x;
    printf("Result %lf With n= %i With last x= %lf\n", a,n,x);
    n++;
}while(x > eps);
}

```

Варіант 2 (З використанням рекурентної формули)

```

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

int main(void)
{
    double x=1,a = 1, eps=0.0001;
    int n=2;

    do
    {

        x*= 2*pow(n,n)/pow(n+1,n);
        a+= x;
        printf("Result %lf With n= %i With last x= %lf\n", a,n,x);
        n++;
    }while(x > eps);
}

```

Результат роботи 1 варіанту програми

```
jharvard@appliance (~/.Dropbox/Labs): ./lab2-1
Result 1.000000 With n= 0 With last x= 1.000000
Result 3.000000 With n= 1 With last x= 2.000000
Result 5.000000 With n= 2 With last x= 2.000000
Result 6.777778 With n= 3 With last x= 1.777778
Result 8.277778 With n= 4 With last x= 1.500000
Result 9.506578 With n= 5 With last x= 1.228800
Result 10.494232 With n= 6 With last x= 0.987654
Result 11.277579 With n= 7 With last x= 0.783347
Result 11.892814 With n= 8 With last x= 0.615234
Result 12.372382 With n= 9 With last x= 0.479568
Result 12.743971 With n= 10 With last x= 0.371589
Result 13.030498 With n= 11 With last x= 0.286527
Result 13.250549 With n= 12 With last x= 0.220050
Result 13.302806 With n= 13 With last x= 0.052257
Result 13.304691 With n= 14 With last x= 0.001886
Result 13.304841 With n= 15 With last x= 0.000150
Result 13.304849 With n= 16 With last x= 0.000007
```

Де останній рядок і є відповідь до завдання.

Результат роботи 2 варіанту програми

```
jharvard@appliance (~/.Dropbox/Labs): ./lab2-2
Result 1.000000 With n= 0 With last x= 1.000000
Result 3.000000 With n= 1 With last x= 2.000000
Result 5.000000 With n= 2 With last x= 2.000000
Result 6.777778 With n= 3 With last x= 1.777778
Result 8.277778 With n= 4 With last x= 1.500000
Result 9.506578 With n= 5 With last x= 1.228800
Result 10.494232 With n= 6 With last x= 0.987654
Result 11.277579 With n= 7 With last x= 0.783347
Result 11.892814 With n= 8 With last x= 0.615234
Result 12.372382 With n= 9 With last x= 0.479568
Result 12.743971 With n= 10 With last x= 0.371589
Result 13.030498 With n= 11 With last x= 0.286527
Result 13.250549 With n= 12 With last x= 0.220050
Result 13.418974 With n= 13 With last x= 0.168425
Result 13.547513 With n= 14 With last x= 0.128539
Result 13.645367 With n= 15 With last x= 0.097854
Result 13.719700 With n= 16 With last x= 0.074333
Result 13.776057 With n= 17 With last x= 0.056357
Result 13.818712 With n= 18 With last x= 0.042656
Result 13.850949 With n= 19 With last x= 0.032236
Result 13.875278 With n= 20 With last x= 0.024329
Result 13.893617 With n= 21 With last x= 0.018339
```

```
Result 13.907424 With n= 22 With last x= 0.013808
Result 13.917810 With n= 23 With last x= 0.010386
Result 13.925615 With n= 24 With last x= 0.007805
Result 13.931475 With n= 25 With last x= 0.005860
Result 13.935871 With n= 26 With last x= 0.004396
Result 13.939167 With n= 27 With last x= 0.003296
Result 13.941636 With n= 28 With last x= 0.002469
Result 13.943485 With n= 29 With last x= 0.001849
Result 13.944868 With n= 30 With last x= 0.001383
Result 13.945903 With n= 31 With last x= 0.001035
Result 13.946676 With n= 32 With last x= 0.000773
Result 13.947254 With n= 33 With last x= 0.000578
Result 13.947685 With n= 34 With last x= 0.000431
Result 13.948007 With n= 35 With last x= 0.000322
Result 13.948248 With n= 36 With last x= 0.000240
Result 13.948427 With n= 37 With last x= 0.000179
Result 13.948561 With n= 38 With last x= 0.000134
Result 13.948660 With n= 39 With last x= 0.000100
```

Де останній рядок відповідь до завдання.

Пояснення результатів

При виконанні дій такого масштабу як множення або підняття до степеня Числа які мають соті, тисячні і тд. , виходять за межі визначеності типу Float, в наслідок чого відбувається заокруглення. Оскільки у нас у двох випадках різні дії (а саме: у першій програмі значно більше дій множення, що зумовлює більшу кількість заокруглень), тобто результат буде спотворюватись по різному.

Легко помітити, що до $n = 12$ включно, розбіжності немає, бо числа ще цілком нормального порядку. А починаючи з 13 дві програми заокруглюють по різному, в наслідок різної кількості заокруглень, про що було згадано вище.

Помітно, що перша програма різко зрізає числа, тому швидше прямує до виходу з циклу(член ряду швидше стає меншим за задану межу точності).

Висновок: це програма коректна до межі точності , яка дорівнює 0,220050.