

Міністерство освіти і науки, молоді та спорту України  
Національний технічний університет України  
«Київський політехнічний інститут»  
Фізико-технічний інститут

## **Лабораторна робота з програмування № 3**

**Виконав:**

Студент 2 курсу  
групи ФЕ-81  
Адамантіс М. В.

**Перевірив:**

Прогонов Д. О.

Київ 2019

## Формулювання завдання

Використовуючи оператор циклу while з передумовою та постумовою, розв'язати наступні задачі:

1. Знайти суму ряду з точністю  $\epsilon=10^{-4}$ , загальний член якого

$$a_n = \frac{(-1)^{n-1}}{n^n}$$

1) Дано натуральне число  $n$  ( $n \geq 2$ ). Знайти всі прості числа, що не перевищують число  $n$ , використовуючи решето Ератосфена. Решетом Ератосфена називається наступний метод. Випишемо підряд всі цілі числа від 2 до  $n$ . Першим простим числом є число 2. Підкреслимо його, а всі більші за нього числа, кратні числу 2, викреслимо. Першим числом в ряду чисел, що залишились, є число 3. Підкреслимо його як просте число, а всі більші за нього числа, кратні числу 3, викреслимо. Першим числом серед тих, що залишились, є число 5, оскільки 4 вже закреслено. Підкреслимо його як просте число, а всі більші за нього числа, кратні числу 5, викреслимо і т.д. |

## Програмний код

```
#!/usr/bin/env python3
```

```
import sys, os
```

```
sys.path.append(os.path.abspath(os.path.join(os.path.dirname(__file__), "..")))
```

```
from inc.errorhandling import handle
```

```
import math
```

```
def a(i):
```

```
    return pow(-1, i-1)/pow(i, i)
```

```
def main():
```

```
    EPSILON = pow(10, -4)
```

```
    try:
```

```
        a_sum = 0
```

```
        i = 0
```

```
        while True:
```

```
            curr = a(i)
```

```
            if abs(curr) <= EPSILON:
```

```
                break
```

```
            a_sum += curr
```

```
            i += 1
```

```
        print("a_({0}): {1}".format(i + 1, a_sum))
```

```
    except Exception as e:
```

```
        handle(e)
```

```

if __name__ == "__main__":
    main()

# main2.py
#!/usr/bin/env python3
import sys, os
sys.path.append(os.path.abspath(os.path.join(os.path.dirname(__file__), "..")))
from inc.errorhandling import handle

def primes_lower_than(n):
    primes = []
    arr = list(range(2, n))
    for i in range(2, n):
        if 0 not in [i % x for x in primes]:
            primes.append(i)
    return [1] + primes

try:
    n = int(input('enter n>'))
    if n <= 0:
        raise ValueError('n must be greater than 0')
except Exception as e:
    handle(e)

print(primes_lower_than(n))

```

## Приклад виконання

**IPT-2020-FE81** on  **work/m.adamantis** [] via  **v4.8.2**

→ **python3** results/lab3/main.py

a\_(7): -0.21654921296296298

**IPT-2020-FE81/results/lab3** on  **work/m.adamantis** []

→ **python3** main2.py

enter n> -6

n must be greater than 0

A git pull a day keeps the segfault away