

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Образовательная программа бакалавриата «Программная инженерия»

**МИКРОПРОЕКТ №1**

**по дисциплине «Архитектура вычислительных систем»**

**Пояснительная записка**

Исполнитель:

студент 194 группы 2 подгруппы

Е. А. Ермаченко

17 октября 2020 г.

**Москва 2020**

## Содержание

<b>1. Текст задания.....</b>	<b>2</b>
<b>2. Вычисление числа Пи посредством дзета-функции Римана .....</b>	<b>3</b>
<b>3. Реализация программы .....</b>	<b>4</b>
<b>4. Тестирование программы .....</b>	<b>6</b>
<b>Список использованной литературы.....</b>	<b>7</b>
<b>ПРИЛОЖЕНИЕ.....</b>	<b>8</b>

## 1. Текст задания

Разработать программу вычисления числа **Пи** с точностью не хуже 0,1% посредством дзета-функции Римана (использовать FPU).

## 2. Вычисление числа Пи посредством дзета-функции Римана

Дзета-функция Римана  $\zeta(s)$  определяется с помощью ряда Дирихле:

$$\zeta(s) = \frac{1}{1^s} + \frac{1}{2^s} + \frac{1}{3^s} + \dots \quad (1)$$

где  $s \in \mathbb{C}$ .

В области  $\{s \mid \operatorname{Re}(s) > 1\}$ , этот ряд сходится, является аналитической функцией и допускает аналитическое продолжение на всю комплексную плоскость без единицы. [1]

Для вычисления числа Пи с помощью дзета-функции Римана нужно взять  $\zeta(2)$  (дзета-функцию Римана от числа 2) [2]. Получается:

$$\zeta(2) = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \dots = \frac{\pi^2}{6} \quad (2)$$

После вычисления  $\zeta(2)$  нужно умножить результат на 6 и взять положительное значение корня:

$$\pi = \sqrt{\zeta(2) * 6} \quad (3)$$

### 3. Реализация программы

Для реализации данной программы была написана процедура PI.

В данной процедуре присутствует один цикл. В данном цикле мы считаем дзета-функцию Римана от следующего члена  $\zeta(2)$ , находим число Пи от полученного результата и сравниваем с предыдущим результатом числа Пи. Цикл продолжается, пока разность по модулю полученного Пи и предыдущего Пи не будет меньше или равна 0.001 (0.1%).

Далее приведена блок схема процедуры Пи:

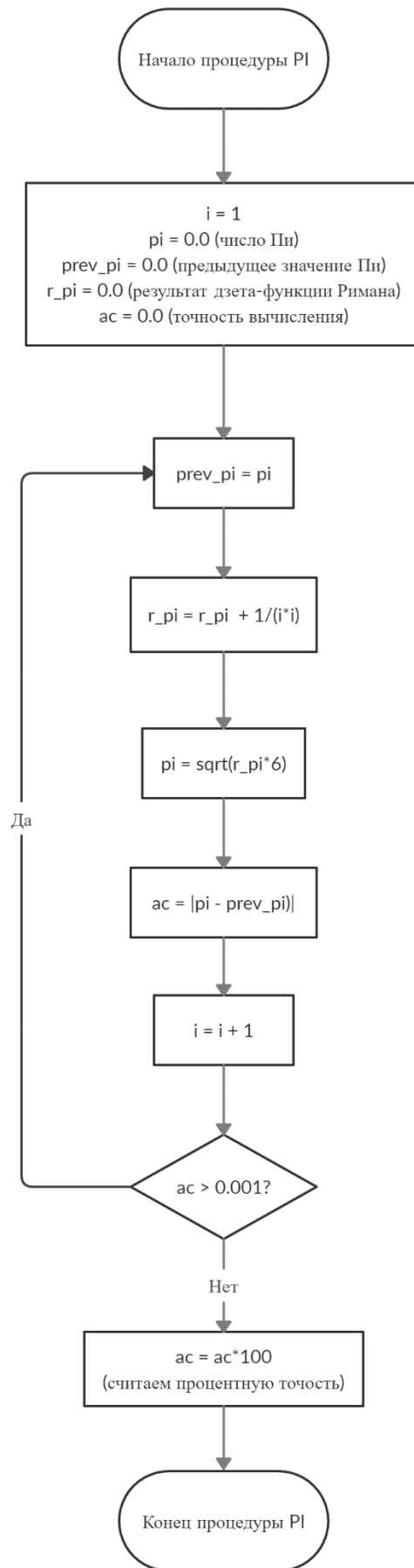


Рисунок 1. Блок – схема процедуры Пи

#### 4. Тестирование программы

Данная программа всегда выводит один результат: результат подсчета числа Пи с помощью дзета-функции Римана с точностью, не большей 0.01%.

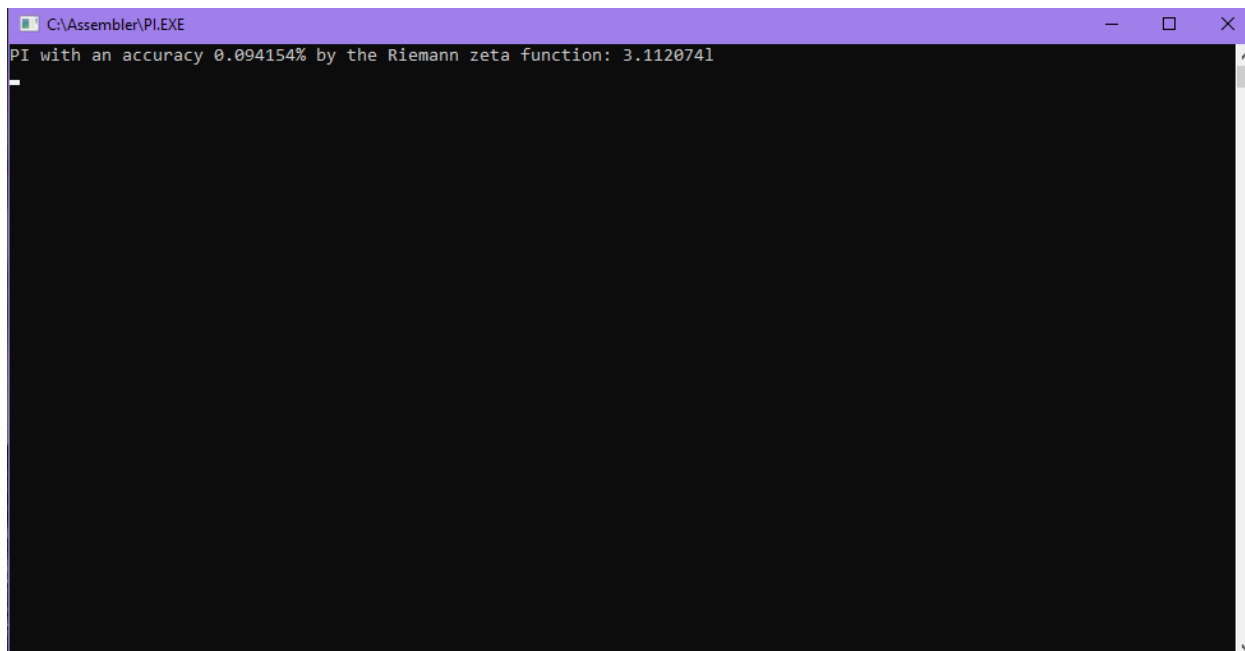


Рисунок 2. Результат программы

## **Список использованной литературы**

- 1) Титчмарш, Е.К. Теория дзета-функции Римана/Е. К. Титчмарш; пер. М. А. Евграфов. - М.: Москва, 1953г.
- 2) Пи (число) [Электронный ресурс] – Режим доступа: свободный, URL: [https://ru.wikipedia.org/wiki/Пи\\_\(число\)](https://ru.wikipedia.org/wiki/Пи_(число)) (дата обращения: 16.10.2020)



## ПРИЛОЖЕНИЕ

Код программы:

```
format PE console
entry start

include 'win32a.inc'

section '.data' data readable writable
    fmt_string db 'PI with an accuracy %lf%% by the Riemann zeta
function: %f1',13, 10, 0

    r_pi dq 0.0 ; (pi^2)/6 - результат дзета-функции Римана
    pi dq 0.0 ; само число Пи, полученное с помощью дзета-функции
Римана
    prev_pi dq 0.0 ; предыдущее число Пи, полученное с помощью
дзета-функции Римана
    ac dq 0.0 ; точность вычисления
    i dd 1 ; количество итераций цикла
    tmp dq ? ; переменная для хранения временных значений
    n_ac dq 0.001 ; требуемая точность вычисления Пи
    six dd 6 ; число 6
    hun dd 100 ; число 100
    stk dd ? ; переменная для хранения стека

PI:
    mov [stk], esp

lp:
    fld [pi]
    fstp [prev_pi] ; prev_pi=pi

    fild [i]
    fimul [i]
    fstp [tmp] ; tmp=i*i

    fld1
    fdiv [tmp]
    fstp [tmp] ; tmp=1/i*i

    fld [r_pi]
    fadd [tmp]
    fst [r_pi] ; r_pi=r_pi+1/i*i

    fimul [six]
    fsqrt
    fst [pi]; pi= sqrt(r_pi*6)

    fsub [prev_pi]
```

```

    fabs
    fstp [ac] ; ac = |pi(i) - pi(i-1)|

    fld [ac]
    fcomp [n_ac] ; |pi(i) - pi(i-1)| > 0.001?

    fstsw ax
    sahf ; перемещаем результат сравнения в регистр

    inc [i] ; i++
ja lp

    fld [ac]
    fimul [hun]
    fstp [ac] ; ac=ac*100(процентная точность)

    mov esp, [stk]
ret

section '.code' code readable executable
start:
    call PI

    invoke printf, fmt_string, dword[ac], \
        dword[ac+4], dword[pi], dword[pi+4]

    invoke getch
    invoke ExitProcess, 0

section '.idata' import data readable
    library kernel, 'kernel32.dll', \
        msvcrt, 'msvcrt.dll'

    import kernel, \
        ExitProcess, 'ExitProcess'

    include 'api\kernel32.inc'

    import msvcrt, \
        printf, 'printf', \
        getch, '_getch'

```