Отчет ИДЗ-2

Лев Алексеевич Светличный

БПИ 225

Вариант 21

Дата 2023-11-05

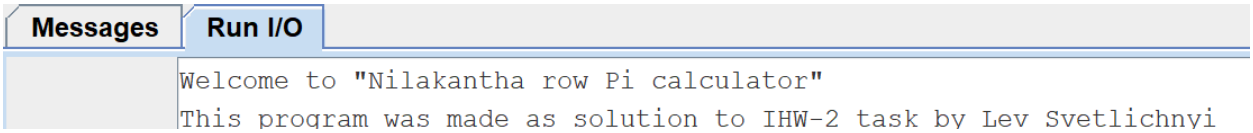
Условие

Разработать программу вычисления числа π с точностью не хуже 0,05% посредством ряда Нилаканта.

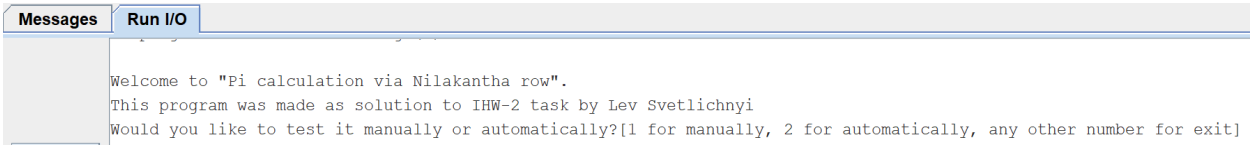
Решение

Программа имеет вид:

- Приветствие пользователя

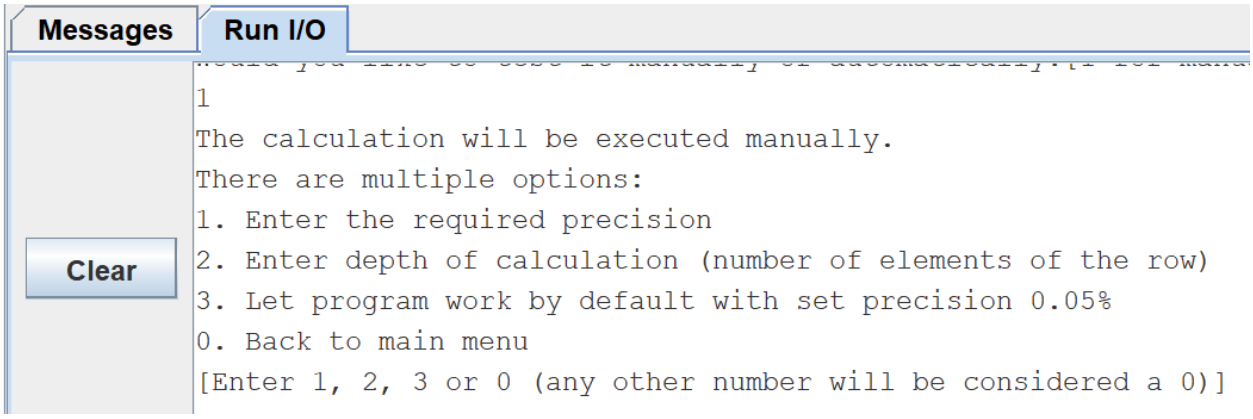


- Главное меню. Здесь пользователь может выбрать режим использования программы: ручной для задания с клавиатуры точности вычисления или глубины - количества - вычислений, и автоматический - программа самостоятельно высчитывает значения числа π для различных входных данных и выводит результат. Также пользователь может выбрать покинуть программу (в качестве реализации повтора решения).



- Ручной (manual) режим. Здесь пользователю предоставлены опции ввода точности вычисления (количества знаков после запятой), а также вычисление по умолчанию (основные требования к заданию - вычисления числа π через ряд

Нилаканта с точностью 0.05%). Также пользователь может покинуть ручной режим для выхода в главное меню.



- Автоматический (auto) режим. Здесь пользователю выводятся результаты вычисления с разной степенью точности и глубины вычислений, заданных автоматически. По завершении пользователь возвращается в главное меню.
- Выход (exit). Программа завершается.

Ряд Нилаканта

$$\pi = 3 + \frac{4}{2 \cdot 3 \cdot 4} - \frac{4}{4 \cdot 5 \cdot 6} + \frac{4}{6 \cdot 7 \cdot 8} + ... = \\ = 3 + 4 \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{2n(2n+1)(2n+2)}$$

Этот ряд используется для вычисления числа π.

В программе для этого отведены регистры:

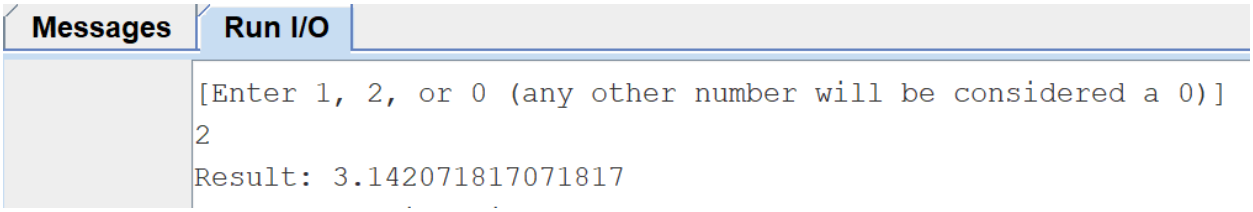
- fs0 - хранение суммы ряда π
- fs1 - счетчик элементов ряда n
- fs2 - хранение текущего знака +/-

Алгоритм:

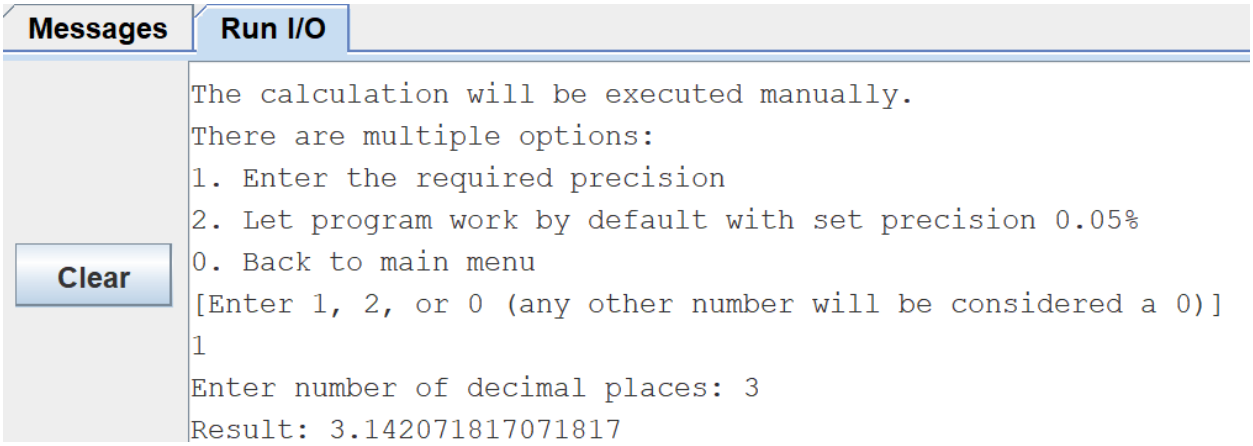
1. Получение необходимой точности

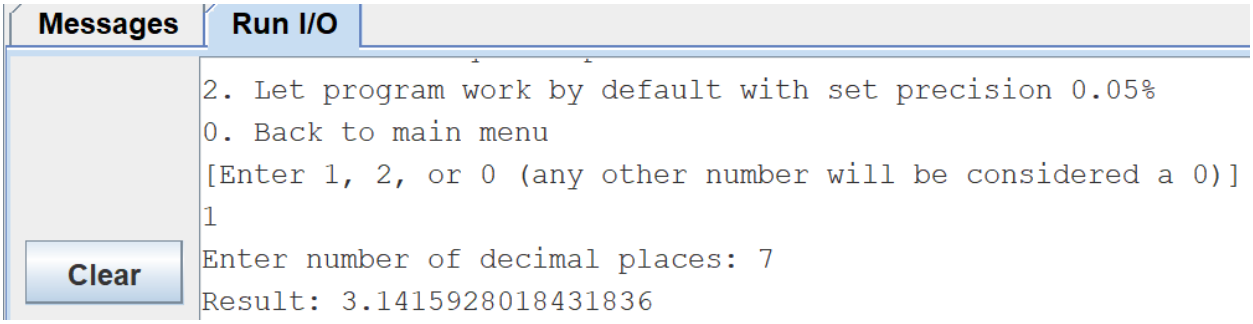
Тестовые испытания

По умолчанию с точностью 0.05%:

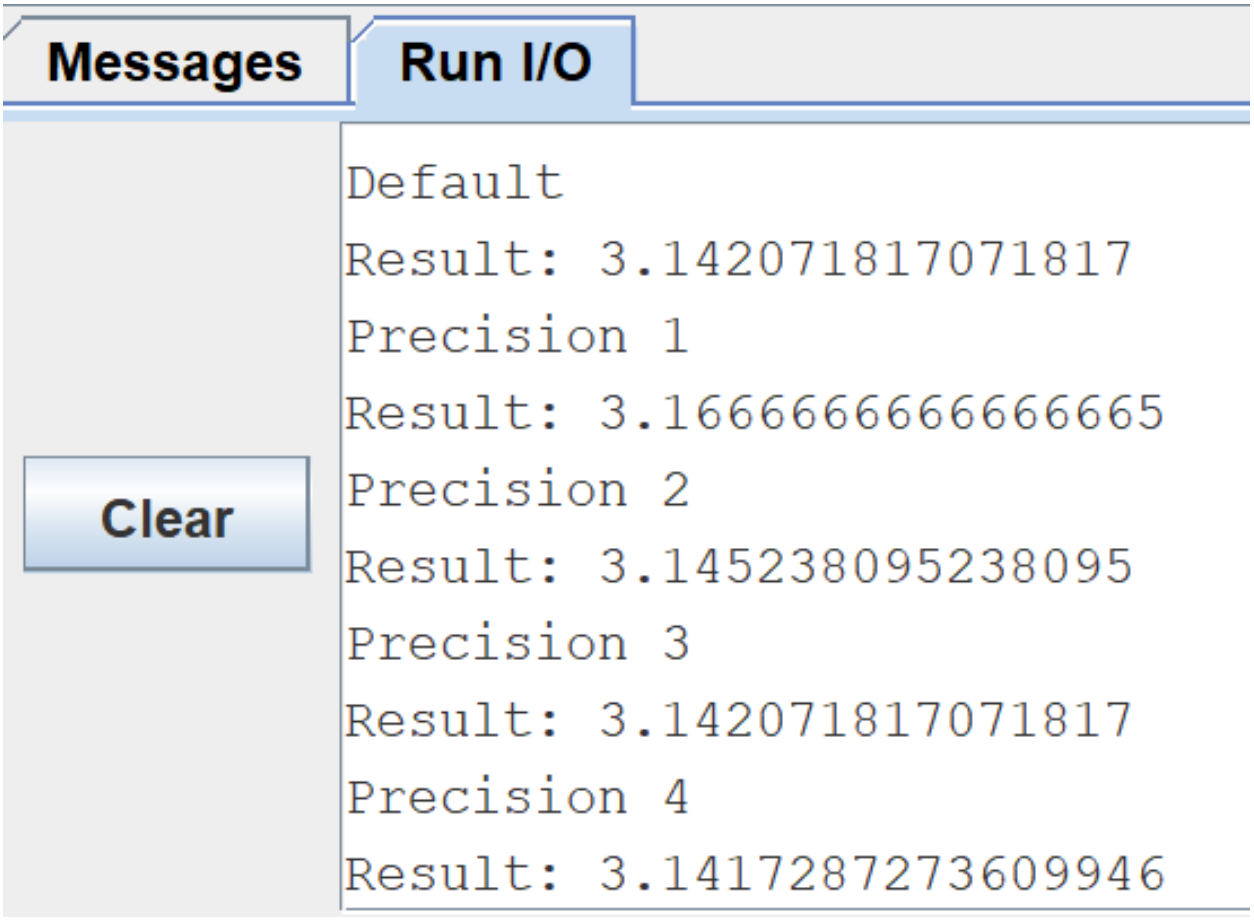


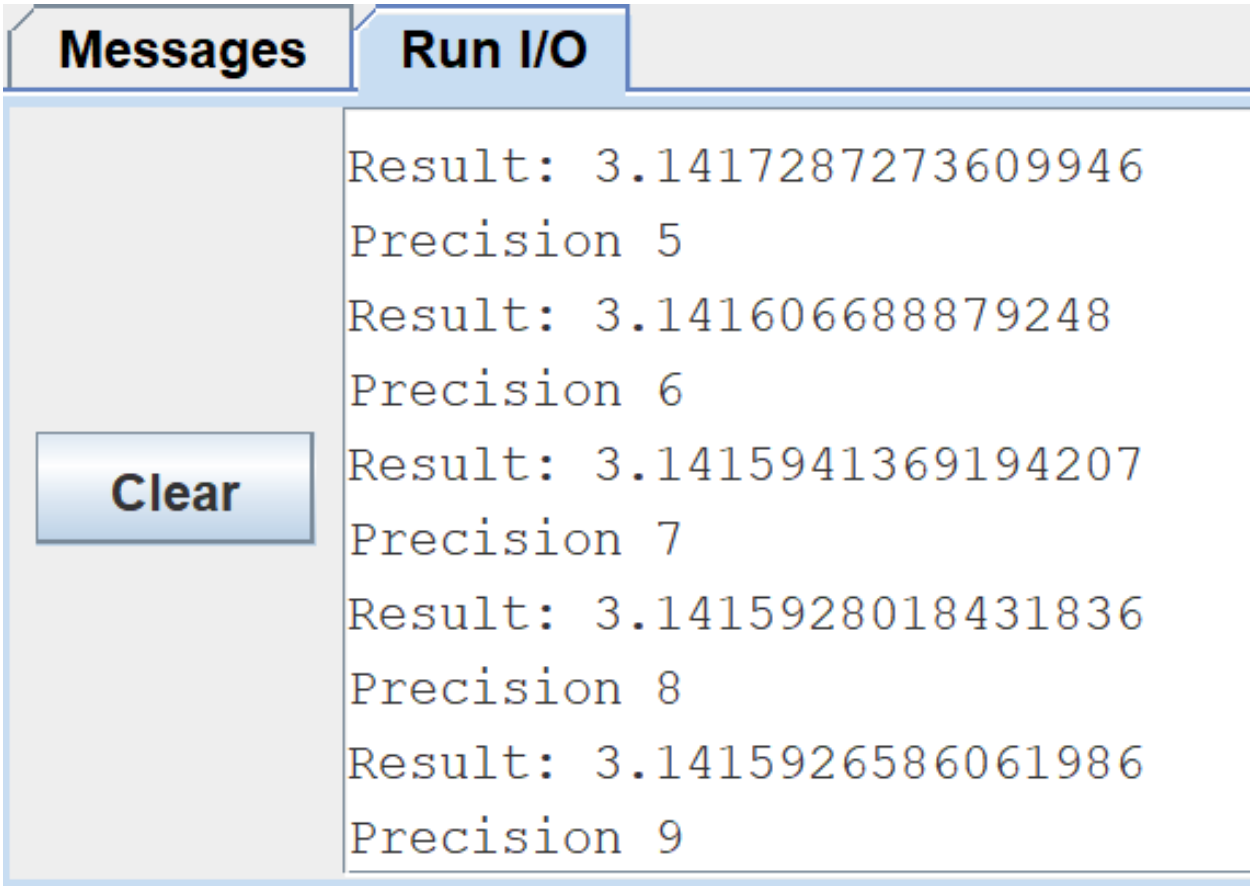
Различное количество знаков после запятой (извините, я устал и округлять не буду, но можно заметить, что первые n знаков после запятой, где n - введенная точность, являются корректными для числа π):





Автоматические тестовые прогоны:


Messages
Run I/O
Clear
Default
Result: 3.142071817071817
Precision 1
Result: 3.1666666666666665
Precision 2
Result: 3.145238095238095
Precision 3
Result: 3.142071817071817
Precision 4
Result: 3.1417287273609946



Тестовые прогоны на языке C++:

```
#include <bits/stdc++.h>
using namespace std;

double calculatePI(int prec)
{
  double prev = 1, PI = 3, n = 2, sign = 1;

  while (prec <= 0 ? abs((PI - prev) / prev) >= 0.0005 : abs(PI - prev) >= pow(10, -prec-1)) {
        prev = PI;
    PI = PI + (sign * (4 / ((n) * (n + 1)
                * (n + 2))));
    sign = sign * (-1);
    n += 2;
  }
  return PI;
}

// Driver code
int main()
{
    cout << "Enter precision: ";
```

```
    int n;
    cin >> n;
  cout << fixed << setprecision(16)
    << "\nThe approximation of Pi is "
    << calculatePI(n) << endl;
  return 0;
}
```

```
Enter precision: 0

The approximation of Pi is 3.1420718170718169

Process finished with exit code 0
```

```
Enter precision: 1

The approximation of Pi is 3.1396825396825396

Process finished with exit code 0
```

```
Enter precision: 3

The approximation of Pi is 3.1416353566793886

Process finished with exit code 0
```

```
Enter precision: 7

The approximation of Pi is 3.1415926486140626

Process finished with exit code 0
```

Удовлетворение программы требованиям

✓ 4-5 баллов

- Решение приведено в качестве программы на ассемблере (см. исходный код). Ввод осуществляется с консоли, вывод - на экран.
- В работе присутствуют комментарии.
- Подпрограммы реализованы в виде макрос, и потому это требование некорректно (особенно в связи с выполненными требованиями на 6-7 и 9 баллов относительно подпрограмм).
- В отчете представлено полное тестовое покрытие и результаты различных прогонов программы (см. раздел “Тестовые испытания”)

✓ 6-7 баллов

- Подпрограммы могут использоваться повторно с различными параметрами (собственно говоря, они и используются таким образом в “ручном режиме”).
- В подпрограммах используются локальные переменные, хранимые во “временных регистрах” t0-t6, ft0-ft11
- В местах вызова функции присутствуют комментарии, объясняющие предназначение функции и переданные параметры, а так же возвращаемый результат.

✓ 8 баллов

- Разработанные подпрограммы (макросы) могут использоваться многократно (осуществлено возможностью передавать различные наборы данных).

- В рамках программы реализовано автоматическое тестирование (”автоматический режим”), проверяющий работоспособность программы при различных данных.
- Проведены дополнительные тестовые прогоны аналогичной программы на языке C++ для сравнительной проверки результатов (см. раздел “Тестовые испытания”).

✓ 9 баллов

- В программе используются макросы для реализации ввода и вывода данных. Макросы поддерживают повторное использование с другими исходными данными.
- Макросы использовались с самого начала написание программы 😉

✓ 10 баллов

- Программа разбита на несколько файлов - 1 .asm файл и несколько .s файлов с полезными функциями. Поскольку программа разработана под несколько режимов - ручной и автоматический, потому нет нужды вручную запускать различные единицы компиляции.
- Макросы для ввода и вывода выведены в отдельную библиотеку “input_output_lib”, которую можно использовать в других программах.
- Отчет полностью заполнен всеми критериями, указанными в файле с требованиями.

Использованные источники

Получение числа Пи через ряд Нилаканта (Википедия)

Курс "Архитектура вычислительных систем"