

Национальный исследовательский университет
"Высшая школа экономики"
Факультет компьютерных наук
Программная инженерия

Микропроект №1.
Программирование на языке ассемблера.

Текст задания

ВАРИАНТ 9. Разработать программу вычисления числа π с точностью не хуже 0,05% посредством ряда Нилаканта (использовать FPU).

Решение задания

В секции данных мы объявляем все нужные для вычисления π вещественные переменные, а также строковые переменные для информирования о промежуточных и итоговых результатах.

```
section '.data' data readable writable
; strings for output
steps_string db "Calculation completed. The Nilakantha Series took %d steps.",10,0
pi_string db "Iteration %d. Calculated Pi = %lf", 10, 0
answer_string db "Required accuracy: %lf. ",10,0
accuracy_info db "Machine Pi: %lf. Final answer: %lf. Resulting error: %lf.",10,0

; variables used to calculate the answer and exit conditions
res dq 3.0
x1 dq 2.0
x2 dq 3.0
x3 dq 4.0
stepValue dq 2.0
fraction dq 0.0
numerator dq -4.0
denominator dq 0.0
sign dq -1.0
zero dq 0.0
i dd 0
accuracy dq 0.0005
prev dq 3.0
machine_pi dq ?
error dq ?
```

В следующей таблице описаны все используемые переменные, их исходные значения и их предназначения:

res	3.0	Переменная, содержащая итоговый результат вычисления π с заданной точностью.
x1	2.0	Первая из трех переменных в знаменателе формируемой на каждой итерации дроби.
x2	3.0	Вторая из трех переменных в знаменателе формируемой на

		каждой итерации дроби.
x3	4.0	Последняя из трех переменных в знаменателе формируемой на каждой итерации дроби.
stepValue	2.0	Значение, на которое каждое из чисел в знаменателе увеличивается.
fraction	0.0	Переменная, содержащая формируемую на каждой итерации дробь, которая будет прибавлена к res.
numerator	-4.0	Числитель дроби, прибавляемой к res на каждой итерации.
denominator	0.0	Знаменатель дроби, прибавляемой к res на каждой итерации.
sign	-1.0	Переменная для изменения знака дроби на каждой итерации.
zero	0.0	Начальное значение вычисления знаменателя прибавляемой дроби на каждом этапе.
i	0	Переменная для хранения количества пройденных итераций до достижения нужного результата.
accuracy	0.0005	Заданная точность вычисления.
prev	3.0	Переменная, в которой хранится res, полученный на прошлой итерации.

machine_pi	не задано	Число, равное машинному Пи.
error	не задано	Число, равное итоговому отклонению от машинного Пи.

Почти все вычисления в программе происходят в цикле piLoop. Цикл разбит на несколько блоков, каждый выполняющий определенную задачу в процессе вычисления res.

В первом блоке мы вычисляем знаменатель для текущей дроби, при этом сначала занулив его:

```
; calculating denominator of fraction that will be added: x1*x2*x3
FLD [denominator]
FMUL [zero]
FADD [x1]
FMUL [x2]
FMUL [x3]
FSTP [denominator]
```

Во втором блоке мы увеличиваем значения переменных, участвующих в формировании знаменателя, и сохраняем их в памяти:

```
; changing denominator product values for next loop: x1 +=2, x2 += 2, x3 += 2
FLD [x1]
FADD [stepValue]
FSTP [x1]
FLD [x2]
FADD [stepValue]
FSTP [x2]
FLD [x3]
FADD [stepValue]
FSTP [x3]
```

В третьем блоке мы меняем знак прибавляемой дроби путем изменения знака числителя:

```
;calculating numerator: multiplying numerator by -1
FLD [numerator]
FMUL [sign]
FSTP [numerator]
```

В четвертом блоке мы вычисляем значение прибавляемой дроби на текущей итерации:

```
; calculating fraction: +-4 / (x1 * x2 * x3)
FLD [numerator]
FDIV [denominator]
FSTP [fraction]
```

В пятом блоке мы прибавляем полученную дробь к нашему результату:

```
; adding calculated fraction to our answer
FLD [res]
FADD [fraction]
FSTP [res]
```

В шестом блоке мы инкрементируем счетчик итераций и выводим текущее состояние результата:

```
; printing current iteration and its result
add [i], 1
invoke printf, pi_string, [i], dword[res], dword[res+4]
```

В предпоследнем блоке мы проверяем условие для выхода, то есть сравниваем модуль разности текущего результата и результата, полученного на прошлой итерации, с нужной точностью, и если модуль оказывается меньше точности, то выходим из цикла piLoop:

```
; the comparison part
FLD [res]
FSUB [prev]
FABS
FCOM [accuracy]
FSTSW AX
SAHF
jnb endMet
```

В последнем блоке цикла мы присваиваем значение текущего результата результату, полученному на прошлой итерации, и начинаем новую итерацию:

```
; prev = res
FLD [res]
FSTP [prev]
jmp piLoop
```

По окончании работы цикла мы выводим количество проделанных итераций, исходную точность, а также, предварительно вычислив погрешность и записав в переменную машинное значение Π , сам машинный Π и итоговую погрешность:

```
; printing total step count and required accuracy.
invoke printf, steps_string, [i]
invoke printf, answer_string, dword[accuracy], dword[accuracy+4]

; calculating error and getting machine pi
FSTP [machine_pi]
FLDPI
FSTP [machine_pi]
FLDPI
FSUB [res]
FABS
FSTP [error]

; printing machine pi, our result and final error.
invoke printf, accuracy_info, dword[machine_pi], dword[machine_pi+4], \
    dword[res], dword[res+4], \
    dword[error], dword[error+4]
```

ПРИЛОЖЕНИЕ 1. Текст программы.

format PE console

entry start

include 'win32a.inc'

section '.code' code readable executable

start:

FINIT

piLoop:

; calculating denominator of fraction that will be added: $x1 * x2 * x3$

FLD [denominator]

FMUL [zero]

FADD [x1]

FMUL [x2]

FMUL [x3]

FSTP [denominator]

; changing denominator product values for next loop: $x1 += 2, x2 += 2, x3 += 2$

FLD [x1]

FADD [stepValue]

FSTP [x1]

FLD [x2]

FADD [stepValue]

FSTP [x2]

FLD [x3]

FADD [stepValue]

FSTP [x3]

;calculating numerator: multiplying numerator by -1

FLD [numerator]

FMUL [sign]

FSTP [numerator]

; calculating fraction: $\pm 4 / (x1 * x2 * x3)$

FLD [numerator]

FDIV [denominator]

FSTP [fraction]

; adding calculated fraction to our answer

FLD [res]

FADD [fraction]

FSTP [res]

; printing current iteration and its result

add [i], 1

invoke printf, pi_string,[i], dword[res],dword[res+4]

; the comparison part

FLD [res]

```

    FSUB [prev]
    FABS
    FCOM [accuracy]
    FSTSW AX
    SAHF
    jb endMet

```

```

    ; prev = res
    FLD [res]
    FSTP [prev]
    jmp piLoop

```

endMet:

```

    ; printing total step count and required accuracy.
    invoke printf, steps_string, [i]
    invoke printf, answer_string, dword[accuracy], dword[accuracy+4]

```

```

    ; calculating error and getting machine pi
    FSTP [machine_pi]
    FLDP
    FSTP [machine_pi]
    FLDP
    FSUB [res]
    FABS
    FSTP [error]

```

```

    ; printing machine pi, our result and final error.
    invoke printf, accuracy_info, dword[machine_pi], dword[machine_pi+4], \
        dword[res], dword[res+4], \
        dword[error], dword[error+4]

```

```

    invoke getch
    invoke ExitProcess, 0

```

section '.data' data readable writable

```

    ; strings for output
    steps_string db "Calculation completed. The Nilakantha Series took %d steps.", 10, 0
    pi_string db "Iteration %d. Calculated Pi = %lf", 10, 0
    answer_string db "Required accuracy: %lf. ", 10, 0
    accuracy_info db "Machine Pi: %lf. Final answer: %lf. Resulting error: %lf.", 10, 0

```

```

    ; variables used to calculate the answer and exit conditions
    res dq 3.0
    x1 dq 2.0
    x2 dq 3.0
    x3 dq 4.0
    stepValue dq 2.0
    fraction dq 0.0
    numerator dq -4.0
    denominator dq 0.0
    sign dq -1.0
    zero dq 0.0

```

```
i dd 0
accuracy dq 0.0005
prev dq 3.0
machine_pi dq ?
error dq ?
```

```
section '.idata' import data readable
library kernel, 'kernel32.dll',\
    msvcrt, 'msvcrt.dll',\
    user32, 'USER32.DLL'
```

```
include 'api\user32.inc'
include 'api\kernel32.inc'
import kernel,\
    ExitProcess, 'ExitProcess',\
    HeapCreate, 'HeapCreate',\
    HeapAlloc, 'HeapAlloc'
include 'api\kernel32.inc'
import msvcrt,\
    printf, 'printf',\
    sprintf, 'sprintf',\
    scanf, 'scanf',\
    getch, '_getch'
```


ПРИЛОЖЕНИЕ 2. Список используемых источников.

1. [https://newtonov.ru/chemu-ravno-chislo-pi/#:~:text=%D0%A0%D1%8F%D0%B4%20%D0%9D%D0%B8%D0%B%D0%B0%D0%BA%D0%B0%D0%BD%D1%82%D0%B0.&text=%CF%80%20%3D%203%20%2B%204%2F\(2,%D0%9F%D0%BE%20%D1%8D%D1%82%D0%BE%D0%BC%D1%83%20%D0%B8%D0%B4%D0%B5%D0%BC%20%D0%B4%D0%B0%D0%BB%D1%8C%D1%88%D0%B](https://newtonov.ru/chemu-ravno-chislo-pi/#:~:text=%D0%A0%D1%8F%D0%B4%20%D0%9D%D0%B8%D0%B%D0%B0%D0%BA%D0%B0%D0%BD%D1%82%D0%B0.&text=%CF%80%20%3D%203%20%2B%204%2F(2,%D0%9F%D0%BE%20%D1%8D%D1%82%D0%BE%D0%BC%D1%83%20%D0%B8%D0%B4%D0%B5%D0%BC%20%D0%B4%D0%B0%D0%BB%D1%8C%D1%88%D0%B)
2. https://www.frolov-lib.ru/books/bsp/v02/ch12_4.htm
3. <https://drive.google.com/file/d/1cYZ68FZQJwEwntH5b9n8LPpLmHcvX7pN/view?usp=sharing>