НИУ ВШЭ

Факультет компьютерных наук

Программная инженерия

Программа для вычисления значения биномиальной функции (1+x)m с помощью степенного ряда

Поляков Лев Алексеевич, БПИ199

Содержание

1. Текст задания
2. Применяемые расчетные методы
3. Тестовые примеры
4. Список использованных источников
5. Приложения

Текст задания

Вариант 17. Разработать программу, вычисляющую с помощью степенного ряда с точностью не хуже 0,05% значение биномиальной функции (1+x)m для заданного параметра m и x (использовать FPU)

Применяемые расчетные методы

Функция может быть представлена рядом Тейлора:

(1+x)m = 1 +

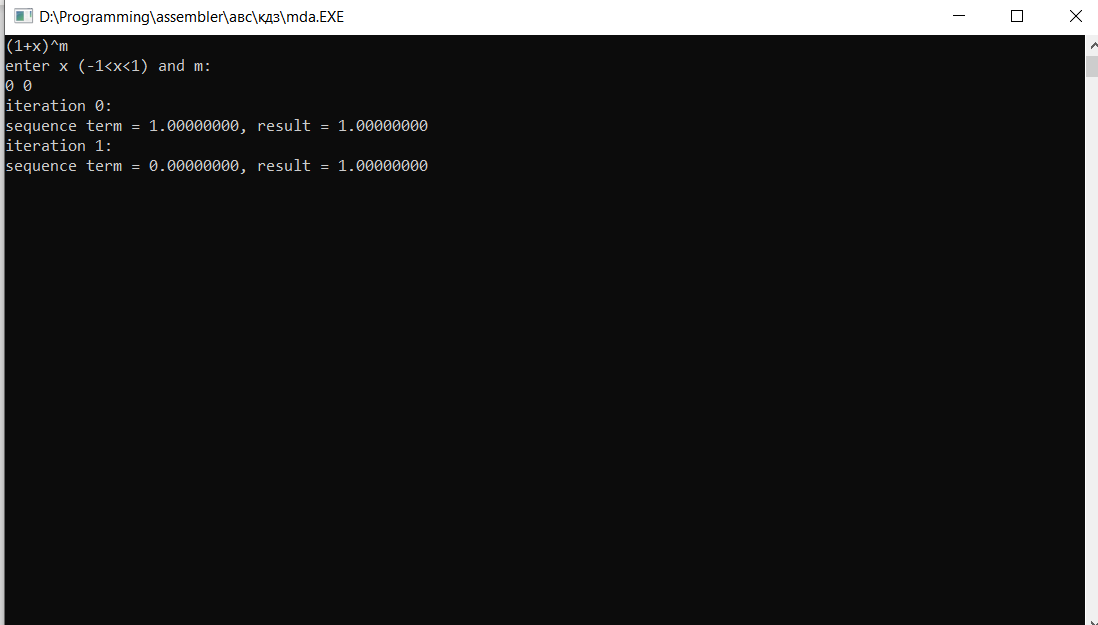
или 1 +

Заметим, что при mN формула превращается в обычный бином Ньютона, и в какой-то момент очередной член ряда станет равным 0. В остальных случаях же это неверно, и ряд с какого-то момента станет знакопеременным. В таком случае, чтобы ряд сходился, по признаку Лейбница надо чтобы последовательность из модулей членов ряда монотонно убывала. Найдем отношение модуля k+1-ого члена к модулю k-ого. Это будет ||. Переходя к пределу при k->oo, дробь будет стремиться к -1, и так как мы потребовали |x|<1, то отношение будет меньше 1. Значит последовательность монотонно убывающая, и ряд сходится. Тогда при больших k остаток ряда можно условно считать бесконечно убывающей геометрической прогрессией со знаменателем -x. Как оценить остаток ряда? Сумма прогрессии будет равна b1/(1-q) = b1/(1+x), где b1 это очередной член ряда. Это сопоставить с эпсилоном довольно сложно, однако мною было замечено, что если взять b1 примерно как eps^2, то погрешность становится незначительной. Тогда достаточно бежать в цикле, пока очередной член ряда по модулю больше eps^2

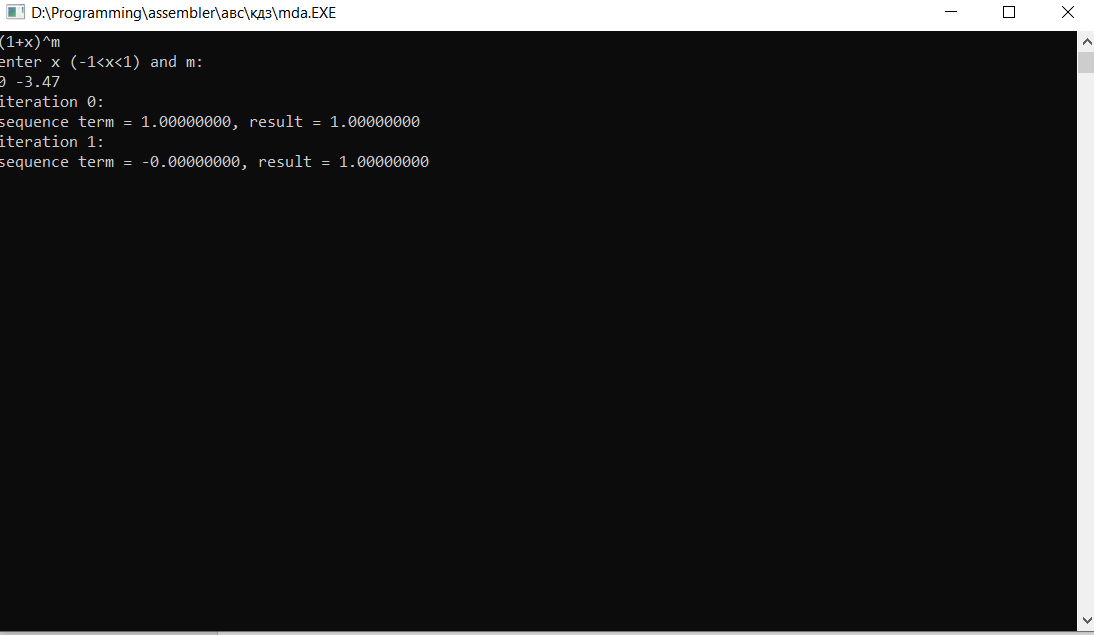
Тестовые примеры

Результаты сравнивались с результатами калькулятора на Android

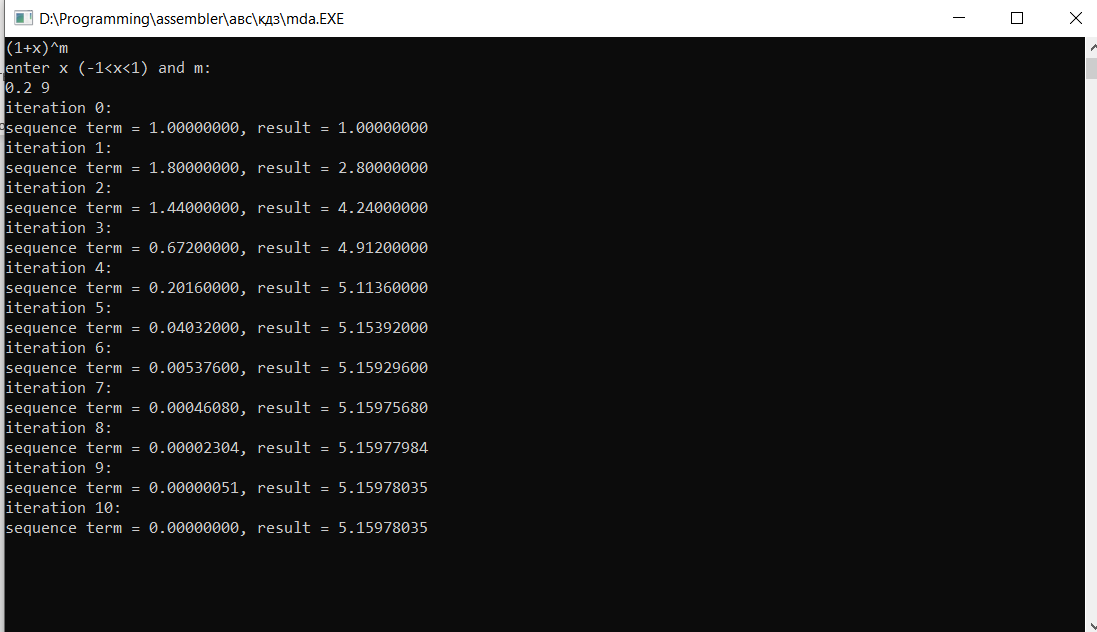
Результат калькулятора = 1



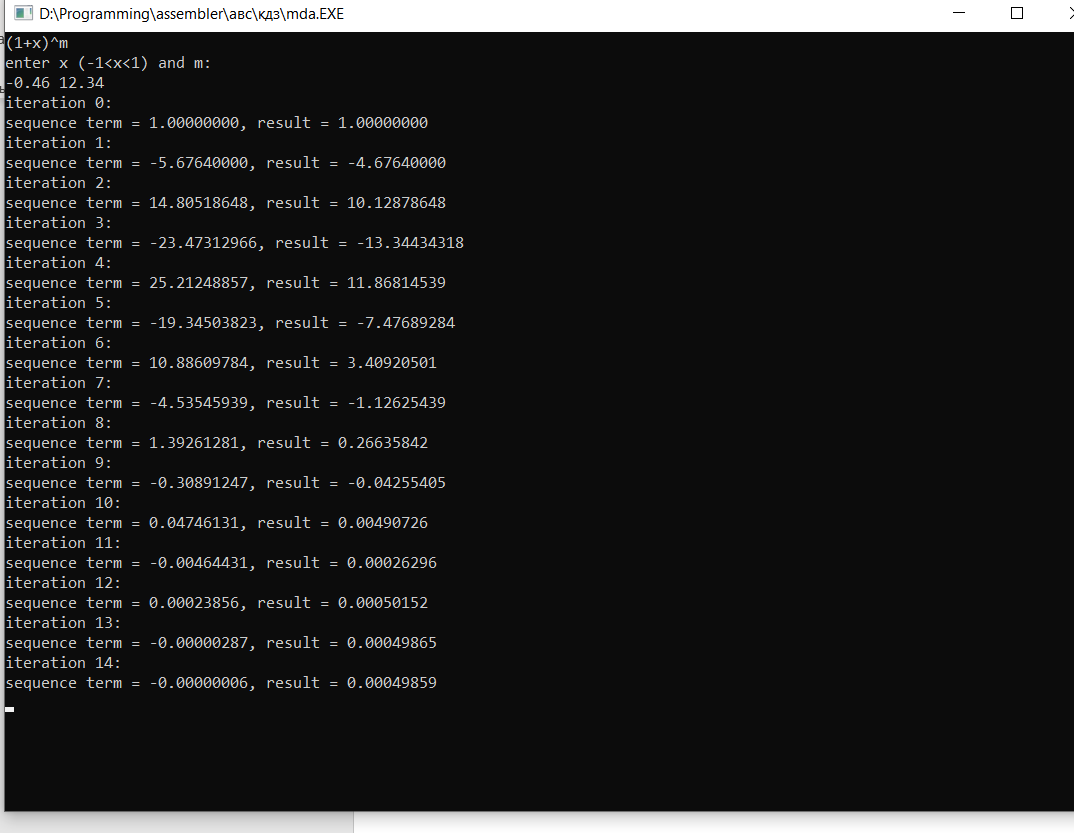
Результат калькулятора = 1



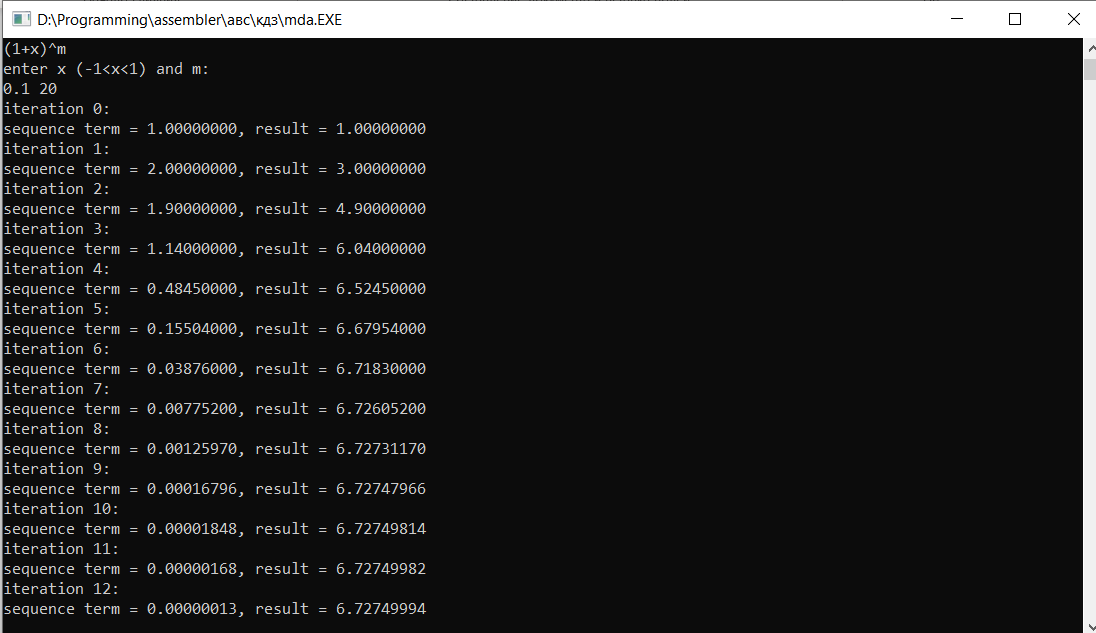
Результат калькулятора = 5,159780352



Результат калькулятора = 0,0004985847562…



Результат калькулятора = 6,7274999493256…



Список использованных источников

<http://natalia.appmat.ru/c&c++/assembler.html>

<http://flatassembler.narod.ru/fasm.htm#2-1-13>

<https://programmersforum.ru/showthread.php?t=227444&page=1>

<https://ru.wikipedia.org/wiki/%D0%A0%D1%8F%D0%B4_%D0%A2%D0%B5%D0%B9%D0%BB%D0%BE%D1%80%D0%B0#%D0%A0%D1%8F%D0%B4%D1%8B_%D0%9C%D0%B0%D0%BA%D0%BB%D0%BE%D1%80%D0%B5%D0%BD%D0%B0_%D0%BD%D0%B5%D0%BA%D0%BE%D1%82%D0%BE%D1%80%D1%8B%D1%85_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B9>

Приложения

Код программы:

format PE console

entry Start

include 'win32a.inc'

section '.data' data readable writable

enterStr db "(1+x)^m",10,"enter x (-1<x<1) and m:",10,0

ffStr db "%lf %lf",0

ffnlStr db "sequence term = %.8lf, result = %.8lf",10,0

inlStr db "iteration %d:",10,0

nl db 10,0

; номер итерации

k dd 0

; вводимый x

x dq ?

; вводимый m

m dq ?

; текущий член ряда

term dq ?

; текущая сумма ряда

sum dq ?

; константа 1

one dq 1.0

; погрешность

epsilon dq 0.0005

; квадрат погрешности

epsilonSq dq ?

section '.code' code readable executable

; макроинструкция для вывода чисел с плавающей запятой

macro PrintFloat str, [args] {

reverse

push dword [args+4]

push dword [args]

common

push str

call [printf]

}

Start:

; finit

; ввод x и m

cinvoke printf, enterStr

cinvoke scanf, ffStr, x, m

; term = 1, sum = 1

fld qword [one]

fst qword [term]

fstp qword [sum]

; epsilonSq = epsilon^2

fld qword [epsilon]

fmul qword [epsilon]

fstp qword [epsilonSq]

; главный цикл

MainLoop:

cmp [k], 1000000 ; максимум итераций

je PrintResult

; вывод текущих term и sum

cinvoke printf, inlStr, [k]

PrintFloat ffnlStr, term, sum

fld qword [m]

fisub dword [k] ; st0 = m-k

fild dword [k]

fadd qword [one] ; st0 = k+1

fdivp st1, st0 ; st0 = (m-k)/(k+1)

fmul qword [x] ; st0 = (m-k)/(k+1)\*x

fmul qword [term] ; st0 = st0\*term

fst qword [term] ; term = (m-k)/(k+1)\*x\*term

fadd qword [sum] ; st0 = st0+sum

fstp qword [sum] ; sum = sum+term

inc [k] ; k++

; если |term| <= epsilon^2 тогда break

; (term <= epsilon^2 && -term >= epsilon^2)

fld qword [epsilonSq] ; epsilon^2

fld qword [term]

fcomi st1

jnbe Last ; если term > epsilon^2

ffree st0 ; st0 = null

ffree st1 ; st1 = null

fld qword [epsilonSq]

fsub qword [epsilonSq]

fsub qword [epsilonSq] ; -epsilon^2

fld qword [term]

fcomi st1

jb Last ; или если -term < epsilon^2

ffree st0 ; st0 = null

ffree st1 ; st1 = null

jmp PrintResult ; break

Last:

ffree st0 ; st0 = null

ffree st1 ; st1 = null

jmp MainLoop

PrintResult:

; вывод результата

cinvoke printf, inlStr, [k]

PrintFloat ffnlStr, term, sum

Finish:

cinvoke getch

cinvoke ExitProcess, 0

section '.idata' import data readable

library kernel32, 'kernel32.dll',\

msvcrt, 'msvcrt.dll'

import kernel32,\

ExitProcess, 'ExitProcess',\

HeapCreate, 'HeapCreate',\

HeapAlloc, 'HeapAlloc',\

HeapFree, 'HeapFree'

import msvcrt,\

printf, 'printf',\

scanf, 'scanf',\

getch, '\_getch'