ЗАДАНИЕ № 6 ПО ПРАКТИКУМУ

для студентов 1 потока 1 курса ф-та ВМК МГУ
в 2015/2016 учебном году, весенний семестр

Тема: «Сборка многомодульных программ. Вычисление корней уравнений и определенных интегралов».
Языки программирования: Си, ассемблер NASM.

ПОСТАНОВКА ЗАДАЧИ

С заданной точностью ε вычислить площадь плоской фигуры, ограниченной тремя кривыми, уравнения которых $y = f_1(x)$, $y = f_2(x)$ и $y = f_3(x)$ либо заранее определены вариантом задания, либо задаются в текстовом виде на этапе сборки программы. Во втором случае необходимо разработать две программы: основную — для вычисления площади и вспомогательную — для построения исполняемого кода, вычисляющего значения функций.

При решении задачи необходимо следующее.

- С некоторой точностью ε_1 вычислить абсциссы точек пересечения кривых, используя предусмотренный вариантом задания метод приближенного решения уравнения $F(x) = 0$. В вариантах задания, где уравнения кривых фиксированы, отрезки, где программа будет искать точки пересечения, и где применим используемый метод, следует определить вручную. В вариантах задания, где уравнения задаются в текстовом виде, отрезки для поиска пересечения задаются вместе с уравнениями.
- Представить площадь заданной фигуры как алгебраическую сумму определенных интегралов и вычислить эти интегралы с некоторой точностью ε_2 по квадратурной формуле, предусмотренной вариантом задания.

Величины ε_1 и ε_2 подобрать вручную так, чтобы гарантировалось вычисление площади фигуры с точностью ε.

ВАРИАНТЫ ЗАДАНИЯ

Замечание: варианты, помеченные вертикальной чертой слева, имеют повышенную сложность.

I. Уравнения кривых

Варианты 1–10, обычной сложности. Точность вычислений ε = 0.001.
$y = f_i(x)$:

1)	$f_1 = 2^x + 1$,	$f_2 = x^5$,	$f_3 = (1 - x) / 3$
2)	$f_1 = 3\,(0.5 / (x + 1) + 1)$,	$f_2 = 2.5x - 9.5$,	$f_3 = 5 / x$ $(x > 0)$
3)	$f_1 = \exp(-x) + 3$,	$f_2 = 2x - 2$,	$f_3 = 1 / x$
4)	$f_1 = \exp(x) + 2$,	$f_2 = -1 / x$,	$f_3 = -2\,(x + 1) / 3$
5)	$f_1 = 0.35x^2 - 0.95x + 2.7$,	$f_2 = 3x + 1$,	$f_3 = 1 / (x + 2)$
6)	$f_1 = 0.6x + 3$,	$f_2 = (x - 2)^3 - 1$,	$f_3 = 3 / x$
7)	$f_1 = \ln(x)$,	$f_2 = -2x + 14$,	$f_3 = 1 / (2 - x) + 6$
8)	$f_1 = \exp(x) + 2$,	$f_2 = -2x + 8$,	$f_3 = -5 / x$
9)	$f_1 = 3 / ((x - 1)^2 + 1)$,	$f_2 = \mathrm{sqrt}(x + 0.5)$,	$f_3 = \exp(-x)$
10)	$f_1 = 1 + 4 / (x^2 + 1)$,	$f_2 = x^3$,	$f_3 = 2^{-x}$

Вариант 11, повышенной сложности. Точность вычислений ε = 0.001.

Плоская фигура задается во время сборки программы в виде текстового описания. Фигура ограничивается графиками трех функций, отрезок, на котором эти графики пересекаются, заранее определен и указан в этом же текстовом файле. Используется следующий формат текстового файла.

Первая строка содержит два вещественных числа, разделенные пробелами. Числа задают границы отрезка, на котором следует искать точки пересечения кривых. Следующие три строки описывают функции, используя для этого польскую обратную запись. Каждая строка содержит разделенные пробелами термы: переменная величина *x*, вещественное число, операция.

Переменная задается символом `'x'`. Вещественные числа задаются в формате, поддерживаемом функцией `scanf`. Также должны поддерживаться константные значения `'e'` и `"pi"`. Поддерживаемые операции:

- бинарные — `'+'`, `'-'`, `'*'`, `'/'`;
- унарные — `"sin"`, `"cos"`, `"tan"`, `"ctg"`.

Пример:

```
0.0 4.0
2 x 4 / tan -
x
0.2 pi *
```

Имя текстового файла не фиксируется и должно задаваться при сборке в виде переменной окружения:

```
$ SPEC_FILE=in.txt make
```

Для обработки текстового файла потребуется разработать и реализовать вспомогательную Си-программу, которая по заданному файлу с описанием строит ассемблерный листинг с функциями, вычисляющими заданные выражения. Для вычисления необходимо использовать команды сопроцессора x87. Для приведенного выше файла будет получен код, приведенный на следующей странице.

Польская запись считывается и по ней строится промежуточное представление: дерево, описывающее выражение над переменной *x* и вещественными числами. При необходимости вычисляются производные, которые также представляются в виде деревьев. Следует учитывать, что слишком высокие деревья не могут быть вычислены на стеке регистров x87 без предварительного преобразования, уменьшающего их высоту.

После того, как был получен ассемблерный листинг, начинается сборка основной программы, вычисляющей площадь фигуры. Ассемблерный листинг транслируется `nasm`, полученный объектный модуль компонуется с модулем, полученным от компилятора.

В `Makefile` должна быть описана полная процедура сборки, с учетом всех зависимостей: предварительного построения вспомогательной программы и генерации ассемблерного листинга.

При решении усложненного задания к показателям по контестам дополнительно прибавляются: две задачи в случае, если вычисление производных не требуется; три задачи, если производится вычисление производных.

```
section .data
   const1 dq 2.0
   const2 dq 4.0
   const3 dq 0.2

section .text
f1:
   push ebp
   mov  ebp, esp
   fld  qword [const1]
   fld  qword [ebp + 8]
   fld  qword [const2]
   fdivp
   fptan
   fxch
   fstp st1
   fsubp
   pop  ebp
   ret

f2:
   push ebp
   mov  ebp, esp
   fld  qword [ebp + 8]
   pop  ebp
   ret

f3:
   push ebp
   mov  ebp, esp
   fld  qword [const3]
   fldpi
   fmulp
   pop  ebp
   ret
```

II. Методы приближенного решения уравнений

1. Метод деления отрезка пополам.
2. Метод хорд (секущих).
3. Метод касательных (Ньютона).
4. Комбинированный метод (хорд и касательных).
5. Реализуется два различных метода решения уравнений. Для каждого метода строится исполняемый файл, выбор метода происходит на этапе сборки программы. Обе функции, реализующие методы, размещаются в общем файле, внутри директив условного препроцессирования. Ключ `-D` задает символ для препроцессирования, определение ключей задается в `Makefile`.

При решении усложненного задания к показателям по контестам дополнительно прибавляется одна задача.

III. Квадратурные формулы

1. Формула прямоугольников.
2. Формула трапеций.
3. Формула Симпсона (парабол).

Таким образом, вариант определяется следующими параметрами: (1) способом задания кривых и их набором для вариантов 1-10, (2) методом приближенного решения уравнений, (3) квадратурными формулами.

ТРЕБОВАНИЯ К ПРОГРАММЕ

1. Функции, вычисляющие значения f_1, f_2, f_3, и их производных (в случае необходимости производных) реализуются на языке ассемблера с соглашением вызова cdecl. Все остальные функции программы реализуются на языке Си.
2. Основная программа должна поддерживать опции командной строки, при задании которых:
   a) печатаются абсциссы точек пересечения кривых,
   b) печатается число итераций, потребовавшихся на приближенное решение уравнений при поиске точек пересечения.
3. Программа должна поддерживать ключ командной строки `-help`, выводящий на печать все допустимые ключи командной строки.
4. Вычисление с точностью ε_1 корня x уравнения $f(x) = g(x)$ на отрезке $[a, b]$ должно быть реализовано в отдельной Си-функции `root(f, g, a, b, eps1)`. Если используется метод касательных или комбинированный метод, то у `root` должно быть еще два параметра функционального типа, позволяющие вызывать производные функций f и g.
5. Вычисление с точностью ε_2 величины определенного интеграла от функции $f(x)$ на отрезке $[a, b]$ должно быть реализовано в отдельной Си-функции `integral(f, a, b, eps2)`.
6. Си-функции `root` и `integral` должны быть предварительно протестированы. Основная программа должна предоставлять возможности тестирования, активируемые опцией командной строки. Фактические параметры вызова тестируемых Си-функций задаются в командной строке; параметры функционального типа задаются по номеру соответствующей функции.
7. Сборка программы должна осуществляться при помощи утилиты make. Соответствующий файл должен явно или неявно описывать зависимости между всеми целями сборки. Должны быть определены цели *`all`* и *`clean`*, первая из которых полностью собирает программу, а вторая — удаляет все промежуточные файлы (в частности, объектные модули). Сдаваемый архив должен включать в себя `Makefile`.
8. Программа сдаётся в виде .zip-архива, содержащего в себе все необходимые файлы с исходным кодом, а также отчёт в формате PDF.
9. Программа должна быть снабжена поясняющими комментариями в объёме, достаточном для её понимания. Все глобальные и статические переменные должны быть документированы в комментариях.
10. Выбор конкретного метода решения уравнений должен управляться определёнными символами на этапе препроцессирования и передаваться через ключ `-D`.

ЛИТЕРАТУРА

1. Ильин В.А., Садовничий В.А., Сендов Бл.Х. Математический анализ. Т.1 — М.: Наука, 1985.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

1. Методические указания о численных методах и их реализации в программе приведены в методическом пособии «Задания практикума на ЭВМ» Трифонов Н.П., Пильщиков В.Н., задание 1.
2. Необходимо помнить о том, что внешние имена на платформе Windows снабжаются ведущим подчёркиванием. На этой платформе функцию `f1` следует в Ассемблерном коде называть `_f1`. Си-код при этом не меняется. К сдаче должен быть подготовлен вариант программы, предназначенный для запуска в UNIX-окружении.
3. Аналитическую работу с кривыми упрощают онлайн-сервисы, например, http://fooplot.com/