

l_2 . Согласно единственности поставленной краевой задачи [5] система (18) имеет единственное решение из l_2 и разрешима численно методом редукции [9].

Заряд на сферическом сегменте вычисляется по формуле $Q = 2\pi\epsilon a \left[(1 - \cos \theta_0) y_0 + \sum_{n=1}^{\infty} y_n (P_{n-1}(\cos \theta_0) - P_{n+1}(\cos \theta_0)) \right]$, ϵ — диэлектрическая проницаемость среды.

Произведен расчет конденсаторной емкости $K = Q/2\pi\epsilon a V$ данной физической системы с использованием табулированных интегралов B_n [10] на ЭВМ ЕС-1022. На рис. 2 показана зависимость K от параметра $\mu = 2a/\gamma_0^2$ при некоторых значениях угла раствора θ_0 сферического сегмента S . В частности, $K = 2,2515$ ($\mu = 0,5$, $\theta_0 = \pi/2$), $K = 3,7046$ ($\mu = 0,7$, $\theta_0 = 31\pi/32$), $K = 1,0008$ ($\mu = 0,9$, $\theta_0 = \pi/8$).

Заметим, что разработанная методика может быть применена, когда внутри бесконечного параболоида находится несколько тонких сферических сегментов. В этом случае используются теоремы сложения [11], связывающие сферические гармонические функции, отнесенные к центрам сегментов.

ЛИТЕРАТУРА

1. Уфлянд Я. С. Метод парных уравнений в задачах математической физики. — Л., 1977, с. 220.
2. Тучкин Ю. А., Шестопалов В. П. — Дифференц. уравнения, 1982, т. 28, № 4, с. 663.
3. Ерофеев В. Т. — ЖТФ, 1982, т. 52, № 3, с. 412.
4. Иоссель Ю. Я., Кочанов Э. С., Струнский М. Г. Расчет электрической емкости. — Л., 1981, с. 288.
5. Тихонов А. Н., Самарский А. А. Уравнения математической физики. — М., 1972, с. 735.
6. Градштейн И. С., Рыжик И. М. Таблицы интегралов, сумм, рядов и произведений. — М., 1962, с. 1100.
7. Ерофеев В. Т. — Весті АН БССР. Сер. фіз.-мат. наук, 1972, № 4, с. 42.
8. Виноградов С. С., Тучкин Ю. А., Шестопалов В. П. — Докл. АН СССР, 1978, т. 242, № 1, с. 80.
9. Канторович Л. В., Крылов В. И. Приближенные методы высшего анализа. — М. — Л., 1962, с. 708.
10. Ling C. B., Lin J. — Math. comput., 1972, v. 26, p. 529.
11. Caola M. J. — J. Phys., 1978, v. A11, № 2, p. 23.

Поступила в редакцию
08.09.82.

Кафедра уравнений
математической физики

УДК 681.3.06

Г. А. ДРОБУШЕВИЧ, П. В. ГЛЯКОВ,
И. В. КОМАРОВСКИЙ, П. П. КОЖИЧ

К ИТОГАМ ВСЕСОЮЗНОЙ СТУДЕНЧЕСКОЙ ОЛИМПИАДЫ ПО ПРОГРАММИРОВАНИЮ

По поручению Министерства высшего образования СССР и ЦК ВЛКСМ в конце декабря 1982 г. в БГУ имени В. И. Ленина была проведена Всесоюзная олимпиада «Студент и научно-технический прогресс» по ЭВМ и программированию. В олимпиаде приняли участие 14 команд союзных республик и г. Москвы. Каждая команда состояла из 4 участников и преподавателя — руководителя команды. Отличных успехов добились представители белорусской школы программирования — студенты факультета прикладной математики Белгосуниверситета имени В. И. Ленина. Николай Демеш, Галина Пудловская и Евгений Терехов заняли соответственно 1—3 места в личном зачете, а команда университета (все ФПИМ) уверенно заняла первое место. Большая заслуга в подготовке ее к олимпиаде руководителя команды — старшего преподавателя кафедры общего программирования И. И. Млынчика.

Жюри, состоявшее из преподавателей всех прибывших команд, выбрало следующие четыре задачи.

1. Разработать алгоритм, который из трех символов A , B и C формирует слово длины n такое, что никакие два соседних под слова не совпадают друг с другом.

2. Разработать алгоритм, который для двух натуральных k , $n (1 \leq k \leq n)$ находит и выдает в виде одномерных массивов все различные представления числа n суммой k натуральных чисел. Представления числа, отличающиеся только порядком слагаемых, считаются одинаковыми.

3. Дан одномерный массив целых неотрицательных чисел $a_1, \dots, a_n, a_{n+1}, \dots, a_{2n}, a_{2n+1}, \dots, a_{mn}$ (m, n — целые положительные числа). Получить одномерный массив $b_0, b_1, \dots, b_n$ и вывести его на печать, начиная с b_0 . Элементы $b_i (i=0, 1, \dots, n)$ определяются из соотношения:

$$\sum_{k=1}^m (x - a_{(k-1)n+1}) \dots (x - a_{kn}) = b_0 x^n + b_1 x^{n-1} + \dots + b_n.$$

Данные: $n \leq 9$, $m \leq 99$ отперфорированы в колонках 1—3 первой перфокарты, $a_i (0 \leq a_i \leq 9)$ — последовательно в колонках 1— n каждой из следующих m перфокарт.

4. Заданы последовательные файлы F_1 и F_2 , тип записи которых соответственно $Z_1 = \langle \text{ПРИЗНАК, ИНФ} \rangle$ и $Z_2 = \langle \text{ПРИЗНАК, ИНФ} \rangle$. Записи в файлах упорядочены по неубыванию значений признака, причем в пределах файла могут быть повторяющиеся записи (т. е. записи с одинаковым значением признака). Составить и запрограммировать алгоритм формирования выходного последовательного файла F_3 , $Z_3 = \langle \text{ПРИЗНАК, ИНФ} \rangle$, без повторяющихся записей. Для каждого k , являющегося значением признака одного из входных файлов, сформировать в F_3 запись с этим значением признака и значением компоненты ИНФ, равным сумме значений одноименных компонент входных файлов из всех записей со значением признака равным k . Программе для хранения промежуточных данных должно быть достаточно фиксированного объема памяти (оперативной и внешней).

Дополнительно были заданы следующие ограничения. Значения компонент ПРИЗНАК и ИНФ являются числовыми и занимают четыре байта памяти. Конкретный формат этого значения необходимо было участникам олимпиады выбрать самим. Длины записей файлов F_1, F_2, F_3 равнялись 28 байтам, т. е. часть записи с 9 по 28 байты на содержательном уровне в задаче не использовалась (в записи поля с 1 по 4 байт и с 5 по 8 байт содержат значения соответственно компонент ПРИЗНАК и ИНФ). Кроме того, требовалось написать поток управляющих операторов на трансляцию, редактирование и выполнение программы, т. е. надо было продемонстрировать по операционной системе минимум знаний, касающихся этапов прохождения задачи от написания исходного текста программы до окончательного ее выполнения. Длину блока надо было рассчитать, исходя из длины записи.

Типичной ошибкой в решениях задачи 1 было то, что при разработке алгоритма не предусматривалась необходимость обратного хода. Наиболее часто встречаемая ошибка в решениях задачи 2 заключалась в том, что при генерации представлений числа n формировались либо некоторые одинаковые представления, либо часть представлений пропущена.

Общие для решений первых двух задач ошибки заключались в следующем: при построении циклов использовались неверные начальные и конечные значения переменной цикла, при записи индексных переменных указывались неверные значения индексов, иногда выходящие за диапазон изменений, не исследовалось поведение алгоритма в экстремальных ситуациях (например, когда $n=1$).

Недостатком представленных решений по задачам 1, 2 следует считать то, что при разработке алгоритмов не нашли должного отражения вопросы выбора эффективного алгоритма, оценки его сложности и требуемой памяти, так как оценка сложности и памяти является одним из основных критериев эффективности алгоритма.

Положительными чертами решений задач 1, 2 является то, что некоторые студенты пытались доказать корректность предлагаемых алгоритмов, использовали рекурсивные методы программирования, применяли структурный подход при разработке алгоритмов.

Задача 3 была предложена с целью проверки умения использовать подпрограммы, выбирать конструкции языка, структуры данных и эффективный метод программирования. Задача допускает короткое и изящное решение.

Около 79 % представленных решений выполнялись на языке ПЛ/1; работ, оцененных выше двух баллов, на языке ПЛ/1 — 63, на языке Фортран — 33 %. Студенты, использовавшие язык ПЛ/1, получили более красивые решения: конструкции языка зачастую подсказывали ход решения. Из анализа следует вывод, что лучшие программисты пользовались языком ПЛ/1; вероятно, он формирует более совершенное алгоритмическое мышление, чем Фортран.

Анализ показал, что только в четырех работах (БССР, Азербайджан) приводилась оценка эффективности алгоритмов и программ.

В процессе тестирования программ на ЭВМ только девять участников получили правильные результаты.

В работах студентов МГУ проявился чисто математический подход к решению — сведение задачи к известным, несмотря на то, что трудоемкость последних больше трудоемкости исходной. Однако заслуживают внимания навыки использования реальных ЭВМ: программы студентов факультета вычислительной математики и кибернетики МГУ П. П. Зелинского и А. М. Панова заняли соответственно 2-е и 9-е места в специальном конкурсе на ЭВМ.

Для работ студентов вузов Москвы, а также технических вузов республик характерно стремление получить результат, не взирая на средства: выбирается очевидный, известный алгоритм и для его реализации привлекаются мощные средства языков (рекурсия, встроенные функции), хотя результат можно получить более простыми средствами. Машинный эксперимент показал, что эти программы не являются лучшими по быстродействию, памяти и возможностям. Однако, с точки зрения полноты знания возможностей языка ПЛ/1, работы студентов вузов Москвы оставляют приятное впечатление.

В качестве примера удачного решения можно привести программу студентки факультета прикладной математики БГУ имени В. И. Ленина Г. М. Пудловской (см. рисунок). По результатам машинного эксперимента эта программа оказалась лучшей по всем характеристикам: времени счета, памяти, скорости получения результата (время от формулировки условия до получения результата). Она показала лучшее время даже по сравнению с Фортран-программами, которые традиционно считаются более быстродействующими (использовались стандартные компиляторы уровня F ОС ЕС версии 4.0).

В настоящее время качество работы программиста оценивается не только по тому, что сделано, но и как делалось. Вопросы технологии программирования, документирования, умения создавать не программы, а программные изделия, очень актуальны. Как показали работы участников, в большинстве вузов этим вопросам уделяется мало внимания. Хорошая программа почти всегда результат нисходящего проектирования, однако представленные блок-схемы (9 работ) имели тот же уровень детализации, что и сама программа. Только в 10 работах применялись иерархические структурные описания в виде блок-схем, псевдокода. В подавляющем большинстве работ студенты ограничивались нестрогим словесным описанием.

```

POL: PROC OPTIONS (MAIN);
  DCL (M, N) BIN FIXED (31);
  GET EDIT (N, M) (F (1), F (2));
  V:BEGIN;
    DCL A(N) BIN FIXED (31);
    DCL (B(N + 1), Q(N + 1)) BIN FIXED (31);
    B = 0;
    IOJ = 0;
    ON ENDFILE (SYSIN) IOJ = 1;
    GET SKIP EDIT (A) (9 F (1));
    DO WHILE (IOJ = 0);
      Q = 0; Q (1) = 1;
      DO I = 1 TO N; /* — ФОРМИРУЕТ ПОЛИНОМ
        Q (X) = (X — A ((K — 1)*N + 1))* . . . *(X — A (K*N))*/
        DO J = I + 1 TO 2 BY — 1; /* УМНОЖАЕМ
          Q = Q* (X — A ((K — 1) * N + I))*/
          Q (J) = Q (J — 1) — Q (J) * A (I);
        END;
        Q (1) = — Q (1) * A (I);
      END;
      B = B + Q;
      GET SKIP EDIT (A) (9 F (1));
    END;
    K = 0;
    DO I = N + 1 TO 1 BY — 1;
      PUT EDIT ('B', K, B (I)) (SKIP, A, F (2), X (5), F (10));
      K = K + 1;
    END;
  END V;
END POL;

```

Наличие комментариев в тексте программы отмечено в 10 работах. В целом на олимпиаде участники показали высокий уровень знания языков программирования. Заслуживают внимания использование рекурсии, встроенных функций ПЛ/1.

Типичные ошибки: а) ввод-вывод — неумение записать список форматов для фиксированного числового материала (обычно студенты сами выбирают формат числового материала, а обработка заданного формата вызвала затруднения); неумение выбрать спецификацию вывода для чисел; б) заикливание; в) использование неподходящих типов переменных; г) идентификаторы, задающие границы индексов массива, активизируются в том же блоке, что и сам массив; д) несоответствие атрибутов фактических и формальных параметров.

Типичны в решениях задачи 4 ошибки в программировании обработки ситуаций, возникающих, когда завершен просмотр некоторых файлов или когда они первоначально были пустыми. Определенные трудности вызвала организация вычислительных процессов продвижения по файлам и суммирования значений из групп смежных записей. Практически отсутствовало описание процесса проектирования в виде последовательности все более детализированных схем-программ на псевдокоде.

При написании потока управляющих операторов задания в управляющих операторах DD необходимо было указать длину блока, которую

требовалось рассчитать, исходя из заданной длины записи. При использовании магнитных дисков важно правильно и рационально определить длину блока с тем, чтобы объем памяти, занимаемый набором данных, и время последовательного его просмотра оказались по возможности минимальными. Однако подавляющее большинство участников при выборе длины блока не учитывало этих критериев. Как правило, либо вообще не осуществлялось блокирование записей, либо коэффициент блокирования выбирался произвольно, что приводило к неполному заполнению дорожки. В остальном большинство участников с этим заданием справилось успешно.

Следует отметить, что участники олимпиады показали хороший уровень знаний по использованию языков программирования и операционной системы ЕС и значительно более низкий уровень в области разработки алгоритмов и программ, а также в вопросах оценки их эффективности. В качестве одного из возможных путей устранения этого недостатка можно предложить метод надъязыкового обучения программированию и использование современных технологий программирования. Первоначально вводятся понятия высокого уровня (абстрактные структуры данных, абстрактные операции над ними), а для записи программ используется псевдокод, отрабатываются основные методы логической обработки структур данных. Затем выполняется конкретизация понятий и детализация процесса решения задачи, т. е. осуществляется переход на следующий уровень, уровень операторов и структур данных алгоритмических языков. И, наконец, последний шаг (если он необходим) — рассмотрение представления данных на физическом уровне.