

**МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО
СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ РСФСР**

**КУЙБЫШЕВСКИЙ ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ
АВИАЦИОННЫЙ ИНСТИТУТ имени академика С. П. КОРОЛЕВА**

Г Р А Ф О Р:
**КОМПЛЕКС ГРАФИЧЕСКИХ ПРОГРАММ
НА ФОРТРАНЕ**
(графические элементы)

**Утверждено
редакционно-издательским
советом института
в качестве
методических указаний
к практическим занятиям
для студентов**

В методических указаниях приведены сведения об основных графических программах пакета ГРАФОР и даны методические рекомендации по их использованию. Указания адресованы студентам специальности 0647.

Составители: Александров О.П., Суханов С.В.

Рецензенты: к.т.н. доц. А.В. Соловов,
к.т.н. доц. В.Б. Гумсников

ВОЗВРАТИТЕ КНИГУ НЕ ПОЗЖЕ

обозначенного здесь срока

Пакет программы ГРАФОР представляет собой набор подпрограмм, написанных на ФОРТРАНе, за исключением программы связи с операционной системой и графическим устройством, написанной на АССЕМБЛЕРЕ. ГРАФОР является удобным в обращении и легко транспортируемым с одного типа ЭВМ на другой математическим обеспечением машинной графики, в основном предназначенным для использования в научных расчетах.

Структура пакета ГРАФОР подобна перевернутой пирамиде, в острие которой находится программа связи с операционной системой ЭВМ и графическим устройством (интерфейс графического устройства). Следовательно, для перехода на работу с новым типом устройства или новой версией операционной системы достаточно сменить только одну программу и пакет будет готов к работе.

На следующем, более высоком уровне, находятся программы, реализующие графические утилиты: перевод пера в указанную точку, вычерчивание вектора, дуги окружности или эллипса, произвольного текста, различных маркеров и т.д. На программах графических утилит базируется второй уровень программ пакета, предназначенный для отображения плоских изображений. К программам второго уровня относятся также программы, как аффинные преобразования на плоскости, разметки числовых осей в декартовых, полярных и логарифмических координатах, проведение полигональных кривых, штриховка и окрашивание плоской области и ряд других программ.

К третьему уровню относятся программы, реализующие различные алгоритмы проекционной машинной графики: программы аффинных преобразований в трехмерном пространстве, выбора аппарата проецирования, вычерчивания пространственных полигональных кривых, отображения каркаса поверхности однозначной функции двух переменных как с удалением, так и без удаления незримых на изображении линий. В состав программ третьего уровня входят также программы вычерчивания изолинии функции двух переменных, заданной в углах прямоугольной регулярной сетки.

Программы, входящие в пакет, охватывают практически все области его применения в научных расчетах. Однако можно отметить, что в пакете не реализованы многие интересные алгоритмы и программы машинной графики. Так, программы вычерчивания каркаса поверхностей и изолиний функции двух переменных работают только с функциями, заданными в узлах прямоугольной сетки. Второе ограничение на функции требует их однозначности. В пакете отсутствуют программы, при помощи

которые можно было бы осуществить построение проекции фигуры не только на расположенную произвольным образом к проектируемому вектору картинную плоскость, но и на любую картинную поверхность. Такие отсутствовали программы получения изображения многогранников с удалением невидимых линий. Тем не менее, отсутствие указанных программ не снижает качества пакета.

Можно считать, что совокупность обработок к программам ГРАФОР, образует язык, позволяющий описать картинку, которая в конечном итоге и зафиксирована на бумаге. Это описание должно подчиняться определенным для языка синтаксическим правилам и семантическим ограничениям. При трансляции программы, написанных на языках высокого уровня (таких как АЛГОЛ, ФОРТРАН, ПЛ/Д), компилятор обнаруживает все синтаксические и семантические ошибки и может сопровождать листинг диагностическими сообщениями. В ГРАФОРе такая возможность отсутствует, поскольку операторы языка интерпретируются во время работы программы и для во время записи программы на магнитную или перфолену в случае использования автономного графопостроителя). Поэтому каждая программа комплекса анализирует обстановку, в которой придется работать, и в случае ошибки выбирается разумное продолжение, которое может привести к частичной или полной потере отдельного графического элемента, но ни в каком случае не повлияет на работу основной программы. Следует, однако, отметить, что, если количество фактических параметров отличается от требуемого, последствия трудно предсказать. Но это не является дефектом ГРАФОР, а специфично для трансляторов ФОРТРАНА.

Таким образом, при использовании комплекса ГРАФОР, программа вывода требуемой картинки на некоторое графическое устройство предоставляет собой программу на ФОРТРАНе, содержащую помимо списка переменных присвоения значений константам, операторы ввода-вывода данных и, если необходимо, прочих операторов ФОРТРАНА, обрабатываемые к программам ГРАФОР. Вывод графических программ производится стандартным образом:

CALL < имя подпрограммы (параметры) >

При описании параметров программ для краткости будем расшифровывать в большинстве случаев лишь назначение каждого из параметров для выполнения конкретной задачи машинной графики (т.е. семантический аспект).

Лишь в тех случаях, когда это имеет принципиальное значение, указем синтаксические данные параметра как элемента языка ФОРТРАН (ни-

пример, является ли этот параметр текстовой константой или именем действительной переменной). В остальных же случаях, если это не оговорено особо, параметры в соответствии с известным принципом умолчания (по первой букве имени) являются или именами действительных и целых переменных или, соответственно, действительными и целыми константами. Пользователю необходимо внимательно следить за тем, чтобы целым и действительным формальным параметрам программы ГРАФОРА в его программе строго соответствовали целые и действительные переменные или целые и действительные константы (в последнем случае не надо забывать ставить десятичную точку, что является частой ошибкой студентов). Если формальный параметр является текстовой константой, в обращении к программе на месте этого параметра пользователь должен указать также текстовую константу (две формы представления текстовых констант в программе рассмотрены при описании программы *PAGE*).

Та часть общей программы на ФОРТРАНе, которая помимо прочих операторов содержит обращение к программам ГРАФОРА, должна начинаться и закончиваться обращениями к программам общей организации. Первые три программы этого раздела комплекса позволяют выбрать единицы измерения (миллиметры, сантиметры, дюймы), а которых на чертеже заданы координаты и размеры всех графических элементов. Нужная единица измерения задается обращением к программе без параметра: *CALL MMS* - миллиметр, *CALL CMS* - сантиметр, *CALL INCHES* - дюйм.

Две другие программы общей организации связаны с понятием страницы, имеющем в ГРАФОРЕ принципиальное значение. Страницей называется прямоугольное поле на бумаге (экране дисплея или другом физическом носителе изображения), в пределах которого должны размещаться все элементы выводимой картинки. С помощью страницы программа выделяет выводимый чертёк из последовательности других изображений (отделяет чертежи одной задачи машинной графикой и чертежи разных задач). Определяя (открывая) страницу, пользователь вводит систему координат, относительно которой описывается выводимое изображение (в которой задаются координаты всех графических элементов). Перо может быть выведено за пределы страницы только после того, как страница закрыта. Поэтому, если координаты графического объекта заданы неправильно, объект либо отсутствует целиком, либо будет изображена лишь его часть, находящаяся в пределах страницы. Границы страницы могут быть очерчены, но даже буду-

чи не очерченными, они незримо присутствуют, не позволяя перу выйти за пределы страницы. Это особенно важно при работе неотлаженных программ, поскольку локализация пера внутри страницы позволяет сохранить в неприкосновенности выведенные ранее чертежи.

Для того чтобы определить (открыть) страницу, пользователь должен обратиться к программе *PAGE* :

CALL PAGE (XL, YL, NAME, N, Y)

Параметры: *XL, YL* - размеры страницы (в предварительно выбранных единицах измерения) по осям *X* и *Y* соответственно. Предельные размеры определяются физическими размерами рабочего поля того графического устройства, которое используется для вывода данного изображения;

NAME - текстовая константа, с помощью которой пользователь идентифицирует страницу (дает данной странице название). Текстовая константа в ФОРТРАНе может быть задана в двух формах:

1. *'RTSUNOK □ 1'* - текст в апострофах;
2. *9H RTSUNOK □ 1* - число символов в константе, *H* - текст;
N - количество символов в названии страницы; если *J = 0*, то границы страницы не вычерчиваются, если *J = 1*, то границы вычерчиваются.*

Для того чтобы закрыть доступ к текущей странице и подготовиться к рисованию следующей (закрыть страницу), пользователь должен обратиться к программе *ENDPG* :

CALL ENDPG (NUMB)

Параметр *NUMB* - текстовая константа, занимающая одно машинное слово (4 символа); выписывается вдоль линии разреза. Авторы ГРАФОРА рекомендуют использовать ее как четырехзначное десятичное число для нумерации страниц. Например:

4H0013 или *'0013'*

При работе с описанными программами общей организации пользователь должен учитывать следующее:

* Авторы рекомендуют не вычерчивать границы страницы, если в этом нет необходимости.

1. Определение страницы с помощью программы *PAGE* обязательно и должно предшествовать всем другим обращениям к программам ГРАФОРА, кроме программы, определяющих исходное состояние (задания единиц измерения, выбор шрифта и наклона букв), т.е. если страница не определена, то любые обращения к программам ГРАФОРА игнорируются. Пользователь до определения страницы может выбрать единицы измерения, выбрать шрифт и задать наклон символов, которыми будет написано название страницы.

2. При определении страницы, если это необходимо, вычерчивается ее граница; в левом нижнем углу, вдоль оси *Y* пишется ее название (текстовая константа *NAME*), вдоль оси *X* выписываются шифр задачи и дата (рис. 1).

3. Левый нижний угол страницы имеет координаты (0, 0), а правый верхний - (X_L , Y_L), где X_L и Y_L - размеры сторон страницы.

4. По окончании вывода на текущую страницу ее необходимо закрыть, лишь после этого можно определить следующую, новую страницу. Таким образом, обращение к программе *ENDPG* обязательно. При выполнении этой программы перо выводится за пределы страницы, проводится линия раздела между страницами справа, параллельно оси *Y* на расстоянии 2 см от правой границы страницы, затем перо ставится на продолжение оси *X* (справа внизу) на 2 см правее линии раздела (см. рис. 1).

5. Задавать и менять единицы измерения необходимо до определения страницы. Внутри страницы этого сделать нельзя, так как после определения страницы обращения к программам *CMS*, *MMS*, *INCHES* игнорируются. Выбранная единица измерения действительна только в пределах одной страницы. После того как страница закрыта, можно менять единицы измерения. При закрытии страницы восстанавливается исходное состояние, при котором единицей измерения считается сантиметр. Таким образом, если необходимо задавать размеры в сантиметрах, то обращение к программе *CMS* можно опустить (т.е. по умолчанию координаты и размеры, указанные в обращениях к программам ГРАФОРА, считаются заданными в сантиметрах). Выбирать шрифт и задавать наклон символов можно в любом месте программы (до и после программ *PAGE* и *END PG*). По умолчанию используются прописные латинские и русские буквы без наклона (см. описание программы вычерчивания символов).

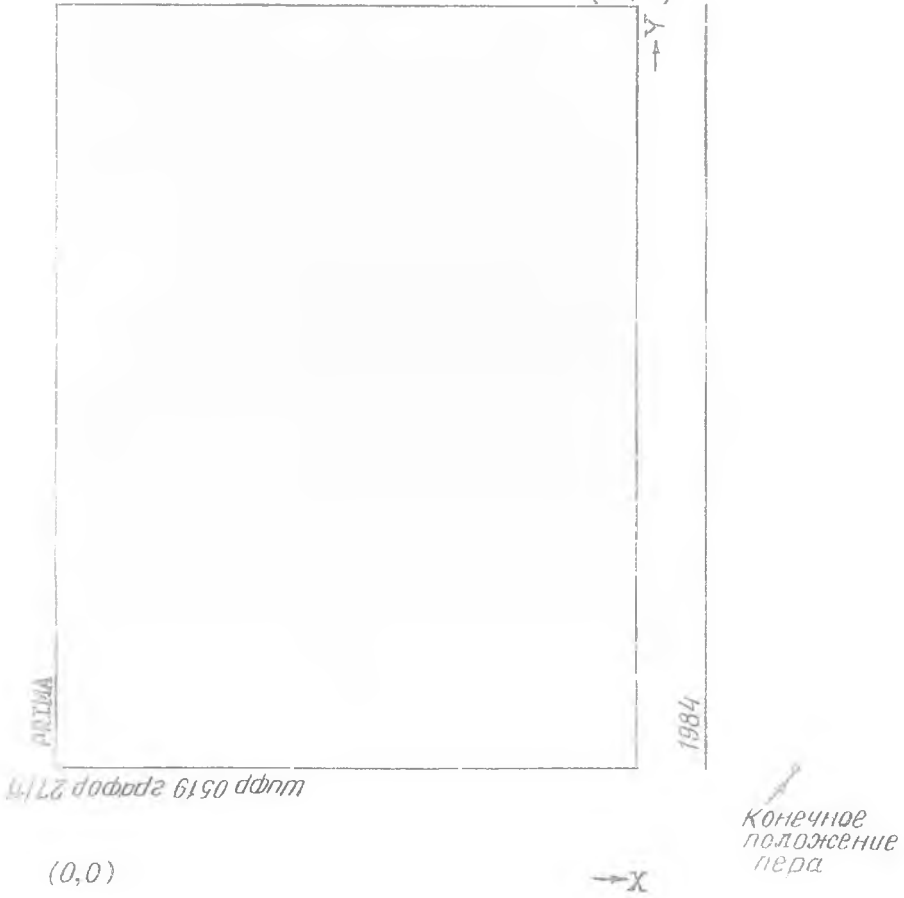
ТЕКСТ:

CALL PAGE (12., 20., 5HPRIMA, 5, 1)

CALL ENDPG (441984)

РЕЗУЛЬТАТ:

(XL, YL)



Р и с. 1. Пример использования программ определения и закрытия страницы

ПРОГРАММЫ ВЫЧЕРЧИВАНИЯ ГРАФИЧЕСКИХ ЭЛЕМЕНТОВ

Большинство программ вычерчивания графических элементов существенно используют в своей работе точку на странице, в которой непосредственно перед началом вычерчивания находится перо графического устройства (условимся далее для краткости использовать термины "текущая точка" или "текущее положение пера"). Поэтому, если необходимо вычерчивать графический элемент из заданной начальной точки, надо предварительно переместить перо в эту точку (например, с помощью программы *MOVE*). Будем считать, что текущая точка имеет координаты X_0 , Y_0 .

Кроме того, надо помнить следующее: если скажем, что угол наклона φ есть угол наклона некоторого отрезка к оси X , это означает, что угол φ отсчитывается от положительного направления оси X до данного отрезка в положительном математическом направлении, т.е. против часовой стрелки.

Программы вычерчивания отрезков прямой

Работая с программами вычерчивания отрезков, пользователю необходимо помнить, что, если точка, в которую должно переместиться перо графического устройства, находится за пределами страницы, значение координаты, превышающее допустимое, заменяется граничным. Прямая проводится (перо перемещается) от текущей точки к точке с модифицированными координатами, вследствие чего картина на границе может быть искажена (рис. 2,а).

Программа *MOVA*. Проводит отрезок прямой (перемещает поднятое перо) из текущей точки под заданным углом к оси X на заданную длину (рис. 2,в).

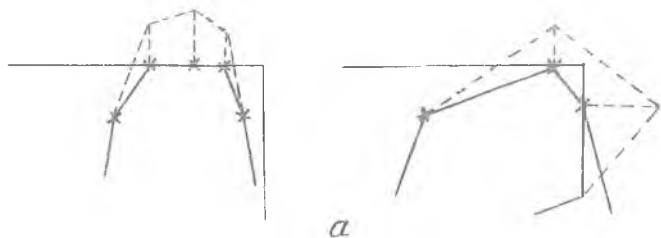
Обращение *CALL MOVA(DL, TH, J)*.

Параметры: *DL* - длина отрезка;

TH - угол наклона отрезка к положительному направлению оси X (в градусах);

J - см. описание программы *MOVE*.

Программа *MOVE*. Проводит отрезок прямой (перемещает поднятое перо) из текущей точки в точку с заданными координатами (рис. 2,б).



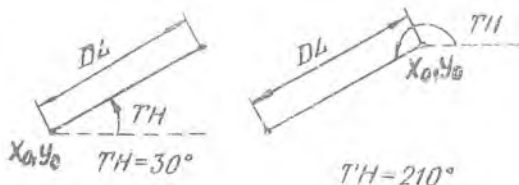
a

CALL MOVE(X,Y,J)



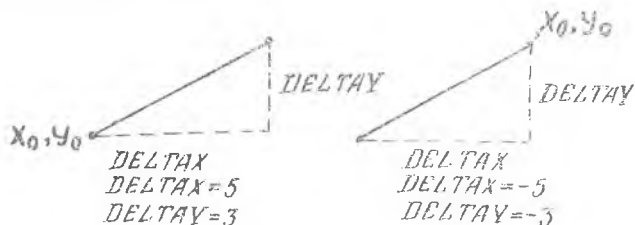
b

CALL MOVA(DL, TH, J)



b

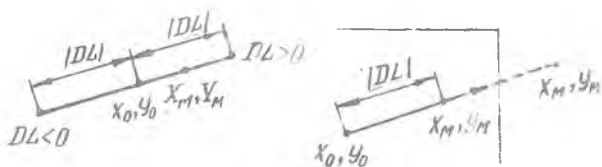
CALL MOVB(DELTA X, DELTA Y, J)



c

Р и с. 2. Программы вычерчивания

CALL MOVG(XM, YM, DL, J)



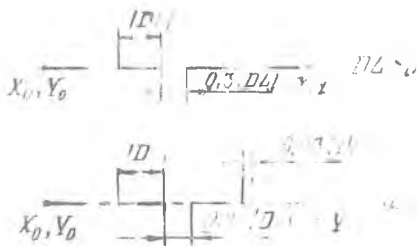
д

CALL FATLIN(XF, YF, D)



е

CALL DASHP(X, Y, DL)



жс

отрезков прямых

Обращение *CALL MOVE(X, Y, J)*.

Параметры: X, Y - координаты конца отрезка;

J - признак вычерчивания линии:

$J = 0$ - перемещение без рисования (перо поднято),

$J = 1$ - вычерчивание отрезка (перо опущено).

Программа *MOVVB*. Проводит отрезок прямой (перемещает поднятое перо) из текущей точки в точку, заданную приращениями координат (рис. 2,г).

Обращение *CALL MOVVB(DELTA X, DELTA Y, J)*.

Параметры: *DELTA X, DELTA Y* - приращения координат по осям X и Y соответственно;

J - см. описание программы *MOVE*.

Программа *MOVVC*. Проводит отрезок (перемещает поднятое перо) из текущей точки на заданное расстояние по прямой, проходящей через заданную точку (рис. 2,д).

Обращение *CALL MOVVC(XM, YM, DL, J)*.

Параметры: XM, YM - координаты точки, лежащей на прямой, по которой происходит перемещение пера;

$|DL|$ - расстояние, на которое перемещается перо (длина отрезка);

J - см. описание программы *MOVE*.

Примечание. Если $DL > 0$, перемещение пера по прямой происходит в сторону точки с координатами XM, YM , а если $DL < 0$, перемещение пера по прямой происходит в противоположную сторону. Сама точка с координатами XM, YM может находиться не только за пределами вычерчиваемого отрезка, но и за пределами страницы.

Программа *FATLIN*. Проводит отрезок заданной толщины из текущей точки в точку с заданными координатами (рис. 2,е).

Обращение *CALL FATLIN(XF, YF, D)*.

Параметры: XF, YF - координаты конца отрезка;

D - толщина линии в миллиметрах.

Примечание. Отрезок вычерчивается трехкратным проведением пера (одна линия в середине и две симметрично по краям на расстоянии D мм друг от друга), поэтому при больших D отрезок может выглядеть как три отдельные тонкие линии.

Программа DASHP. Проводит отрезок пунктирной или штрихпунктирной линии из текущей точки в точку с заданными координатами (рис. 2, ж).

Обращение *CALL DASHP(X, Y, DL)*.

Параметры: X, Y - координаты конца отрезка;

$|DL|$ - длина основного штриха линии.

Примечания: 1. Если $DL > 0$, то проводится пунктирная линия, а если $DL < 0$ - штрихпунктирная линия.

2. В пунктирной линии расстояние между штрихами $0,3 \cdot |DL|$.

3. В штрихпунктирной линии длина промежутка между штрихами $0,23 \cdot |DL|$ и в середине этого промежутка изображается дополнительный штрих (вместо точки) размером $0,03 \cdot |DL|$.

Программы вычерчивания окружностей, спиралей и их дуг

Программа CIRC. Вычерчивает окружность заданного радиуса с центром в текущей точке (рис. 3, а).

Обращение *CALL CIRC(R)*.

Параметр R - радиус окружности в заданных единицах измерения.

Примечание. После вычерчивания окружности перо возвращается в исходную точку с координатами $X_0 Y_0$.

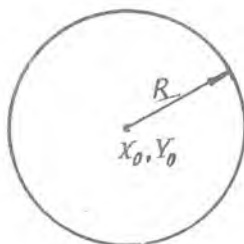
Программа ARCIA. Из текущей точки проводит дугу окружности заданного радиуса с указанными углами наклона к оси X , начального и конечного радиусов и заданным направлением рисования относительно центра окружности (рис. 3, б).

Обращение *CALL ARCIA(R, THD, THF)*.

Параметры: $|R|$ - радиус окружности;

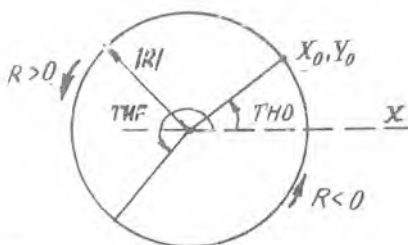
THD, THF - углы наклона соответственно начального и конечного радиусов к положительному направлению оси X ; задаются в градусах.

CALL CIRC(R)



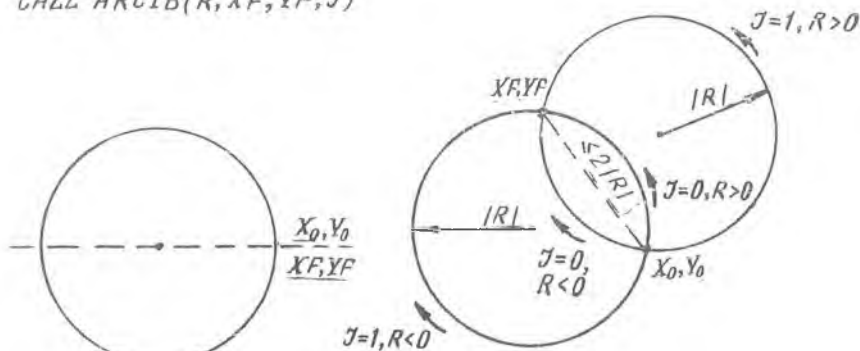
a

CALL ARCTA(R, THO, THF)



b

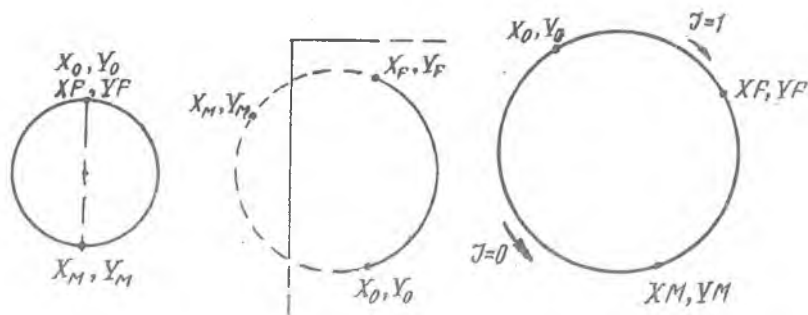
CALL ARCIB(R, XF, YF, J)



б

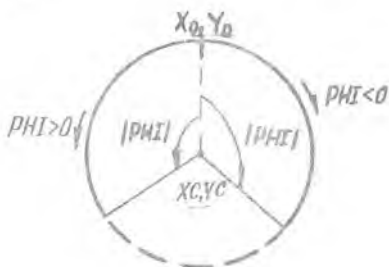
Р и с. 3. Программы вычерчивания окружностей,

CALL ARCIC(XM, YM, XF, YF, J)



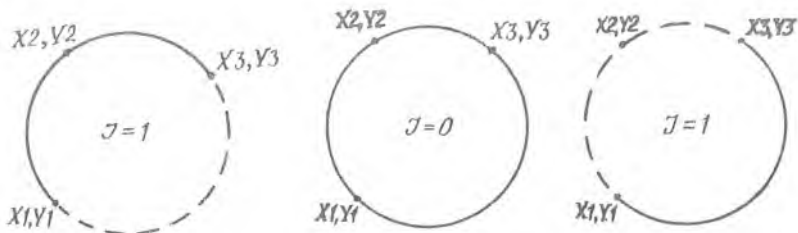
2

CALL ARCID(XC, YC, PHI)



3

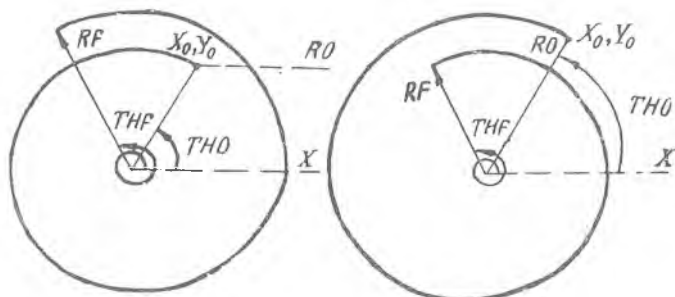
CALL ARC(X1, Y1, X2, Y2, X3, Y3, J)



4

спиралей и их дуг

CALL SPIRAL (THO, THF, RO, RF)

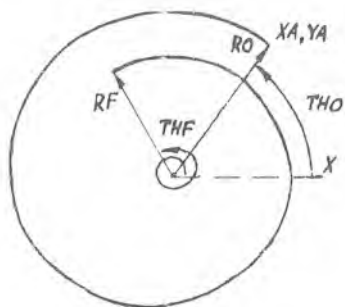


$RF > RO$

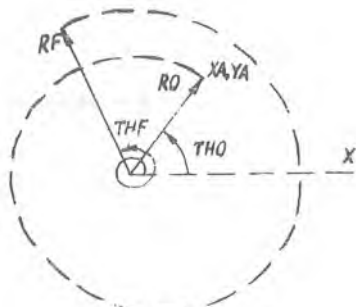
$RF < RO$

ЖС

CALL CIRCLE (XA, YA, THO, THF, RO, RF, L)



$L = 1$
 $RF < RO$



$L = 0$
 $RF > RO$

Ж

Продолжение рис. 3. Программы вычерчивания окружностей, спиралей и их дуг

Примечания: 1. Если $R > 0$, перо движется из начальной точки в конечную против часовой стрелки, а если $R < 0$, то по часовой стрелке.

2. Если $|THO - THE| > 360^{\circ}$, то обращение к программе считается некорректным, хотя геометрически построение такой дуги возможно.

Программа *ARCIB*. Из текущей точки проводит в заданную точку дугу окружности заданного радиуса с заданным направлением рисования относительно центра окружности (рис. 3, в).

Обращение *CALL ARCIB(R, XF, YF, J)*.

Параметры: $|R|$ - радиус окружности;

XF, YF - координаты конечной точки дуги;

J - признак расположения центра окружности:

$J = 0$ - проводится дуга, меньше или равная 180° ,

$J = 1$ - проводится дуга, больше или равная 180° .

Примечания: 1. Если $R > 0$, то перо движется из начальной точки в конечную против часовой стрелки, а если $R < 0$, то по часовой стрелке.

2. Обращение к программе считается некорректным, если расстояние между начальной и конечной точками превышает заданный диаметр.

$$(XF - XO)^2 + (YF - YO)^2 > 4R^2.$$

3. Если конечная точка совпадает с начальной, то, независимо от знака радиуса, вычерчивается полная окружность с углами наклона начального и конечного радиусов к положительному направлению оси X , соответственно 0 и 360° градусов.

Программа *ARCIC*. Из текущей точки проводит в заданную точку дугу окружности, проходящей через дополнительно указанную точку, т.е. дугу окружности, заданной тремя точками (рис. 3, г).

Обращение *CALL ARCIC(XM, YM, XF, YF, J)*.

Параметры: XM, YM - координаты дополнительной точки окружности;

XF, YF - координаты конечной точки дуги;

J - признак принадлежности дополнительной точки:

$J = 0$ - дуга проходит через дополнительную точку,

$J = 1$ - дуга не проходит через дополнительную точку.

Примечания: 1. Обращение к программе считается некорректным, если все три точки находятся на одной прямой. Когда же начальная точка совпадает с конечной, вычерчивается полная окружность с диаметром, равным расстоянию от начальной до дополнительной точки.

2. Если же дуга не проходит через дополнительную точку, последняя может находиться и вне страницы.

Программа ARCID. Из текущей точки проводит дугу окружности заданной угловой величины для указанного положения центра окружности (рис. 3, д).

Обращение *CALL ARCID* (*XC, YC, PHI*).

Параметры: *XC, YC* — координаты центра окружности;

|PHI| — угловая величина дуги (в градусах).

Примечания: 1. Обращение к программе считается некорректным, если значение угла *PHI* по модулю больше 360° : $|PHI| > 360^\circ$.

2. Если $PHI > 0$, дуга вычерчивается из текущей точки против часовой стрелки, если $PHI < 0$, дуга вычерчивается из текущей точки по часовой стрелке.

Программа ARC. Проводит дугу окружности по трем заданным точкам (рис. 3, а).

Обращение *CALL ARC* (*X1, Y1, X2, Y2, X3, Y3, J*).

Параметры:

$\left. \begin{array}{l} X1, Y1 \\ X2, Y2 \\ X3, Y3 \end{array} \right\}$ — координаты трех точек, задающих дугу;

J — признак вычерчивания дуги: $J = 1$ — дугой окружности соединяются три заданные точки; $J = 0$ — через три заданные точки проводится полная окружность; $J = -1$ — вычерчивается дуга, дополнительная к той, которая строится в случае $J = 1$.

Программа SPIRAL. Проводит спираль из текущей точки (рис. 3, в).

Обращение *CALL SPIRAL* (*THO, THF, RO, RF*).

Параметры: *THO, THF* — углы наклона соответственно начального и конечного радиусов к положительному направлению ось *X*; задаются в градусах;

RO, RF — соответственно начальный и конечный радиусы.

Примечание. Если $RF > RO$, рисуется раскручивающаяся спираль, а если $RF < RO$, то - сгущивающаяся. Если $RF = RO$, то рисуется дуга окружности. Вычерчивание производят против часовой стрелки.

Программа CIRCLE. Из заданной точки проводит окружности, спирали, дуги окружностей и дуги спиралей сплошной и штриховой линией (рис. 3,а).

Обращение *CALL CIRCLE(XA, YA, THO, THF, RO, RF, L)*.

Параметры: XA, YA - координаты точки, с которой начинается вычерчивание кривой;

THO, THF - см. описание программ *ARCIA, SPIRAL*;

RO, RF - см. описание программы *SPIRAL*;

L - признак непрерывности линии: $L = 0$ - штриховая линия; $L = 1$ - сплошная линия.

Примечание. См. примечание описания программы *SPIRAL*.

Программа FATARC. Из текущей точки проводит в заданную точку линией заданной толщины дугу окружности заданного радиуса, с заданным направлением рисования относительно центра окружности (см.рис.3,в).

Обращения *CALL FATARC(R, XF, YF, J, D)*.

Параметры: D - толщина линии в миллиметрах;

остальные - см. описание программы *ARCIB*.

Примечания. 1, 2, 3 - см. описание программы (*ARCIB*).

4 - см. описание программы (*FATLIN*).

Программы вычерчивания эллипсов и их дуг

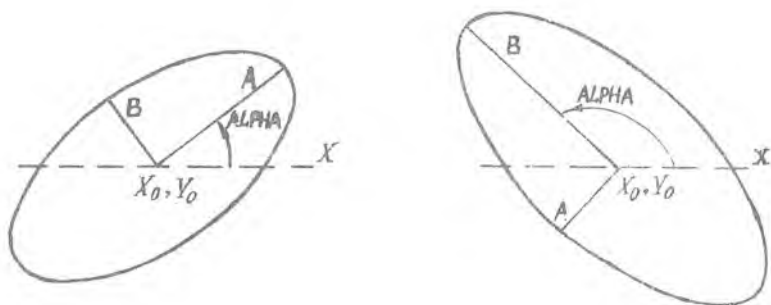
Программа ELLPS. Проводит эллипс с заданными размерами полуосей с центром в текущей точке и с заданным наклоном большой оси к положительному направлению оси X (рис. 4,а).

Обращения *CALL ELLPS(A, B, ALPHA)*.

Параметры: A, B - размеры полуосей;

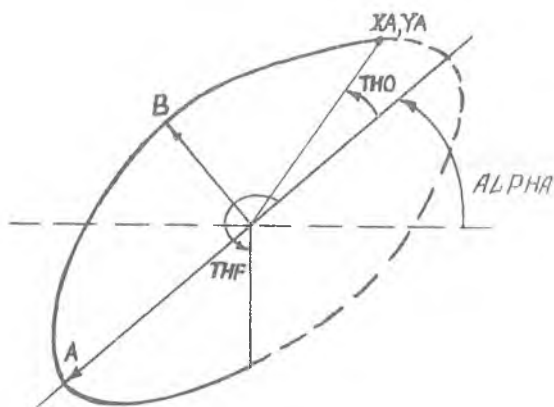
$ALPHA$ - угол наклона большой полуоси эллипса к положительному направлению оси X ; задается в градусах.

CALL ELPS(A, B, ALPHA)



a

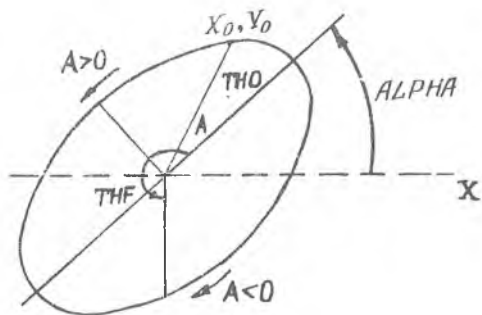
CALL ELIPS(XA, YA, A, B, ALPHA, THO, THF)



b

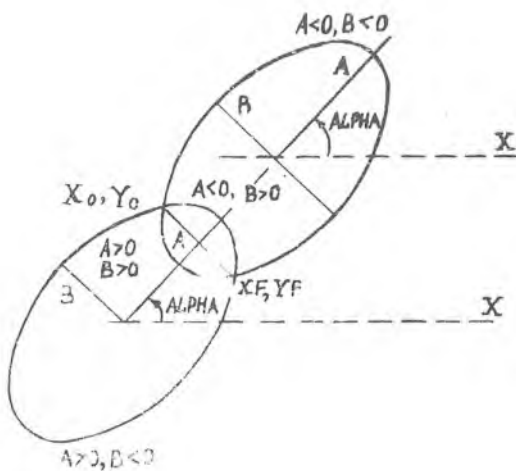
Р и с. 4. Программы вычерчивания

CALL ARCELA(A,B,ALPHA,THO,THF)



8

CALL ARCELB(A,B,ALPHA,XF,YF)



2

Примечания: I. Угол $ALPHA$ может задавать наклон любой полуоси эллипса (как A , так и B) в зависимости от того, какая из них больше.

2. После вычерчивания эллипса перо возвращается в исходную точку с координатами X_0, Y_0 .

Программа $ELIPS$. Проводит из заданной точки дугу эллипса с заданными размерами полуосей, заданными углами наклона одной оси эллипса к оси X и заданными углами наклона начального и конечного радиусов (рис. 4,б).

Обращение $CALL ELIPS(XA, YA, A, B, ALPHA, THO, THF)$.

Параметры: XA, YA - координаты точки, с которой начинается вычерчивание кривой;

A, B - размеры полуосей;

$ALPHA$ - угол наклона полуоси A эллипса к положительному направлению оси X ;

THO, THF - углы между полуосью A и, соответственно, начальным и конечным радиусами; все углы задаются в градусах; отсчитываются против часовой стрелки.

Примечания: I. При равных полуосях получаются окружность или ее дуги.

2. Вычерчивание производится против часовой стрелки.

Программа $ARCELA$. Из текущей точки проводит дугу эллипса с заданными размерами полуосей, заданным углом наклона одной оси эллипса к оси X и заданными углами наклона начального и конечного радиусов; а также с заданным направлением рисования относительно центра эллипса (рис. 4,в).

Обращение $CALL ARCELA(A, B, ALPHA, THO, THF)$.

Параметры: A, B - размеры полуосей;

$ALPHA, THO, THF$ - см. описание программы $ELIPS$.

Примечания: I. Если $A > 0$, то вычерчивание производится из начальной точки в конечную против часовой стрелки, а если $A < 0$, наоборот - по часовой стрелке.

2. Обращение к программе считается некорректным, если $|THO - THF| > 360^\circ$.

Программа $ARCELB$. Из текущей точки в заданную проводит дугу эллипса с заданными размерами полуосей и заданным направлением рисования относительно центра эллипса (рис. 4,г).

Обращение *CALL ARCELB(A, B, ALPHA, XF, YF)*.

Параметры: $|A|, |B|$ — размеры полуосей;

ALPHA — см. описание программ *ELLIPS* и *ARCELA*.

XF, YF — координаты конечной точки дуги.

Примечания: 1. Если $A > 0$, то вычерчивание производится из начальной точки в конечную против часовой стрелки, а если $A < 0$, наоборот — по часовой стрелке.

2. Если $B > 0$, то вычерчивается дуга, меньше или равная 180° . Если $B < 0$, то проводится дуга, больше или равная 180° . На рис. 4, г показаны четыре варианта дуг, изображаемых в зависимости от различных сочетаний знаков A и B .

3. Обращение к программе считается некорректным, если расстояние между начальной и конечной точкой превышает величину диаметра эллипса, параллельного отрезку, соединяющему эти точки.

Программы вычерчивания многоугольников

Программа *BOX*. Из заданной точки вычерчивает прямоугольником заданных размеров со сторонами, параллельными осям координат (рис. 5, а).

Обращение *CALL BOX(X, Y, XL, YL)*.

Параметры: X, Y — координаты заданной точки (левого нижнего угла прямоугольника);

XL, YL — размеры сторон прямоугольника, параллельных соответственно оси X и оси Y .

Примечание. Поднятое перо выводится в точку (X, Y) , и далее вычерчивается прямоугольник в направлении по часовой стрелке. После вычерчивания прямоугольника перо оказывается в точке с координатами (X, Y) .

Программа *RECT*. Из заданной точки вычерчивает прямоугольником заданных размеров, повернутый на заданный угол (рис. 5, б).

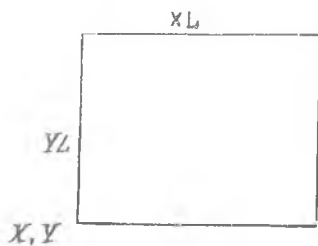
Обращение *CALL RECT(X, Y, H, W, TH)*.

Параметры: X, Y — координаты левого нижнего угла прямоугольника;

H, W — размеры сторон прямоугольника (соответственно, высота и ширина);

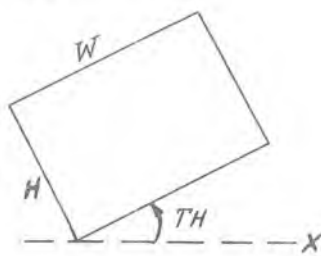
TH — угол наклона стороны прямоугольника с размером W к положительному направлению оси X ; задается в градусах; отсчитывается против часовой стрелки.

CALL BOX(X, Y, XL, YL)



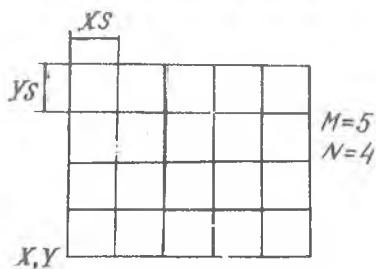
a

CALL RECT(X, Y, H, W, TH)



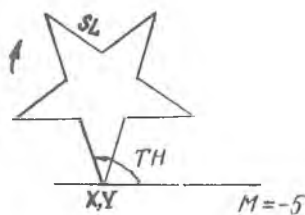
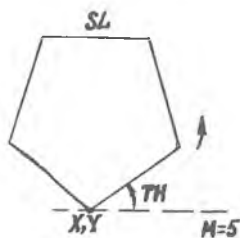
b

CALL GRID(X, Y, XS, YS, M, N)



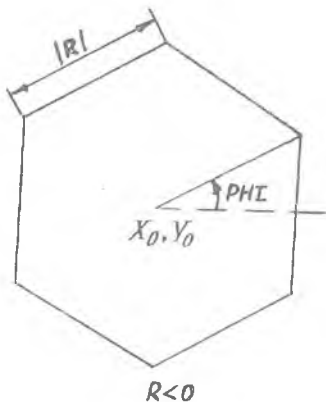
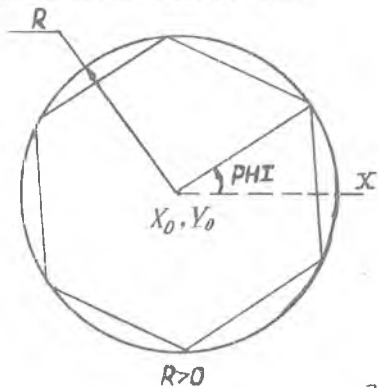
b

CALL POLY(X, Y, SL, M, TH)



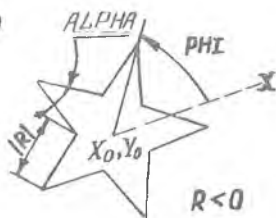
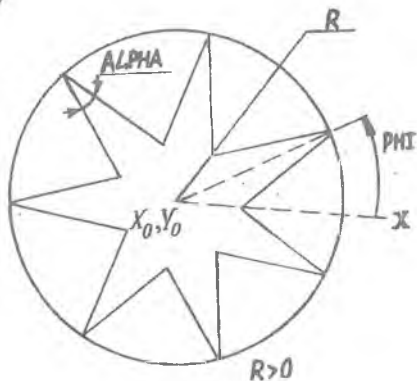
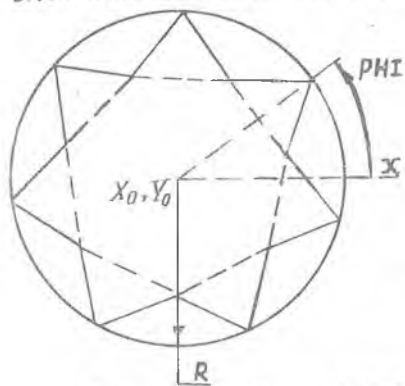
Р и с. 5. Программы вычерчивания

CALL POLG(R, M, PHI)



d

CALL STAR(R, M, ALPHA, PHI)



e

МНОГОУГОЛЬНИКОВ

Примечание. См. описание программы *BOX*.

Программа *GRID*. Вычерчивает из заданной точки прямоугольную сетку со сторонами, параллельными осям координат (рис. 5,в).

Обращение *CALL GRID(X, Y, XS, YS, M, N)*.

Параметры: X, Y - координаты левого нижнего угла сетки;
 XS, YS - размеры ячейки сетки соответственно по осям X и Y ;
 M, N - число ячеек сетки соответственно вдоль осей X и Y .

Примечание. Вычерчивание начинается в точке (X, Y) , в которую перо выводится поднятым, и заканчивается в ней же.

Программа *POLYG*. Из заданной точки вычерчивает правильный многоугольник (рис. 5,г).

Обращение *CALL POLYG(X, Y, SL, M, TH)*.

Параметры: X, Y - координаты точки, с которой начинается вычерчивание многоугольника;
 SL - размер стороны многоугольника;
 M - число сторон многоугольника;
 TH - угол наклона стороны, с которой начинается вычерчивание, к положительному направлению оси X ; задается в градусах.

Примечания: 1. Если $M > 0$, то вычерчивается выпуклый многоугольник, если $M < 0$, то вычерчивается невыпуклый (звездчатый) многоугольник.

2. Вычерчивание начинается в точке (X, Y) и заканчивается в ней же. Причем выпуклый многоугольник вычерчивается по часовой стрелке, а звездчатый - по часовой стрелке.

Программа *POLG*. Вычерчивает правильный многоугольник с центром в текущей точке (рис. 5,д).

Обращение *CALL POLG(R, M, PHI)*.

Параметры: $|R|$ - радиус описанной окружности или размер стороны многоугольника;
 M - число сторон многоугольника;
 PHI - угол в градусах, отсчитываемый против часовой стрелки от положительного направления

оси X до ближайшего луча, проведенного в вершину из центра многоугольника.

Примечания: 1. Если $R > 0$, то задается радиус описанной окружности, если $R < 0$, то задается размер стороны многоугольника, равный $|R|$.

2. По окончании работы программы перо возвращается в исходную точку.

Программа *STAR*. Вычерчивает звездчатый многоугольник с центром в текущей точке (рис. 5,е).

Обращение *CALL STAR(R, M, ALPHA, PHI)*.

Параметры: $|R|$ - см. описание программы *POLG* ;

$ALPHA$ - угол раствора луча;

M - число лучей звезды;

PHI - угол между положительным направлением оси X и осью ближайшего в направлении против часовой стрелки луча звезды; углы задаются в градусах.

Примечания: 1. См. описание программы *POLG*.

2. Если $ALPHA = 0$, то изображается правильная звезда-такая, у которой лучи, идущие через один, ограничены одной и той же прямой.

3. Если $ALPHA \geq 180 (M-2)/M$, то вместо звезды рисуется правильный выпуклый многоугольник.

4. По окончании работы программы перо возвращается в исходную точку.

Вычерчивание символов

Программа *SET*. Обращением к программе *SET* можно выбрать один из четырех наборов символов, приведенных в таблице.

Обращение *CALL SET(J)*.

Параметр J - номер набора символов: $J = 0$ - прописные русские и латинские буквы, цифры и знаки; $J = 1$ -

строчные русские и латинские буквы, цифры и знаки; $J = 2$ - прописные греческие буквы, циф-

ры, знаки и специальные математические символы;

$J = 3$ - строчные греческие буквы, цифры, зна-

ки и специальные математические символы. Цифры

и знаки во всех наборах одинаковые.

Примечания: 1. Обращаться к программе *SET* можно как до, так и после определения страницы. Информация о выбранном наборе

будет сохраняться до следующего обращения к программе SET в пределах одной страницы.

2. Если не было ни одного обращения к программе SET (т.е. по умолчанию), то выбираются прописные русские и латинские буквы. При закрытии страницы (ENDPG) восстанавливается исходное состояние (набор с прописными русскими и латинскими буквами).

Программа ITALIC. Обращением к программе ITALIC можно задать наклон (курсив) при вычерчивании символов (рис. 6,а).

Обращение CALL ITALIC(J).

Параметр J - признак выбора курсива: J = 0 - без наклона (курсива), J = 1 - правый наклон (курсив); J = -1 - левый наклон (курсив).

Примечания: 1. При J = ±1 каждый символ текста после данного обращения вычерчивается с наклоном к строке текста. Величина наклона (отклонение от вертикальной оси верхней части символа) составляет 0,4 от высоты символа.

2. Обращаться к программе ITALIC (так же как и к программе SET) можно и до определения страницы и после него. Будучи заданным, курсив распространяется на все тексты до следующего обращения к программе ITALIC в пределах одной страницы. Если не было ни одного обращения к этой программе, то задается стандартный режим - без наклона. Этот не стандартный режим устанавливается при закрытии страницы (ENDPG).

Программа SYMBOL. В заданном месте на странице вычерчивает строку заданного программистом текста символами заданной высоты с заданным углом наклона к оси X (рис. 6,б).

Обращение CALL SYMBOL(X, Y, SIZE, JTEXT, N, THETA).

Параметры: X, Y - координаты левого нижнего угла первой буквы текста или приращения к конечным координатам правого нижнего угла последней буквы текста, предшествующего данному;

SIZE - высота прямоугольника, в котором вычерчивается символ;

JTEXT - текстовая константа, содержащая заданный текст, может быть задана в двух формах: JTEXT = C - число символов в константе, N - заданный текст; JTEXT = C^N - заданный текст в апострофах;

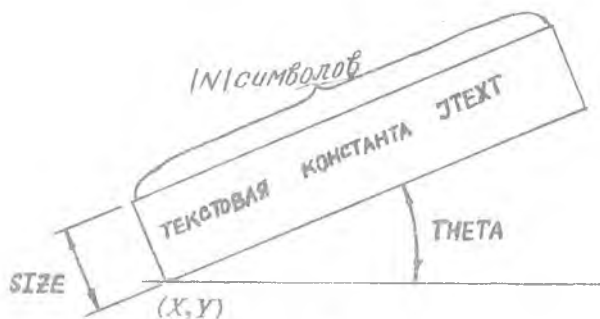
N - число символов в строке заданного текста;

N	SET(0)	SET(1)	SET(2)	SET(3)	N	SET(0)	SET(1)	SET(2)	SET(3)
1	/	/	/	/	35	C	c	Г	У
2	+	+	+	+	36	D	d	Δ	δ
3	-	-	-	-	37	E	e	E	ε
4					38	F	f	Ф	Ф
5	39	G	g	Г	У
6))))	40	H	h	H	η
7	\$	\$	\$	\$	41	I	i	I	i
8	*	*	*	*	42	J	j	Θ	θ
9	,	,	,	,	43	K	k	К	κ
10	((((44	L	l	Λ	λ
11	=	=	=	=	45	M	m	М	μ
12	'	'	'	'	46	N	n	N	ν
13	:	:	:	:	47	O	o	O	ο
14	Б	б	П	п	48	P	p	Π	π
15	Г	г	У	у	49	Q	q	Ξ	ξ
16	Д	д	С	с	50	R	r	Ρ	ρ
17	Ж	ж	С	с	51	S	s	Σ	σ
18	З	з	≠	≠	52	T	t	Τ	τ
19	И	и	⇒	⇒	53	U	u	Υ	υ
20	Й	й	Ξ	ξ	54	V	v	Ψ	ψ
21	А	а	∇	∇	55	W	w	Ω	ω
22	П	п	≠	≠	56	X	x	X	χ
23	Ф	φ	>	>	57	Y	y	Υ	υ
24	Ц	ц	≥	≥	58	Z	z	Z	ζ
25	Ч	ч	<	<	59	0	0	0	0
26	Ш	ш	≤	≤	60	1	1	1	1
27	Щ	щ	∞	∞	61	2	2	2	2
28	Ы	ы			62	3	3	3	3
29	Ь	ь	≡	≡	63	4	4	4	4
30	Э	э	~	@	64	5	5	5	5
31	Ю	ю	┘	\	65	6	6	6	6
32	Я	я	±	%	66	7	7	7	7
33	А	α	Α	α	67	8	8	8	8
34	В	β	Β	β	68	9	9	9	9

CALL ITALIC (-1) левый наклон
 CALL ITALIC (0) без наклона
 CALL ITALIC (1) правый наклон

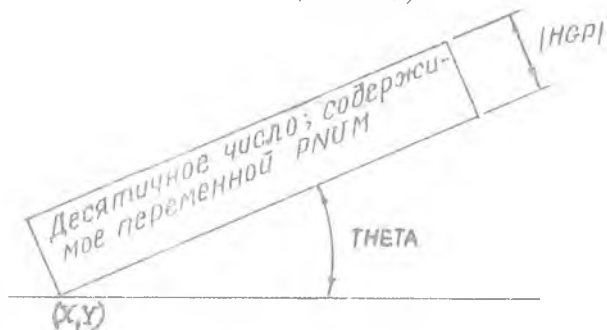
а

CALL SYMBOL (X, Y, SIZE, УТЕХТ, N, ТНЕТА)



б

CALL NUMBER (X, Y, HGR, PNUM, N, ТНЕТА)



в

Р и с. 6. Программы вычерчивания символов

ТНЕТА - угол наклона строки текста к оси X ,
в градусах.

Примечания: 1. Если $N > 0$, то X, Y считаются координатами на странице; если $N < 0$, то X, Y считаются приращениями. Последний вариант удобен в том случае, когда надо написать друг за другом несколько строк текста, отличающихся друг от друга наклоном, высотой, а также набором символов. Удобно пользоваться приращениями и для вычерчивания подстрочных (индексов) и надстрочных (степеней) знаков.

2. Текст, выходящий за пределы страницы, как слева, так и справа обрезается по ее границам.

3. Высота символов (точнее высота многоугольника, в котором вычерчивается символ) задается пользователем и не зависит от выбранного набора символов. Ширина этого прямоугольника составляет $2/3$ высоты. На пробел отводится такое же место, как и на любой другой символ.

4. Стандартные средства ввода-вывода (алфавитно-цифровые дисплеи, устройства подготовки перфокарт/перфолент, алфавитно-цифровые печатающие устройства и т.д.) дают возможность использовать лишь нулевой набор символов *SET(0)* (см. описание программы *SET* и таблицу) и не позволяют использовать строчные русские и латинские буквы, прописные и строчные буквы греческого алфавита, а также специальные математические символы (знаки интеграла, суммы, кланторы и т.д.). Поэтому в текстовой константе *JTEXT* символы первого *SET(1)*, второго *SET(2)* и третьего *SET(3)* наборов должны кодироваться символами нулевого набора, занимающими в наборе те же позиции, что и кодируемые символы (для знаков и цифр эта кодировка совпадает). Например, русская строчная буква "б", знак пересечения множеств \cap и знак интеграла \int , являющиеся в своих наборах четырнадцатыми символами (как видно из табл., все они занимают четырнадцатую позицию), кодируются в текстовой константе *JTEXT* одной и той же прописной русской буквой "Б".

Программа *NUMBER*. В заданном месте страницы, с заданным углом наклона к оси X вычерчивает заданное десятичное число с заданным числом знаков после запятой (рис. 6, в).

Обращение *CALL NUMBER(X, Y, HGR, PNUM, N, THETA)*.

Параметры: $X, Y, THETA$ - см. описание программы *SYMBOL*,

HGR - высота прямоугольника, в котором вычерчивается каждая из цифр числа;

PNUM - имя действительной переменной, которой перед обращением к программе *NUMBER* должно быть присвоено значение числа, подлежащего вычерчиванию;

N - число дробных знаков (справа от десятичной точки) в вычерчиваемом числе ($N \leq 6$).

П р и м е ч а н и я 1. Число может состоять из знака числа, целой и дробной частей. Если число положительное, то знак отбрасывается.

2. Если целая часть содержит больше шести цифр, то старшие разряды игнорируются, а младшим предшествует символ *.

3. Можно написать лишь старшие цифры числа, отбрасывая указанное количество младших цифр. Для этого необходимо задать $N < 0$; тогда отбрасываются $|N|$ младших цифр целой части.

4. В целых числах десятичная точка не вычерчивается (число дробных разрядов равно нулю: $N = 0$).

5. Если $HCP > 0$, то X и Y считаются координатами на странице, если $HCP < 0$, то X и Y считаются приращениями (см. примечание 1 к программе *SYMBOL*).

6. См. примечания 2,3 к программе *SYMBOL*.

АЛФАВИТНЫЙ УКАЗАТЕЛЬ ПРОГРАММ

<i>ARC</i>	18	<i>DASHD</i>	13	<i>MOYE</i>	10
<i>ARCELA</i>	24	<i>ELIPS</i>	22	<i>NUMBER</i>	35
<i>ARCELB</i>	24	<i>ELPS</i>	22	<i>POLG</i>	27
<i>ARCIA</i>	15	<i>FATARC</i>	20	<i>POLYG</i>	27
<i>ARCIB</i>	16	<i>FATLIN</i>	13	<i>RECT</i>	26
<i>ARCIC</i>	16	<i>GBID</i>	26	<i>SET</i>	30
<i>ARCID</i>	18	<i>ITALIC</i>	32	<i>SPIRAL</i>	20
<i>BOX</i>	26	<i>MOYA</i>	10	<i>STAR</i>	30
<i>CIRC</i>	15	<i>MOYB</i>	11	<i>SYMBOL</i>	32
<i>CIRCLE</i>	20	<i>MOYC</i>	11		

Составители: Олег Петрович Александров,
Сергей Васильевич Сухаянов

ГРАФОР: КОМПЛЕКС ГРАФИЧЕСКИХ
ПРОГРАММ НА ФОРТРАКе
(графические элементы)

Редактор Е.Д. Алякилова
Техн. редактор Н.М. Калениук
Корректор В.Г. Филиппова

Подписано в печать 10.12.85г. Формат 60x84¹/₁₆.
Бумага оберточная белая. Печать оперативная.
Усл.п.л. 1,9. Уч.-изд.л. 1,8. Т. 500 экз.
Заказ 173 Бесплатно.

Курьбинский ордена Трудового Красного Знамени
авиационный институт имени академика С.П.Королева,
г. Курьбинск, ул. Молодогвардейская, 151.

Областная типография имени В.П.Мяги,
г. Курьбинск, ул. Венцева, 60.