***1. Теория.***

Вид программы

Program n1 !имя программы

implicit none ! функция исключающая автоматическое задание переменных

!блок описания переменных

!тело программы

pause !функция для задержки окна выполнения программы

end program n1 ! !-комментарии

По умолчанию если не задана функция implicit none можно задавать переменные в теле программы. Тип переменных определяется при этом первой буквой переменной: i, j, k, l, m, n – integer; все остальные – real.

Можно задать свои условия следующим образом:

implicit intteger (a-c), real (x-z), character (len=10) (n)

Вывод переменных и массивов в окно можно следующими образами:

print \*, a

write(\*,\*) a, b

1.1. Константы.

Типы данных используемые в Fortran стандартны. Возможно несколько вариантов записи.

integer(4) a /2/, b /4/

integer :: a=2, b=4

integer(2) a, b

data a / 2/ & b /4/

или

data a, b /2, 4/

Вариантов существенно больше, но остальные варианты отличаются только синтаксисом. Разновидности integer 1, 2, 4. Просто integer соответствует integer(4).

Массивы задаются следующим образом:

integer :: a(10), b(2,3)

integer, dimension(10) :: a

а – одномерный массив, b – двумерный. Максимальная размерность массива – 7.

Аналогично объявляются данные вещественного типа real. Разновидности real 4,8. Просто real соответствует real(4).

Можно задать массив в несколько этапов:

real(8) :: arr

pointer arr !создает ссылку на массив arr

dimension arr(:, :) ! задали двумерный массив

allocate(arr(5, 3)) ! в памяти выделяется место под указанный размер массива

data arr / 1, 2, 3, 4, 5\*0, 2\*-2, 4\*100/

Константы можно задавать любого типа. При этом целые константы можно использовать для задания разновидности данных и размера массива.

integer, parameter :: n=4

real(n) :: arr(n, n-2)

Возможно записать тип данных с включенными выражениями, как показано в примере (деление только целочисленное).

Константы нельзя переопределить в программе. Вещественные и комплексные константы нужны для непосредственных расчетов. Так наиболее часто используются константы пи и мнимая единица.

Другая запись константы:

complex(4) z

parameter (z = (12.6, -5.2))

Возможно задание произвольного типа данных, к примеру:

type state

character(5) state !строка в 5 символов

integer(4) dim

real(8) lam

end type state

type(state) Rossler

Rossler = state(‘chaos’, 3, 1.58867)

1.2. Встроенные процедуры.

Существует множество встроенных функций и процедур. Тут приведены основные из них.

abs(a) – возвращает модуль числа(любого типа данных: полученный результат будет того же типа и разновидности данных что и переменная а). Еще раз обращается внимание, что a может быть одномерным или многомерным массивом. В этом случае результатом процедуры будет массив той же длины, составленный из элементов применения данной процедуры к элементу исходного массива. Данное правило распространяется на все функции в Fortran.

aint(a [, kind]) – обрезает вещественную величину а в сторону нуля до целого числа. -2.4 даст -2, 2.8 даст 2. [, kind] – здесь, и далее, необязательный аргумент. Можно записать:

real(2) a

aint(a, 4) ! даст результат с integer(4)

aint(a) ! даст результат с той же разновидностью, что была у a – integer(2)

anint(a [, kind]) – возвращает вещественное число соответствующее целому числу, полученному в результате округления вещественного а. -2.4 даст -2.0, 2.8 даст 3.0.

nint(a [, kind]) – возвращает целое число, соответствующее округлению вещественного а.

celing(a [, kind]) – возвращает наименьшее целое, большее или равное значению вещественного а. 8.01 даст 9, -5.6 даст -5.

floor(a [, kind]) – возвращает наибольшее целое, меньшее или равное значению вещественного а. 8.01 даст 8, -5.6 даст -6.

dim(a, b) – возвращает a-b, если a>b и 0, если a<b. a и b должны быть одинакового типа, вещественные или целые (допускаются разные разновидности у a и b)

dprod(x,y) – возвращает произведение двойной точности вещественных x и y, если они real(4).

mod(a,p) – возвращает остаток деления a на p. Оба целые или оба вещественные.

modulo(a,p) –возвращает a по модулю p. Оба целые или оба вещественные. a=q\*p+r. q- любое целое. r – результат функции.

sign(a,b) – дает +а если b больше или равно 0, и –а если b меньше нуля.

amax0(a1,a2,a3,…) – возвращает вещественный максимум из двух и более значений целого типа.

max(a1,a2,a3,…) – возвращает максимум из двух и более значений. Типы данных совпадают.

max1(a1,a2,a3,…) – возвращает целый максимум из двух и более значений вещественного типа. Все функции имеют аналогичный вид для минимумов.

Математические функции: exp(x), log(x), log10(x), sqrt(x), sin(x) – x в радианах, sind(x) - x в градусах, asin(x), asind(x), cos(x), cosd(x), acos(x), acosd(x), tan(x), tand(x), cotan(x), cotand(x), atan(x), atand(x), atan2(y, x) – atan(y/x), atan2d(y, x), sinh(x), cosh(x), tanh(x).

Границы определения и знаки можно погуглить.

1.3. Комплексные числа.

Комплексный тип данных задается с помощью complex. Разновидности complex 4,8. Просто complex соответствует complex (4).

Задание значений:

complex(8) :: c=(3.0, 4.0) – первое число действительная часть, второе мнимая. Тип complex только вещественный.

либо

complex(4), parameter :: i\_c=(0.0, 1.0)

complex :: z

z = 3+i\_c\*4

Все вычисления с комплексными числами выполняются автоматически:

z = (3+i\_c\*4)\*\*2

z = (3+i\_c\*4)\*(4-i\_c\*3)

z = (3+i\_c\*4)/(4-i\_c\*3)

Встроенная функция conjg(z) возращает комплексно-сопряженное. При этом z может быть массивом. Тогда функция применяется ко всему массиву. Аналогично если a, b, c – массывы равной длины можно записать a=b+c. Что будет аналогично циклу:

do i=1,n,1

a(i)=b(i)+c(i)

end do

Встроенная функция aimag(z) возращает мнимую часть комплексной переменной.

abs(z) – возращает модуль комплексного числа, тоесть abs(z)=sqrt(x\*\*2+y\*\*2). z может быть массивом.

Когда параметрами тригонометрических и логарифмических функций являются комплексные числа, то функции возаращают комплексное число, аргумент которого равен главному значению агрумента комплексного числа в радианах (определен от – пи до пи).

exp(z)=exp\*\*x\*(cos(y)+i\*sin(y))

log(z)=log(sqrt(x\*\*2+y\*\*2))+i\*atan(x/y)

sqrt(z)= (sqrt(x\*\*2+y\*\*2)\*exp\*\*(-i\*0.5\*atan(x/y))

Триганометрические функции аналогично.

***2. Задание.***

1.1. Константы.

Использовать в программе следующие типы констант: целые (в том числе для задания размера массива), вещественные, комплексные.

1.2. Встроенные процедуры.

Продемонстрировать задачу с использованием встроенных тригонометрических и алгебраических функций.

1.3. Комплексные числа.

Преобразовать двумерный массив в комплексный вектор. Продемонстрировать пример встроенных операций с комплексными числами (включая работу с логарифмическими и тригонометрическими функциями).