

*Балтийский государственный технический
университет
Им Д.Ф. Устинова «Военмех»*



КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ

*Санкт-
Петербург*



M Mathcad

Используемая версия Mathcad 15

Цель работы:

Ознакомиться со встроенными функциями и ключевыми словами программы. Научиться решать вычислительные задачи с использованием программной системы Mathcad.

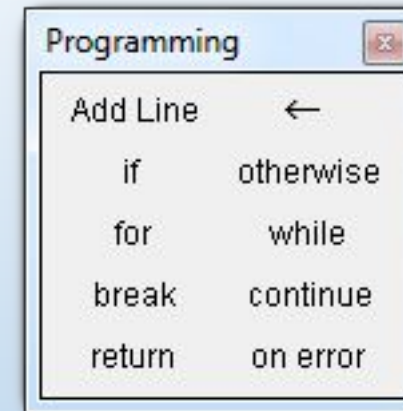


Для повышения гибкости Mathcad в системе предусмотрена возможность написания небольших программ для решения тех проблем, которые не могут быть реализованы стандартными средствами. Обычно прибегать к программированию приходится в тех случаях, когда стандартные средства либо не могут решить задачу, либо неэффективны.

Встроенный язык программирования Mathcad позволяет решать самые различные, в том числе и довольно сложные, задачи и является серьёзным подспорьем для расчётов.

Для вставки программного кода в документы Mathcad имеется специальная панель инструментов.

Большинство кнопок этой панели выполнено в виде текстового представления операторов программирования, поэтому их смысл легко понятен.



Подсказка: операторы программирования могут быть введены только с этой панели, но никак не с клавиатуры

Add Line	- добавить программную строку
←	<p>- оператор локального присваивания. В программе нельзя использовать оператор присваивания ":", вместо него используется оператор локального присваивания, отличие которого заключается в том, что локальная переменная определена только внутри своего блока и при выходе из программы теряет своё значение.</p> <p>Пример:</p> <pre> s ← 0 i ← 1 </pre>
if	<p>- условный оператор создаёт конструкцию вида:</p> <p>■ <i>if</i> ■</p> <p>где первый операнд выполняется, если справедливо условие являющееся вторым операндом, например:</p> <p>$x \leftarrow -1 \text{ if } x < 0$</p> <p>С оператором if возможны более сложные конструкции при использовании ещё одного оператора otherwise реализующего альтернативу. Аналог традиционной конструкции</p> <p>Если ... То ... Иначе ...</p> <p>Пример:</p> <pre> x ← -1 if x < 0 x ← 1 otherwise </pre>



for

- оператор цикла обеспечивает повторяющиеся вычисления если известно количество шагов.

$for\ i \in \mathbf{i}$

Первый операнд - переменная цикла, это интервальная переменная и её значения определены во втором операнде. Третий операнд - тело цикла может состоять из блока операторов и выполняется пока не исчерпаются все значения переменной цикла.

Пример:

$$\left. \begin{array}{l} s \leftarrow 0 \\ for\ i \in 0..100 \\ s \leftarrow s + i^2 \end{array} \right\}$$
break

оператор **break** введён для повышения гибкости программирования и позволяет закончить цикл досрочно, не исчервав всего списка.

Рассмотрим следующую задачу: нам необходимо найти первое вхождение 0 в числовом массиве и вернуть его индекс.

$$f(M) := \left. \begin{array}{l} for\ k \in 0..last(M) \\ \quad break\ if\ M_k = 0 \\ k \end{array} \right\}$$

Мы приводим работающую программу, где введена функция $last(M)$ которая возвращает последний индекс массива. Возвращаемым значением программы является последний выполняемый оператор - k .

continue

ещё один полезный оператор, позволяющий прервать выполнение текущей итерации и перейти к следующей.

Рассмотрим, например, задачу нахождения максимального и минимального элемента массива.

$$minmax(M) := \left. \begin{array}{l} min \leftarrow M_0 \\ max \leftarrow M_0 \\ for\ k \in 1..last(M) \\ \quad if\ M_k < min \\ \quad \quad min \leftarrow M_k \\ \quad \quad continue \\ \quad max \leftarrow M_k\ if\ M_k > max \\ \left(\begin{array}{l} min \\ max \end{array} \right) \end{array} \right\}$$

$$B := \begin{pmatrix} 1 \\ -5 \\ 4 \\ 8 \end{pmatrix} \quad minmax(B) = \begin{pmatrix} -5 \\ 8 \end{pmatrix}$$

Действительно, если $M_k < min$, то нет смысла проверять $M_k > max$, а лучше перейти к следующему шагу цикла.



<p>while</p>	<p>- оператор цикла с предусловием используется в тех случаях, когда заранее неизвестно количество шагов необходимых для решения задачи.</p> <p><i>while</i> ■</p> <ul style="list-style-type: none"> ■ <p>Условие проверяется перед началом каждого шага цикла. Реализуем, например, алгоритм вычисления квадратного корня a, используя итерационную формулу</p> $x_n = \frac{x_{n-1}}{2} + \frac{a}{2x_{n-1}}$ <p><i>sqrt(a)</i> :=</p> <table style="border-left: 1px solid black; border-right: 1px solid black; padding-left: 10px;"> <tr> <td style="padding-right: 5px;">$x \leftarrow \frac{a}{2}$</td> </tr> <tr> <td style="padding-right: 5px;">$\varepsilon \leftarrow 1$</td> </tr> <tr> <td style="padding-right: 5px;"><i>while</i> $\varepsilon > TOL$</td> </tr> <tr> <td style="padding-right: 5px; border-right: 1px solid black;">$z \leftarrow \frac{x}{2} + \frac{a}{2 \cdot x}$</td> </tr> <tr> <td style="padding-right: 5px; border-right: 1px solid black;">$\varepsilon \leftarrow x - z$</td> </tr> <tr> <td style="padding-right: 5px; border-right: 1px solid black;">$x \leftarrow z$</td> </tr> </table> <p style="margin-left: 10px;">x</p> <p>$sqrt(2) = 1.414$ $sqrt(10000) = 100$</p>	$x \leftarrow \frac{a}{2}$	$\varepsilon \leftarrow 1$	<i>while</i> $\varepsilon > TOL$	$z \leftarrow \frac{x}{2} + \frac{a}{2 \cdot x}$	$\varepsilon \leftarrow x - z $	$x \leftarrow z$
$x \leftarrow \frac{a}{2}$							
$\varepsilon \leftarrow 1$							
<i>while</i> $\varepsilon > TOL$							
$z \leftarrow \frac{x}{2} + \frac{a}{2 \cdot x}$							
$\varepsilon \leftarrow x - z $							
$x \leftarrow z$							
<p>return</p>	<p>- оператор return служит для прекращения работы программы и возврата результата.</p> <p><i>return</i> ■</p> <p>Допустим, нам необходимо вычислить позицию первого вхождения числа в массив.</p> <p><i>num(x, M)</i> := <i>for</i> $i \in 0..last(M)$</p> <p style="padding-left: 40px;"><i>return</i> i <i>if</i> $x = M_i$;</p> <p>$num(4, B) = 2$</p>						
<p>on error</p>	<p>- оператор on error служит для обработки ошибочных ситуаций типа "деления на ноль".</p> <p>■ <i>on error</i> ■</p> <p>Например, нам необходимо описать функцию $f(x) := \frac{1}{x}$, чтобы не было особенности в 0.</p> <p>$f(x) := 0$ <i>on error</i> $\frac{1}{x}$</p> <p>Тогда</p> <p>$f(0) = 0$ $f(0.01) = 100$</p> <p>Примечание. Оператор on error может использоваться в арифметических выражениях.</p>						



Вариант 1

Написать программу реализующую поиск совпадающих значений в двух матрицах A и B произвольной размерности. Предварительно импортировать матрицы из текстовых документов с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: V{номер варианта} _ {матрица или массив}
Например V1_B – матрица B для варианта 1



Вариант 2

Написать программу реализующую формирование массива В из одномерного массива А произвольной размерности так, чтобы в начале массива В стояли все положительные элементы, а затем все отрицательные. Порядок следования отдельно положительных и отдельно отрицательных элементов в массиве В должен быть таким же, как в массиве А.

Использовать один цикл.

Предварительно импортировать массив из текстового документа с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: V{номер варианта} _ {матрица или массив}
Например V2_A – массив А для варианта 2



Вариант 3

Написать программу реализующую перестановку местами максимального и минимального элементов массива A произвольной размерности.

Предварительно импортировать массив из текстового документа с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: V{номер варианта} _ {матрица или массив}
Например V3_A – массив A для варианта 3



Вариант 4

Написать программу реализующую замену первого элемента

i -й строки суммой всех элементов этой строки в матрице A произвольной размерности.

Предварительно импортировать матрицу из текстового документа с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: $V\{\text{номер варианта}\}_{\text{матрица или массив}}$
Например $V4_A$ – матрица A для варианта 4



Вариант 5

Написать программу реализующую вычисление произведения строго положительных элементов массива A произвольной размерности и вычисление суммы его элементов попадающих в интервал $[a, b]$.

Вывод результата сделать в виде массива в котором первый элемент равен произведению, а второй элемент сумме

Предварительно импортировать массив из текстового документа с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: V{номер варианта} _ {матрица или массив}
Например V5_A – массив A для варианта 5



Вариант 6

Написать программу реализующую поворот матрицы A произвольной размерности на угол $5\pi/2$ радиан.

Предварительно импортировать массив из текстового документа с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: V{номер варианта} _ {матрица или массив}
Например V6_A – матрица A для варианта 6



Вариант 7

Написать программу реализующую вычисление количества

элементов массива A произвольной размерности, удовлетворяющих условию $a \leq |y_i| \leq b$ и количества положительных элементов.

Вывод результата сделать в виде массива в котором первый

элемент равен количеству элементов, удовлетворяющих условию, а второй количеству положительных элементов.

Предварительно импортировать массив из текстового документа с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: $V\{\text{номер варианта}\}_{\{\text{матрица или массив}\}}$

Например $V7_A$ – массив A для варианта 7



Вариант 8

Написать программу реализующую перестановку местами минимального элемента матрицы A произвольной размерности и максимального элемента матрицы B произвольной размерности. Вывод результата сделать в виде массива в котором первый элемент новая матрица A, а второй матрица B.

Предварительно импортировать матрицы из текстовых документов с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: V{номер варианта} _ {матрица или массив}
Например V8_A – матрица A для варианта 8



Вариант 9

Написать программу реализующую формирование из матриц A и B одинаковой размерности новой матрицы C путём записи больших элементов из соответствующих ячеек матриц A и B.

Предварительно импортировать матрицы из текстовых документов с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: V{номер варианта} _ {матрица или массив}
Например V9_A – матрица A для варианта 9



Вариант 10

Написать программу реализующую нахождение такой строки матрицы A произвольной размерности в которой сумма элементов этой строки максимальна. Вывод результата сделать в виде массива в котором первый элемент равен номеру строки с максимальной суммой её элементов, а второй сумма её элементов.

Предварительно импортировать матрицы из текстовых документов с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: $V\{\text{номер варианта}\}_{\{\text{матрица или массив}\}}$
Например $V10_A$ – матрица A для варианта 10



Вариант 11

Написать программу реализующую поиск совпадающих значений в двух матрицах A и B произвольной размерности. Предварительно импортировать матрицы из текстовых документов с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: V{номер варианта} _ {матрица или массив}
Например V11_V – матрица B для варианта 11



Вариант 12

Написать программу реализующую формирование массива В из одномерного массива А произвольной размерности так, чтобы в начале массива В стояли все положительные элементы, а затем все отрицательные. Порядок следования отдельно положительных и отдельно отрицательных элементов в массиве В должен быть таким же, как в массиве А.

Использовать один цикл.

Предварительно импортировать массив из текстового документа с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: V{номер варианта} _ {матрица или массив}
Например V12_A – массив А для варианта 12



Вариант 13

Написать программу реализующую перестановку местами максимального и минимального элементов массива A произвольной размерности.

Предварительно импортировать массив из текстового документа с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: V{номер варианта} _ {матрица или массив}
Например V13_A – массив A для варианта 13



Вариант 14

Написать программу реализующую замену первого элемента

i -й строки суммой всех элементов этой строки в матрице A произвольной размерности.

Предварительно импортировать матрицу из текстового документа с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: $V\{\text{номер варианта}\}_{\text{матрица или массив}}$

Например $V14_A$ – матрица A для варианта 14



Вариант 15

Написать программу реализующую вычисление произведения строго положительных элементов массива A произвольной размерности и вычисление суммы его элементов попадающих в интервал $[a, b]$.

Вывод результата сделать в виде массива в котором первый элемент равен произведению, а второй элемент сумме

Предварительно импортировать массив из текстового документа с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: $V\{\text{номер варианта}\}_{\text{матрица или массив}}$
Например $V15_A$ – массив A для варианта 15



Вариант 16

Написать программу реализующую поворот матрицы A произвольной размерности на угол $5\pi/2$ радиан.

Предварительно импортировать массив из текстового документа с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: V{номер варианта} _ {матрица или массив}

Например V16_A – матрица A для варианта 16



Вариант 17

Написать программу реализующую вычисление количества

элементов массива A произвольной размерности, удовлетворяющих условию $a \leq |y_i| \leq b$ и количества положительных элементов.

Вывод результата сделать в виде массива в котором первый

элемент равен количеству элементов, удовлетворяющих условию, а второй количеству положительных элементов.

Предварительно импортировать массив из текстового документа с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: $V\{\text{номер варианта}\}_{\text{матрица или массив}}$

Например $V17_A$ – массив A для варианта 17



Вариант 18

Написать программу реализующую перестановку местами минимального элемента матрицы A произвольной размерности и максимального элемента матрицы B произвольной размерности. Вывод результата сделать в виде массива в котором первый элемент новая матрица A, а второй матрица B.

Предварительно импортировать матрицы из текстовых документов с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: V{номер варианта} _ {матрица или массив}
Например V18_A – матрица A для варианта 18



Вариант 19

Написать программу реализующую формирование из матриц A и B одинаковой размерности новой матрицы C путём записи больших элементов из соответствующих ячеек матриц A и B.

Предварительно импортировать матрицы из текстовых документов с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: V{номер варианта} _ {матрица или массив}
Например V19_A – матрица A для варианта 19



Вариант 20

Написать программу реализующую нахождение такой строки матрицы A произвольной размерности в которой сумма элементов этой строки максимальна. Вывод результата сделать в виде массива в котором первый элемент равен номеру строки с максимальной суммой её элементов, а второй сумма её элементов.

Предварительно импортировать матрицы из текстовых документов с помощью стандартной функции READPRN.



Подсказка: Текстовые файлы к варианту задания выбираются следующим образом: $V\{\text{номер варианта}\}_{\{\text{матрица или массив}\}}$
Например $V20_A$ – матрица A для варианта 20

