

= 2 athPiper* % Computer %\$gebra &y!tem ; or 1eginner!.....	22
=.1 9umeric 8!. &ymbol\$c Computation!.....	22

1- Predicate ; unction!.....	89
1- .1 ; inding Prime umber! / ith % ' oop.....	90
1- .2 ; inding The ' ength 4(% &tring / ith The ' engthCD; unction.....	92
1- .3 Converting umber! To &tring! / ith The &tringCD; unction.....	93
1- .- ; inding Prime umber! / hich , nd / ith ? C%nd 2 u\$ti-\$ine ; unction Ca\$!D	93
1- .= , : erci!e!.....	9-
1- .=.1 , : erci!e 1.....	9=
1- .=.2Cerci!e 1	9=p ;
1= ' i!t!* 63Â.....> te !e %!D/ i	

17.1.10 Ta 09

19.2 , :ercile!	13?
19.2.1 , :ercile 1	13?
19.2.2 , :ercile 2	13?
19.2.3 , :ercile 3	138
19.2.- , :ercile -	138
19.2.= , :ercile =	138

2 Introduction

and here is the same formula in text form*

a M : N2 I - 0h0: I 3/?

2 of the computer algebra systems contain a mathematics-oriented programming

v.93u - 09/21/09

3 Downloading And Installing MathRider

% + , ' %

MathRider is a Java-based application and there (ore a current version of &un5!

v.93u - 09/21/09

& ' (\$)

13. æøw\$ ö & ' (\$)

totalAmount) and loop75.

The process of associating a value with a variable is called **assignment** or **binding**. The value is stored in memory, and the variable is a reference to that memory location. You can think of it as putting a label on a box. The label is the variable name, and the box contains the value. The process of putting the label on the box is called assignment. The process of putting the value in the box is called binding.

'et! reca\$cu\$ate —

Setter! o(their name!. The 5 5 in can be !. ug à(a!
parenthe!e! to an e: pre!!ion and the 5 5 in can be thoug à(out
or removing parenthe!e! (rom an e: pre!!ion.

5 . !! à

2.2. Exercise „

) ? 1 ; = 8 8 

v.93u - 09/21/09

button! in the upper right corner of the page. The user to move back and forth between previous and next page and the button navigate back to the home page.

The function name in the function tree is point to section in the BT2 ' documentation so the user can access function information either by navigating

4 Using MathRider As A Programmers Text Editor

We have covered some of MathRider's mathematical capabilities and this section

v.93u - 09/21/09

IntraAP -

v.93u - 09/21/09

Introduction To Programming

v.93u - 09/21/09

v.93u - 09/21/09


```
/      2      5 =  
      *  
/#
```


3 I

8 I
1 8 I

/ #4

/ 2 5 =
* -


```
;D218024617g2 can have their content T6$ d1 ped6
```

nTearate6\$ine! i de!ire

• @B/

v.93u -

v.93u - 09/21/09

bracket! after the variable it is bound to. % character position is determined

by its distance from the start side of the string starting at 1. ; or else: ampse) in the

*

00p`p€"P00Á? dLú €

:.11.2 %\$ercise 2

```

-   =   < 1 8 :   1           1 < 8 8           =           8
      =           ;           ;           , 8   =           ,8  ( )
= 8           :   4;           :           :
      1           8           8 :           = : :           8           ,8  ( )
= 8           :           :           :   4;           :

```

```
*5 ?976, .RD STG +H U &-EP@COV I
```

```
,8 ( 2      K L)I
,8 ( 2      K L)I
,8 ( 2      K L)I
```

:.11.3 %\$ercise 3

```

E   ,8  ( ) = 8           :   4;           :   8 8
      =           =   < 1           1 (           4           <   <   :
      8           ) *

```

```
*5 QWXY/!Z ( ) [ J 2A#MNKL\0 5I I
```

```
,8 ( 2      K L)I
,8 ( 2      K L)I
,8 ( 2      K L)I
```

:.11.4 %\$ercise 4

```

-   =   < 1   1 4           1   :   :   4;           B
  4   7           1 4 < 8           1   = 4           1 4
      4   =           = >   =           4 8 ; :

```

```
/4
```

```
#
```

```

-           1 4           1   :   :   4;
      B           4

```

```
#
```

```
*5 ?976, .RD STG +H U &-EP@COV I
```

```
< *5 ;8: =1 ]3 4 ' < >@ pHS $ p 0Pa 0ç ö
```

```
      * _  
      & : , == 8 *  
      > &4  
/#
```

1> Rectangular Selection Mode And Test Area .litting

@

&

4ne capabi\$ity that 2 ath. ider ha! that a cming

C

•
" 5 "

v.93u - 09/21/09

I
2 : %%
K
2 22

v.93u - 09/21/09

v.93u - 09/21/09

1, Predicate & unctons

% i! a (unction that either return!


```

4 * -
4 * .
4 * .
4 * .
4 * -
" 4 * .
$ 4 * -
% 4 * .
/ #

```

This program is ordered (air\$y " e\$) but it is limited because it prints a line (or each

G 1 ()
%

) the e: pre!!ion i! a

and the 09 e: pre!!ion i! an

' i!t! can a\$!o ho\$d other \$!t! a! !ho" n in the (o\$\$o"

4 G *5 M %2 %2 4@*8 @+8 @@52 %N
M %2 %2M 2 2 N2 %N

4 G K L
%

4 G K L
%

4 G K L
M 2 2 N

4 G K L
%

The e: pre!!ion in the

po!ition in the \$!t i! another

" ich contain! th

9d

) a@D-

Z7ö , 0

101A956T000000000 !c%£ Dp`À. öa !ñ€•ß & `a! `a:ö Ð \$ `! !ö

% can also be used to search through a list. The following program uses a
(function and an (function to search through a list to see if it contains
the number 3. It is found in the third element.

5 0

; or the (o\$\$" ing e: ercile!) create a ne" 2 ath. ider " or#!heet (i\$e ca\$\$ed

C

D. ; or e: amp\$e) 6ohn &mith5!

" or#!heet " ou\$d be ca\$\$ed*

%(ter thi! " or#!heet h. 5been c'reated) as a byoucie that



12.5.2 Exercises

; or the (os) ing e: ercile!) create a ne" 2 ath. ider " or#!heet (i\$e ca\$ed

C

v.93u -



The following code should contain a function being used to locate a number in a list

```
7 (M %2 2 2 2 2 2 2 2 "2 $N2 )
```

```
7 (M %2 2 2 2 2 2 2 2 "2 $N2 )
```

The function can also be used with predicate function like Contain to change their result to the opposite truth value

```
+ 7 (M %2 2 2 2 2 2 2 2 "2 $N2 )
```

13.1.3 find

```
. : ( 2 )
```

The function searches a list (or the first occurrence of a given element). If the element is found the function returns the index of the element. If it is not found the function returns nil.

```
. : (M 2 2 2 $ "2 N2 )
```

```
. : (M 2 2 2 $ "2 N2 " )
```

13.1.4, Count

```
7 ( 2 )
```

13.1.2)elect;<

```
& 8 ( : 8 = 8 2 )
```

return! a \$!t that contain! a\$ the e: pre!!ion! in a \$!t " hich ma#e a

insert! an element into a list at a given index)
delete! an element from a list at a given index) and

delete! an





13.1.14 **6ea.**)ort;<

D & (2 8 4)

13.2.3)u+;<

```
& 4( )
```

can (ind the !um o(a \$!t that i! pa!!ed to it*

```
InR te!t' i!t *M . andomInteger8ector(10)1)99D
```

```
M 2$ 2"%2 2 2$ 2 %2 2 2 N
```

```
& 4( G )
```

```
G *5 %
M 2 2 2 2 2 2 2 2"2$2 %N
```

```
& 4( G )
```


/ #4

The values that are returned from user defined functions can also be assigned to variables. The following example uses a `MathPiper` class to define a function called `add` and then this function is used in the `MathPiper` constructor to assign a value (even integer) to the variable `fact`.

/4

```

K      1      :      1
      G

```


"
\$
%
/#

1:.1.2

```
/#
```

1:.1.3 The **break** function

The **break** function is used to end a loop early and here is it! casting (format*

```
9 3()
```

1:.1.,



; or, achD) but it i! !impSer than the!e t" o. Bere are the t" o ca9p3" ! gb 1V8/ 0à2 K c



