

ACCESS 2010
Week 1 a.m., June 14-18
group assignment via modular arithmetic

Let's assign names to the numbers from 1 to 21. Let's number the 7 summer work weeks of ACCESS from 1 to 7. Assume the week 1 group assignments are shown column by column in the matrix below:

$$\begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 \end{bmatrix}$$

So group 1 consists of (people with) numbers 1,2,3; group 2 consists of numbers 4,5,6, etc. You can see the Maple commands that made and displayed this matrix at the end of this document. I copied the output matrix and pasted it into my text field above.

Consider the person assigned the number "x". We can recover their location in the matrix (and hence also their initial group assignment) using quotients and remainders from integer division. Notice that their row is determined by the remainder of "x" (actually $x-1$ is more convenient), after we divide by 3. One way to express this is to write

$$x - 1 = 3q + r$$

where q is the quotient of $x - 1$ and r is the remainder. Then x 's row is given by $r + 1$.

The arithmetic of keeping track of remainders is called "modular" or "clock" arithmetic, and the divisor d is called the "clock number" or "modulus". The Maple function that determines the remainders (also called "residues") for divisor d is called the **mod d** function. We incorporate it below.

Create a math field below your current cursor position in a Maple document by clicking your mouse on the $[>$ icon in the menu bar. After typing your math command (which ends with $;$ (to see output) or $:$ (to hide it)), you must hit the "return" or "enter" key to execute the command. You can enter multi-line commands by pressing the "shift" key while you hit "enter". The colon followed by the equal sign, $:=$, is how you assign names to variables, functions, matrices, etc. Functions are defined using the arrow notation, i.e. a minus sign followed by a greater than sign, $->$, and commands end with semicolons $;$ if you want to display output, or colons $:$, if you don't. After entering a command or commands in a math field, hit "return" or "enter" to execute it.

```
[ > R:=x->(x-1 mod 3) +1; #R(x) will equal x's row above
      R:=x->((x-1) mod 3)+1
[ > R(17);R(21); #test the row function
      2
      3
```

Notice that initial group assignment is the initial column of the matrix above and is related to the quotient of $x - 1$ divided by 3. We can get this quotient using the truncation function, "trunc". You might have called this the greatest integer function in one of your previous math classes.

```
[ > C:=x->trunc((x-1)/3)+1;    #C(x) should be x's column
      C:=x->trunc( $\frac{1}{3}x - \frac{1}{3}$ )+1
[ > C(17);C(21);    #check the column function
                        6
                        7
```

Your assignment for Nick and Zhu:

- 1) Explain the algorithm your group decided to use for making group assignments, in terms of how you moved the rows each week to create the excel document you emailed to Rosemary.
- 2) Then use the "trunc" and "mod" commands to define a Maple function which yields the group assignment for person "x" in week "t". Work in this Maple document, just adding text and math commands below.
- 3) Send Zhu and me your final Maple file as an attachment, to wangzhu@math.utah.edu and korevaar@math.utah.edu. Make sure you've checked that your function works - because that's what one of us is going to do!

Hint: You can name your group assignment function whatever you want. For example, if I choose to call the function "Group", then I would define it by a command that looks like

```
[ > Group:=(x,t)->?????;    #figure out ????? ?!
```

By the way, here are the commands I used to make the matrix displayed at the start of this document:

```
[ > restart:           #restarts Maple, and clears all memory
                          #if you want to use any old commands after
                          #restarting, move your cursor to the command
                          #field you need and hit the enter or return
key
> with(linalg):         #loads the linear algebra package of commands
                          #that contains the matrix and evalm commands
Warning, the protected names norm and trace have been redefined and unprotected
> week[1]:=matrix(3,7); #makes a matrix called week_1 with
                          #three rows and seven columns
                          week_1 := array(1..3, 1..7, [])
```

Here's how I defined the entries:

```
[ > for j from 1 to 7 do   #j is the column index in this do loop
      for i from 1 to 3 do #i is the row index
          week[1][i,j]:=3*(j-1)+i: # := means "is defined to be"
      end do:             #ends subloop indexed with i
    end do:               #ends loop indexed with j

> evalm(week[1]);       #evalm means "evaluate (and display) the matrix
                          #this is what I copied and pasted into the
                          #text field.

                          [ 1  4  7  10  13  16  19 ]
                          [ 2  5  8  11  14  17  20 ]
                          [ 3  6  9  12  15  18  21 ]
```