Lecture #14 – AWK Part II (Chapter 6)

- Conditional Statement

```
if  (expression)
{
        action1
}
else
{
        action2
}
```

```
/Rich/ { if (x) print x }
```

```
{
        if (avg >= 65)
                grade = “Pass”
        else
                grade = “Fail”
}
```

- Conditional Operator

```
expr ? action1 : action2
```

```
grade = (avg >= 65) ? “Pass” : “Fail”
```

- While Loop

```
while (cond)
{
        action
}
```

```
/^[0-9]/ {

        i = 1
        while (i <= 4)
        {
                print $i
                ++i
        }
}
```

- Do Loop

```
do
{
        Action
}
while (cond)
```

```
BEGIN {
        do {
                ++x
                print x
        } while (x <= 4)
}
```

- For Loop

```
for (set_counter; test_counter; incr_counter)
        Action
```

set_counter = initial value of loop variable
test_counter = Condition for loop continuation
incr_counter = Increment loop variable (after loop, before test)

```
{
        for (i = 1; i <= NF; i++)
                print $i
}
```

```
{
        for (x = 3; x <= NF; x++)
                if ($x < 0)
                {
                        print “Bottomed out!”
                        break
                }
}
```

- next statement

The next statement gets the next line of input from the input file, and restarts execution of the program at the top of the script.

```
/^dn/ { next }
```

- Arrays

  Array[sub] = value

  You do not have to declare the size of the array, just use them

  flavor[1]=”cherry”
  print flavor[1]

- Associative arrays

  Like before, arrays with arbitrary indicies

  Special version of “for” loop to support associative arrays

  for (var in array)
  Action using array[var]

  Example:

  ```
  for (temp in item)
      print temp, item[temp]
  ```

  Example:

  ```
  { manuf[$1]++ }

  END { for (name in manuf)
          print name, manuf[name]
  }
  ```

- Split

  Used to make arrays out of variables

  n = split(str, array, delim)

  z = split($1, fullname, “ “)

  First name would be fullname[1], and last name would be fullname[2]

- Multi-dimensional arrays

  file[NR, i] = $i
  file[2, 4] = “test test”

- Example: bitmap

```
BEGIN { FS = ","
        WIDTH = 12
        HEIGHT = 12

        for (i = 1; i <=WIDTH; ++i)
            for (j = 1;  j <= HEIGHT; ++j)
            bitmap[i, j] = "O"
}

{
    # read input "x, y"

    bitmap[$1, $2] = "X"
}


END {
    for (i = 1; i <= WIDTH; ++i)
    {
        for (j = 1; j <= HEIGHT; ++j)
            printf("%s", bitmap[i, j])

        printf("\n")
    }
}
```

- Awk built-in functions

sub – substitute (1st occurrence only)

Syntax :

```
sub(regexp, sub string)
sub(regexp, sub string, target string)
```

Examples :

```
awk '{ sub(/Mac/, "Macintosh"); print }' filename
awk '{ sub(/Mac/, "Macintosh", $1); print}' filename
```

gsub – global substitute

same as sub except it works for every occurence in line or field

index – search inside string

Syntax:

index(string, substring)

Examples:

```
$  awk '{ print index("hollow", "low") }' filename
4
```

length – string's length

Syntax:

length
length(string)

Example:

```
$  awk '{ print length("hello")}' filename
5
```

substr – substring

Syntax:

substr(string, start)
substr(string, start, length)

Example:

```
$  awk '{ print substr("Santa Claus", 7, 6)}' filename
Claus
```

- Random numbers

The rand function generates a pseudo-random floating point number b/w 0 and 1. The srand function uses the time of day to seed the rand function.

Example :

```
#  Generate NUM random numbers between 1 and TOPNUM

awk –v NUM=$1 –v TOPNUM=$2 '
```

```
        BEGIN {
                if (NUM <= 0)
                        NUM = 6

                if (TOPNUM <= 0)
                        TOPNUM = 20

                sprintf ("Pick %d of %d\n", NUM, TOPNUM)

                srand()

                for (j = 1; j <= NUM; ++j)
                {
                        do
                        {
                                select = 1 + int(rand() * TOPNUM)
                        } while (select in pick)

                        pick[select] = select
                }

                for (j in pick)
                        printf("%s ", pick[j])

                printf("\n")
        }'
```

- Example: wc

```
BEGIN {
        words = 0
        lines = 0
        chars = 0
}

{
        words += NF
        lines++
        chars += length($0)
}

END {
        print words, lines, chars
}
```

- Example: Print all words that appear more than 10 times in input

```
{
        for (i = 1; i <= NF; i++)
                count[$i]++
}

END {
        for (item in count)
        {
                if (count[item] > 10)
                        print item, count[item]
        }
}
```