Assignment 2

Total : 27pts

In [ ]:

```
# write a program to print your name and student number using variables and variabl
$name = "Sam Shute"
$Number = "0471137"
"Name - $Name
Number - $number"
```

```
Name - Sam Shute

Number - 0471137
```

Name: John Doe
Student #: W0123456

Part A - Write Code! (16pts)

For each item below, determine the appropriate PowerShell code to generate the desired output.

In [ ]:

```
# write a small program that take a numeric variable and print out "EVEN" if the va
# 1 pt
# am I an even number?
$oddEven = 10
if ($oddeven -eq 10){
    Write-Host "even"
}
```

```
even
```

In [ ]:

```
# write a small program that will look at a single character and print out "letter"
# if it is between the letters 'a' and 'z'
# 2 pts

$character = 'x' # am I a letter?
if ($character -eq "x"){
    Write-Host "That's a letter"
}
```

```
That's a letter
```

In [ ]:

```
# compare a given number and if the number is '12345' print "UNLOCKED!" otherwise h
# 2 pts

$code = "12346" # did I disable the alarm?
```

```
if ($code -eq 12346){
    Write-Host "unlocked"
} else {
    Write-Host "alarm"
}
```

```
unlocked
```

In [ ]:

```
# given an integer, determine and then output whether the number is "negative", "po
# 3 pts

$integer = -5 # am I positive, negative or zero?
if ($integer -lt 0){
    Write-Host "That's negative"
} elseif ($integer -eq 0){
    Write-Host "That's zero"
}else{
    Write-Host "That's positive"
}
```

```
That's negative
```

In [ ]:

```
# create a program that solves that age old question,
# "Would you choose $1 Million dollars or get a single penny that doubles every day
# (e.g. Day 1 = 1¢, Day 2 = 2¢, Day 3 = 4¢, ..., Day 30 = ?)
# by calculating the total amount you would get on Day 30
# note 1: just output the amount you would have on Day 30, don't include other amou
# note 2: display the amount in dollars, not pennies
# 3 pts
Write-Host "Would you want 1 million dollers right now or a penny that doubles it a
$penny = 1
for ($penny -le 12){
    $penny * 2

}
```

```
Would you want 1 million dollers right now or a penny that doubles it amount every d
ay dor 30 days?
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
```

In [ ]:

```
# create a program that outputs the following pattern, not including the '#'s,
# using loops (i.e. not individual print statements)
# 5 pts

#         o
#        ooo
#       ooooo
#      ooooooo
#     ooooooooo
#    ooooooooooo
#   ooooooooooooo
#  ooooooooooooooo
# ooooooooooooooooo
#ooooooooooooooooooo
# ooooooooooooooooo
#  ooooooooooooooo
#   ooooooooooooo
#    ooooooooooo
#     ooooooooo
#      ooooooo
#       ooooo
#        ooo
#         o

for ($line = 1; $line -le 15; $line++) {
    (" " * (15 -$line)) + ("oo" * $line)
    }
for ($line = 14; $line -ge 1; $line--) {
    (" " * (15 -$line)) + ("oo" * $line)

}
```

```
              oo
             oooo
            oooooo
           oooooooo
          oooooooooo
         oooooooooooo
        oooooooooooooo
       oooooooooooooooo
      oooooooooooooooooo
     oooooooooooooooooooo
    oooooooooooooooooooooo
   oooooooooooooooooooooooo
  oooooooooooooooooooooooooo
 oooooooooooooooooooooooooooo
oooooooooooooooooooooooooooooo
 oooooooooooooooooooooooooooo
  oooooooooooooooooooooooooo
   oooooooooooooooooooooooo
    oooooooooooooooooooooo
     oooooooooooooooooooo
      oooooooooooooooooo
       oooooooooooooooo
        oooooooooooooo
         oooooooooooo
          oooooooooo
           oooooooo
            oooooo
             oooo
              oo
```

Part B - Figure it out! (11pts)

For the following, you will need to understand how the program works in order to solve the problem.

In [ ]:

```
# modify the following to output the odd numbers "5 3 1 -1 -3 -5"
# 2 pts
$output = ""
for ($i = 5; $i -ge -5; $i -= 2) {
    $output = "$i"
    Write-Host "$output " -nonewline
}
```

```
5 3 1 -1 -3 -5
```

In [ ]:

```
# make a single change to make the following statement True
# 1 pt
10 -le 10 -and ($True -or $False) -and (4 -lt 5)
```

```
True
```

In [ ]:

```
# the following program converts the number '1' to the day 'Sunday',
# modify the program so that if the variable 'day' is between 1-7,
# the program will output the corresponding day 'Sunday'-'Saturday'
# 3 pts
$day = 1
if ($day -eq 1) {
    $day = "Sunday"
}
Write-Host $day
```

```
Sunday
```

In [ ]:

```
# determine what value to change the variable "magic_num" to in order to print out
# 5 pts

# Just change the number to the real magic number! Yep, that's it!
# Please keep your answer from others. The fun is solving the puzzle after all!
$magic_num = 315

# don't modify the following code
$check = 0
while ($check -lt 129) {
    for ($i = 0; $i -lt 5; $i++) {
        $magic_num -= 7
    }
    $check += 9
}
if ($magic_num -eq 0) {
    Write-Host "You found the magic number!"
} else {
    Write-Host "Not yet. Keep trying!"
}
```

```
Not yet. Keep trying!
```