1. (40 points) Give the output for each of the following inputs. No need for explanations if your answer is correct. But wrong answers with (brief) explanations may earn partial credit.

(a) \[
\text{In}[]: \{r, s, t, u, v, w\} \rightarrow \{a, b, c\} \rightarrow \{b\}
\]
\[
\text{Out}[]:
\]

(b) \[
\text{In}[]: \text{Range}[6] \rightarrow \{a, b, c\} \rightarrow \{b\}
\]
\[
\text{Out}[]:
\]

(c) \[
\text{In}[]: \{2, 15, 3, 11, 14, 9\} \rightarrow \{a, b, c, d, e\} \rightarrow \{a, c, e\} \rightarrow \{b \rightarrow d || b \rightarrow d + 1\}
\]
\[
\text{Out}[]:
\]

(d) \[
\text{In}[]: (\#^2 + \#)[3]
\]
\[
\text{Out}[]:
\]

(e) \[
\text{In}[]: \text{Map}[\{\#[[2]], \#[[1]]\}] \& \text{Table}[[i, i^2], \{i, 1, 3\}]]
\]
\[
\text{Out}[]:
\]

(f) \[
\text{In}[]: f[x_] := 2x; \text{Nest}[f, 5, 3]
\]
\[
\text{Out}[]:
\]

(g) \[
\text{In}[]: f[ch_] := \text{FromCharacterCode}[\text{ToCharacterCode}[ch] + 1];
\quad \text{Map}[f, \{\text{"C"}, \text{"a"}, \text{"t"}\}]
\]
\[
\text{Out}[]:
\]

Hint: \text{ToCharacterCode}[\text{"string"]} gives a list of the integer codes corresponding to the characters in a string. \text{FromCharacterCode}[n] gives a string consisting of the character with integer code n.

(h) \[
\text{In}[]: \text{Mod}[15, 3] == 0 \&\& \text{Mod}[15, 4] \neq 0
\]
\[
\text{Out}[]:
\quad \text{Hint: } \text{Mod}[m, n] \text{ gives the remainder on division of } m \text{ by } n.
\]

Next page for Part II.
Do only two of the following problems. Please circle the two you are choosing.

(a) (30 points) Write a rule for finding a palindrome in any given list of strings. (A palindrome is a string that reads the same forward and backward; e.g.: mom, refer, cddc are all palindromes; glass, xyz, are not palindromes). Example:

```
In[ ]: {"hi","mom","break"} apply-your-rule-here
Out[ ]: {mom}
```

```
In[ ]: {"hi","human","break"} apply-your-rule-here
Out[ ]: {"hi","human","break"}
```

If there is more than one palindrome in the input list, it’s enough that your rule find at least one of them.

(b) (30 points) Write a function `primesBetween[m_,n_]` that outputs a list of all prime numbers p such that m<p<n (note that these are strict inequalities). Example:

```
In[ ]: primesBetween[4,12]
Out[ ]: {5,7,11}
```

```
In[ ]: primesBetween[4,11]
Out[ ]: {5,7}
```

(c) (30 points) Write a function `josephuskp[k_,p_]` that shows the survivor in a list of p people when every kth person is removed; make your function also show the list of survivors each time a person is removed. (Recall that the people are seated in a circle.) Example:

```
In[ ]: josephuskp[3,5]
Out[ ]: {{{1,2,3,4,5}, {4,5,1,2}, {2,4,5}, {2,4}, {4}}
``