1- For an arbitrary number $n$, you are required to calculate the sum of the following series:

$$S = 1 - \frac{1}{2!} + \frac{1}{3!} - \frac{1}{4!} + \frac{1}{5!} - \cdots + \frac{1}{n!}$$

a(15p)- Draw a flowchart diagram for the algorithm

b(10p)- Analyze your algorithm for $n=6$.

$N = 6$, $Tpl = 0$, $Nfak = 1$

$I = 1$  
$Nfak = Nfak * I = 1 * 1 = 1$

$1 \mod 2 = 1 \rightarrow Tpl = Tpl + \frac{1}{Nfak} = 0 + \frac{1}{1} = 1$

$I = 2$  
$Nfak = Nfak * I = 1 * 2 = 2$

$2 \mod 2 \neq 1 \rightarrow Tpl = Tpl - \frac{1}{Nfak} = 1 - \frac{1}{2} = \frac{1}{2}$

$Tpl = S = 1 - \frac{1}{2!} + \frac{1}{3!} - \frac{1}{4!} + \frac{1}{5!} - \cdots + \frac{1}{n!}$
\[ I = 3 \quad Nfak = Nfak \ast I = 2 \ast 3 = 6 \]
\[ 3 \mod 2 = 1 \rightarrow Tpl = Tpl + \frac{1}{Nfak} = 1 - \frac{1}{2} + \frac{1}{6} \]
\[ I = 4 \quad Nfak = Nfak \ast I = 6 \ast 4 = 24 \]
\[ 2 \mod 2 \neq 1 \rightarrow Tpl = Tpl - \frac{1}{Nfak} = 1 - \frac{1}{2} + \frac{1}{6} - \frac{1}{24} \]
\[ I = 5 \quad Nfak = Nfak \ast I = 24 \ast 5 = 120 \]
\[ 5 \mod 2 = 1 \rightarrow Tpl = Tpl + \frac{1}{Nfak} = 1 - \frac{1}{2} + \frac{1}{6} - \frac{1}{24} + \frac{1}{120} \]
\[ I = 6 \quad Nfak = Nfak \ast I = 120 \ast 6 = 720 \]
\[ 2 \mod 2 \neq 1 \rightarrow Tpl = Tpl - \frac{1}{Nfak} = 1 - \frac{1}{2} + \frac{1}{6} - \frac{1}{24} + \frac{1}{120} - \frac{1}{720} \]

2- a(20p) - Draw a flowchart diagram for the Insertion sort algorithm to sort a given array in reverse order (the highest-valued item will be the first element in the array).

[Flowchart diagram for Insertion Sort in reverse order]
b(15p) - Use the array \( A = \{1, 8, 5, 11, 3, 9\} \) to analyze your algorithm.

\[
\begin{align*}
I &= 1 \\
J &= 1 \quad (J > 0 \text{ and } A[0] < 8) \Rightarrow \text{True} \\
&\quad J = J - 1 = 0 \\
J &= 0 \quad (J > 0 \text{ and } A[-1] < 8) \Rightarrow \text{False} \\
&\quad A[0] = \text{Temp} = 8 \\
A \text{ becomes } A &= \{8, 1, 5, 11, 3, 9\} \\
I &= 2 \\
\text{Temp} &= A[2] = 5 \\
J &= 2 \quad (J > 0 \text{ and } A[1] = 1 < 5) \Rightarrow \text{True} \\
&\quad J = J - 1 = 1 \\
J &= 1 \quad (J > 0 \text{ and } A[0] < 5) \Rightarrow \text{False} \\
&\quad A[1] = \text{Temp} = 5 \\
A \text{ becomes } A &= \{8, 5, 1, 11, 3, 9\} \\
I &= 3 \\
\text{Temp} &= A[3] = 11 \\
J &= 3 \quad (J > 0 \text{ and } A[2] = 1 < 11) \Rightarrow \text{True} \\
&\quad J = J - 1 = 2 \\
J &= 2 \quad (J > 0 \text{ and } A[1] = 5 < 11) \Rightarrow \text{True} \\
&\quad J = J - 1 = 1 \\
J &= 1 \quad (J > 0 \text{ and } A[0] = 8 < 11) \Rightarrow \text{True} \\
&\quad J = J - 1 = 0 \\
J &= 0 \quad (J > 0 \text{ and } A[-1] < 11) \Rightarrow \text{False} \\
&\quad A[0] = \text{Temp} = 11 \\
A \text{ becomes } A &= \{11, 8, 5, 1, 3, 9\} \\
I &= 4 \\
\text{Temp} &= A[4] = 3 \\
J &= 4 \quad (J > 0 \text{ and } A[3] = 1 < 3) \Rightarrow \text{True} \\
&\quad J = J - 1 = 3 \\
J &= 3 \quad (J > 0 \text{ and } A[2] = 5 < 3) \Rightarrow \text{False} \\
&\quad A[3] = \text{Temp} = 3 \\
A \text{ becomes } A &= \{11, 8, 5, 3, 1, 9\} \\
I &= 5 \\
\text{Temp} &= A[5] = 9 \\
J &= 5 \quad (J > 0 \text{ and } A[4] = 1 < 9) \Rightarrow \text{True} \\
&\quad J = J - 1 = 4 \\
J &= 4 \quad (J > 0 \text{ and } A[3] = 3 < 9) \Rightarrow \text{True} \\
&\quad J = J - 1 = 3 \\
J &= 3 \quad (J > 0 \text{ and } A[2] = 5 < 9) \Rightarrow \text{True} \\
&\quad J = J - 1 = 2 \\
J &= 2 \quad (J > 0 \text{ and } A[1] = 8 < 9) \Rightarrow \text{True} \\
&\quad J = J - 1 = 1 \\
J &= 1 \quad (J > 0 \text{ and } A[0] = 11 < 9) \Rightarrow \text{False} \\
&\quad A[1] = \text{Temp} = 9 \\
A \text{ becomes } A &= \{11, 9, 8, 5, 3, 1\} \quad \text{ALL SORTED!!}
\]
3(20p) - Convert each time formula to the best possible big-O notation. Do not include any spurious constants in your big-O answer.

<table>
<thead>
<tr>
<th>Time Formula</th>
<th>Big-O</th>
</tr>
</thead>
<tbody>
<tr>
<td>10n</td>
<td>n</td>
</tr>
<tr>
<td>$2n^2$</td>
<td>$n^2$</td>
</tr>
<tr>
<td>3 times log (base 2) of n</td>
<td>log n</td>
</tr>
<tr>
<td>$2n^2 + 10n$</td>
<td>$n^2$</td>
</tr>
</tbody>
</table>

4(25p) - Compare the arrays with stacks and queues in terms of their roles, access restrictions and ease of search, insert or delete operations.

Arrays are easy to access, insert and delete for particular items. Appropriate for any data found in a database; personal, financial etc. An item in an array can be accessed immediately, if the index is known, or by searching.

Stacks and queues are usually created and used to carry out a particular task during the operation of a program and they are discarded afterwards. They are designed to enforce restricted access; only one item can be add or removed at a given time. A stack only allows access to the last item inserted while a queue only allows access to the first item inserted.