Q4-i. The value of 'i' doubles for every iteration. For this reason, 'i' will reach the value of 'n' in $\log_2(n)+1$ steps. We can see this by dry-running the code for smaller values of 'n'. For example, if 'n'= 2, the loop will run twice, if 'n'= 4, the loop will run thrice and if 'n'=8, the loop will run 4 times and so on. We can see that there is a growing rate in the number of iterations the loop takes in $O(\log_2(n))$.

Q4-iia. Both the upper and lower bounds will be $O(n^3)$ as each loop will run in $O(n)$.

Q4-iib. Each index of W will store the sum that starts the row index at W in P and goes all the way to the column index in W in P, i.e. W[i][j] = P[i]+P[i+1]+…+P[j-1]+P[j].

Q4-iib.

```
for (x=0; x<n; x++) {
        for (y=x+1; y<n; y++) {
                W[x][y] = W[x][y-1]+P[y];
        }
}
```