Q4

i) The Time complexity of the following program will be O(log(n)) because in our For loop at each iteration our iterator will keep multiplying by 2. Thus will be log(n) base 2

ii) In our lower bound case we will have 0(n) because if n=1, then it will begin the first for loop when x = 0 and when y becomes x+1 it won't fulfill the condition of the second loop that y which is 0+1 is less than n which is 1. Hence O(n) time complexity.

In our upper bound case we will have O(n^3) because if our n is large then it will first run the first loop then run the second loop both of which are O(n). So till here O(n^2) and then in the function when y is n-1 and x is 0 then the loop in the function **func** will run the loop n times hence that is O(n) so the total time complexity of the code fragment is O(n^3).

iii) For all values in our 1-D array P[n], we add all value in array form i to j and store into the 2-D arrays W[i][j]. So for example for the case W[x][y], we add all the values in P[x]+P[x+1]+ . . . + P[y],
where y begins from x+1 and iterates through to n-1.

iv)

```
int x, y, n;   ...
/* P is a 1-D array of size n integers and W is a 2-D array of size n x n integers */

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