Q1

-Total sum calculate kiya

-2D boolean array banayi initialized with false

-the idea is to find whether there exists a subset which makes up the target sum

-base case: 0th column all rows true since zero sum is possible with empty subset

- base case: array ka pehla element agar total sum se chota ya barabar hai to woh khud k jitna sum bana le ga

-2nd row onward filling

-Ek dafa array ka element include karlien or ek dafa exclude kar lien

-agar kisi ek jagah se bhi true a gaya toh is ka matlab hai k ek esa subset exist karta hai jo k woh particular sum bana sakta hai

- at the end last row ap ko batye gi k is array mien koi esa subset hai ya nahi jo k har column number wala sum bana sakta hai

- s1 apko direct mil jaye ga or s2 ap total-s1 se nikal lien ga, phir s1 or s2 ka absolute difference le lein ga or sath sath min abs difference ka track rakhien ga

- last pe min abs difference jis ka track rakh rahe the who return kar dien ga

Q2

* 1d array bane gi jis k har index pe us index tak longest increasing subsequnce ki length pari hogi
* Maxlis k variable se max lis ka track rakhien ga
* Ap outer loop mien har element pe traverse karo ga
* Inner loop mien ap shuruse leke us particular element tak checking karo ga
* Checking mien ye hai k agar shuru wala elemnt current elemnt se chota hai to current elemnt us k sath jora ja sakta hai or us shuru wale elemnt tak ki lenghth plus 1 hojaye ga. Shuruwale ki length plu1 or current element ki length mien se maxchoose kar lien ga or current element ki jagah par rakh sien ge. Isi tarha current elemnt apne peeche wale sare elements k sath checking kare ga is is tarah inner loop poora chal jaye ga
* Hum sath sath maxlis ka track bhi rakh rahe hain
* End pe maxlis wala variable return kar dien ga

Q3

* isPalindrome ka function ye check karta hai k given string palindrome hai ya nahin
* main function mein 1D array hai jis k har index pe right side se le kar us index tak minimum partitions ka count para hai. N+1 the index dummy index hai base case ko handle karne k liye
* outer loop string k har index k liye chale ga starting from N to 1
* inner loop us particular index se le kar end tak chale ga or check kare ga k us waqt i or j ki boundary k andar wali string palindrome hai ya nahi, agar hai to ek partition to us boundary ki aa gayi plus jo baki bachi hui string hai us ki partitions.
* Min partitions ka track rakhien
* Inner loop k bahar us particular index tak ki minimum partitions store kar dien
* Isi tarha har index k liye us se agle sare elements k sath partions bana bana kar dekhein ga or min partitions save kar lien ga
* Jab outer loop pehle index ka result bhi store kar le ga to us index pe minimum partitions pari hongi jo k us poori string mien possible hain
* Humien chahiyien cuts na k partiotions to un minimum partitions mien se 1 minus kar k return kar dein ga

Q4

* Target sum ka function sari array ka sum nikalta hai
* Kuch edge cases hadle karta hai
* Edge case 1: target total sum se bara nahi ho sakta , agar aisa hai to yeh target achieve kare k zero ways hain
* Edge case 2: agar total sum even hai to partitions (even,even) ya (odd,odd) banien gi or unka difference lazmi even hoga. Agar total sum odd hai to partitions (even,odd) ya (odd,even) or unka diference lazmi odd hoga. Ab (even-even) or (odd-odd) even hota hai yani total-target lazmi even hoga. Agar total-target odd hai to who target exist nahi kar sakta or us target ko achieve karne kzero ways hain
* S1-S2=target mien s1 or s2 ek constant number honge to agar hum in mien se koi ek dhond lien to doosra lazmi exist karta hoga or jitnne subsets ka sum is number k equal hoga utne hi ways honge
* S1=total-S2
* S1-S2= total-S2-S2=total-2S2=target
* Total- 2S2=target
* S2= (Total-target)/2
* Hume woh subsets dhondne hain jin ka sum S2 k equal hai. Jitne yeh subsets honge utne hi ways honge
* To yahan se age hum findWays function ko call kardien ga jo k woh subsets dhond de ga jinka ka sum S2 k equal hai
* findWays mien bane gi 2D array
* base case: agar array ka pehle element 0 hai to 0 ka sum wale 2 subsets hain, ek 0 wala subset or doosra empty subset
* base case : agar array ka pehla element zero nahi hai to 0 k sum wala 1 hi subset hai or woh hai empty subset
* base case: agar array ka pehla index zero nahin hai to woh target se chota ya barabar hain to us element wala subset apne equal target ka sum rakhta hoga
* age second row se onwards filling hogi
* har element ko har target k liye ek dafa include or ek dafa exclude kar k check karien k ese kitna subsets hain jin ka sum target k equal hai.
* End pe hamien bottom right corner of the 2D array pe number of subsets mil jayien ga jin ka sum target k equal hai or hum ye value return kardien ga

Q5

* Jo standard algorithm hum ne parha hai woh end pe optimal solution deta hai
* Hum given rod length k liye inner loop dobara chalien ga jis mien k sare k sare solutions ko optimal se match karien ga. Jo jo match hota jaye ga hum ek counterko increment karte jain ga
* At the end counter mien total number of optimal solutions pare honge jo hum return kar dien ga