William Randall	Problem. On the planet Sigma, robots excavate chunks of a very precious cristaline. These chunks can be divided into smaller part only on the Earth. Once a month, a spaceship comes to Sigma to bring cristaline to the Earth. The spaceship cannot take more than k pounds of the load. Naturally, it is
	desirable to take as much cristaline as the spaceship can.
405 167936 HWY	(100 pts) Write an efficient algorithm that will allow bringing the maximal amount of cristaline to the Earth. Prove that it is correct and estimate its time complexity.
initialize 2d array [ [ammount of chunks +1][k+1] // the if subsequence we given som	
Let Chunks [] he an ordered array of avaliable chunk sizes	
possible_ship:	
for i=1 to k:	
T[0][i] = False // ho items in list + sim + 0	
end for	
for i=0 to ammount of chunks:	
T[i][o] = True // Sum; s O then True	
end for	
for i= 1 to ammount of chunks:	
for i=1 to k:	
if Chunks [:-1] 7 i :	
T[i][i] = T[i-1][i] // do not include if Chunk size > weight in	
else	
T[i][i] = T[i-1][i] or $T[i-1][i-1][i-1]$   include or exclude	
endif	
end for	
end for	
Let weighting = sum	
while verant_ind != 0 : / find height index that is the	
if T [ammount of chunks][ re; aut _ iN ] is The then break from While	
weight ind -	
if meiant-ind is 0 then return hull set	
end while	
Continue	
lack lack	

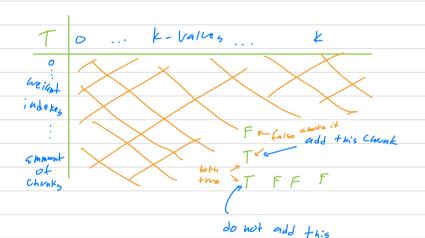
```
Let chunk _ ind be ammount of Chunks
While weight ; nd 1 = 0 // make the list of avaliable chunks
  If T[Chunk-ind-1] [weignt-ind] is True they Chunk-ind-
  else
   Add Chunks [Chunk_ind] to BC
  Chunk_ ind --
  heiant ind = weight _ ind - Chunks [ chunk _ ind]
 en if
end while
Return BC
Time complexity for this will he O (ammount of chunks. K) be cause
filling out the array Twill take the longest ammount of time
for this algorithm.
This alsorithm will return a set of the chunks which will result in
the maximum weight which is below k.
First we make T which is a 21 array of bools which is touc if
there exists a subsequence of Chunks with given sum.
For the column with com=o it is all true.
for no items in the list it is all false.
Then it loops through the Tarray and if the Chunk size is
 greater than the correct k then it will not include said chunk.
 Otherwise it will try to include + exclude the Chunk I take the
```

or of those two values. Once the T array is complete

it finds the largest k value with a true value in the Tamay

initialize list of best chunks, BC

at the index of the ammount of Chunts. Then from that pes; t; on it will look at the value above it in array T and if sqid value is true then that Chunk is not included in the final set, but if it is not true then you include that chunk and you move lower in the k values (aka decrease your weight index by the height ammount of the Chunk you just added to the list). If does this unxil it reaches a sum of 0 then returns the list.



Chunk