Algorithms Laboratory (CS29203)
Assignment 2: Recursion, Divide & Conquer
Department of CSE, IIT Kharagpur

25th August 2022

Question-1

The intelligence department of India is working to decipher the coded words of some secret organizations. These organizations use a special device to transmit message that works similar to the telegraph system used in old days. The machine has 8 switches numberd from 1 to 8, and each switch can transmit 3 to 4 alphabets. For example, the switch 1 has alphabets 'A', 'B' and 'C' associated with it - pressing the switch once will transmit the alphabet 'A', if it is pressed twice consecutively, then the alphabet 'B' will be transmitted, and pressing consecutively three times will transmit 'C'. The code words of the corresponding switches in the machine are as follows:

```
Switch 1 --> A, B, C
Switch 2 --> O, N, M
Switch 3 --> I, H, G
Switch 4 --> J, L, K, Z
Switch 5 --> D, F, E
Switch 6 --> W, Y, X
Switch 7 --> T, U, V, S
Switch 8 --> R, Q, P
```

The goal of intelligence department is to analyze all possible combinations of messages generated from these switches. That is, given a sequences of switches, the target is to find all possible combination of words that can be formed from these switches. For example, for the switch sequence $5, 6$, the possible messages will be the following: DW, DY, DX, FW, FY, FX, EW, EY, EX.

Your task is to write a *recursive* code to solve the above problem. Your program will store the predefined switch codes, take the switch sequence as an integer array, and output all possible combinations of messages that can be generated from the given switch sequences.

Example:

```
Enter the switch sequence: 1, 2, 3
-----
All possible messages are: AOI, BOI, COI, ANI, BNI, CNI, AMI, BMI, CMI, AOH, BOH, COH, ANH, BNH,
                           CNH, AMH, BMH, CMH, AOG, BOG, COG, ANG, BNG, CNG, AMG, BMG, CMG
```

Question-2

At the end of your classes in Nalanda complex, you are going to the adjacent Subway restaurant for eating. You are very hungry and have ordered a big chicken wrap. Suddenly some of your classmates have noticed you and have come inside to take part in the feast from your chicken wrap. So now you have to divide your wrap for k of your friends and yourself (so in total $k+1$ parts). Your friends are greedy & will take the parts that have the large amount of chicken pieces and will leave the part having the least amount of chicken pieces for you. You need to do something about it. You plan an idea to cut the wrap into $k+1$ parts in such a way that you get a reasonably good amount of chicken pieces in your part (although that will be less than the amounts in your friends' parts, but still you want to maximize the amount for you).

Let's say that the size of chicken pieces are not the same and the amount of chicken is quantified by the weight of each piece. The whole wrap can be considered to have a consecutive sequence of chicken pieces with varying weights. Now while cutting the wrap, you want to optimize in such a way that you maximize the total weight of chicken pieces in your part. Assume that the number of pieces and weight of each piece is an integer.

For example, consider a wrap with 6 pieces of chicken where the weights (in grams) of each pieces are given by $6, 3, 2, 8, 7, 5$ sequentially. Let's say there are 2 of your friends, so you have to divide the wrap into 3 parts. So the optimal cutting will consist of parts having the first 2 pieces into the first part (resulting the total weight of $6+3=9$ grams), the second part consists of having the next 2 pieces (total weight of $2+8=10$ grams), and the third part consists of having the last 2 pieces (total weight of $7+5=12$ grams). With this division, your friends get the parts with big amounts of 10 and 12 grams of chicken, and you get the part with 9 grams of chicken. You can't do a partition better than this. Note that the chicken pieces are consecutive inside the wrap and you can't alter their positions.

As another example, if the wrap has 9 pieces with consecutive weights of $1, 2, 3, 4, 5, 6, 7, 8, 9$ grams and there are 5 friends, then the optimal cutting will have the following contents: Part1 ($1+2+3=6$ grams), Part2 ($4+5=9$ grams), Part3 (6 grams), Part4 (7 grams), Part5 (8 grams), Part6 (9 grams). So you get the part with 6 grams of chicken.

You have to write a program that will take the number of friends (k) as an integer, number of chicken pieces and their corresponding weights as an integer array, and print the maximum weight of chicken piece(s) that you can get after the division. You will have to solve this problem in 2 ways:

(a) Implement a brute force solution for the above problem.

(b) Implement a $O(n \log n)$ or better solution (*Hint: think in the idea of binary search*).

Example 1:

```
Enter the number of friends: 2
Enter the number of chicken pieces: 9
Enter the consecutive weights (in grams) of the pieces: 1,2,2,1,2,2,1,2,2
-----
Maximum weight of chicken piece(s) that you will get is: 5 grams
```

Example 2:

```
Enter the number of friends: 8
Enter the number of chicken pieces: 9
Enter the consecutive weights (in grams) of the pieces: 5,6,7,8,9,1,2,3,4
-----
Maximum weight of chicken piece(s) that you will get is: 1 gram
```