1. (a) we use a $k+1 \times n+1$ table M to solve this problem

$M[i, k]$ = the maximum investment using $(f_i, f_{i+1}, \cdots f_n)$ and exactly k dollar.

base cases $M[i, 0] = 0$ and $M[n, k] = f_n(k)$ and for

general case it will be

$$M[i, k] = \max_{\text{for } m \le k} (f_i(m) + M[i+1, k-m])$$

evalution order we will the table k in an increasing order and i in the decreasing order.

and the answer will be $\max_{\text{for all } k} (M[0, k])$.

space: $O(nk)$ table

in total $O(k) + O(nk^2) = O(nk^2)$

Complexity: for each entry there are $O(k)$ values to consider the maximum which also takes $O(k)$, there are $O(nk)$ entries so DP is $O(k) \cdot O(nk) = O(nk^2)$. then find max $O(k)$

Correctness: the correctness should be obvious, for current investment f_i with remaining k dollars, we can invest m from 1 to k dollar in f_i and the rest depends on the ~~the~~ maximum investment using $(f_{i+1}, f_{i+2} \cdots f_n)$ with the remaining mony which is exactly $M[i+1, k-m]$ as we defined then take the max of all possible m will give the maximum required.

notice, the problem asks the maximum investment of $(f_1 \cdots f_n)$ using ~~less than~~ at most k dollar, so at the end, we take max of $M[0,k]$ for all k. to be the answer. Therefore, the DP should give out the correct answer.

1(b): from the recursive formula in (a), it's not hard to find that for $M[i,k]$ fixed i, it only depends on the entris in the $M[i+1,\cdot]$ entries, therefore if we only need to calculate the maximum investment at the end, instead of saving the entire table, two ~~a single~~ columns will be sufficient, one for current values, one for previously computed values and save current values to previously computed values once a column is finished, therefore, two columnells only takes $O(k)$ space in total