Anvesh Muppeel R11840667

(1) Original Revenue of the Rod cutting problem is i.e. from the notes we know

$$r_n = \max_{1 \le i \le n} (p_i + r_{n-i})$$

Here p_i is defined as price of the rod and r_{n-i} is the revenue.

Pseudo code for it

```
CUT-ROD(P, n)
if n=0
    return 0
q = -∞
for i=1 to n
    q = max(q, P[i] + CUTROD(P, n-i))
return q
```

But in problem given that each cut has a fixed cost i.e. C and the price p_i.

Due to the extra cost, whenever each cut happend the cost will be added and revenue will be affected. Including the extra cost into the revenue calculation.

The new cost rod-cutting is calculated as

$$r_n = \max\left(\max_{1 \le i < n} (p_i + r_{n-i} - C),\ p_n \right)$$

$$r_n = \max\left((p_1 + r_{n-1} - c), (p_2 + r_{n-2} - C) \ldots (p_{n-1} + r_1 - c), p_n \right)$$

New pseudo code for above equation

```
1.  CUT-ROD(P,n,c)
2.  let r[0.1...n]
3.  r[0] = 0
4.    for i=1 to n
5.          price = p[i]
6.          for j=1 to i-1
7.                price = max(price, (p[j]+r[i-j]-c))
8.        r[i] = price
9.        return r[n]
```

In the 1st line we are passing cost along with p and n

In 2nd line we are declaring the revenue array.

In 3rd line assigning 0 to r[0] since it is 0

In 4th line we are ~~passing~~ looping from 1 to max length i.e. n.

In 5th line & 6th we are taking max price for max length since we are not cutting for the last value

In 7th line we are calculating the max value & assigning it to revenue array in 8th line

(2) Example

Activity (S)	Starting Time (S_i)	Finishing Time (f_i)	Duration
1	4	6	2
2	1	5	4
3	5	9	4

In the above example two sets are compaitable.
i.e. S{1} and S{2, 3}
from the above sets the derived solution would be (i.e activity of least duration from among the above) is $\boxed{S\{1\}}$
But there according to the maximum-size set ~~of~~ i.e. ~~S{2,3}~~ $\boxed{S\{2,3\}}$ which is optimal solution