

```

1 // Assignment 7 - Priority queuezy
2 class Solution
3 {
4     public:
5     int findMaximizedCapital(int k, int w, vector<int>& profits, vector<int>& capital) {
6         vector<pair<int, int>> projects;
7         for (int i = 0; i < profits.size(); i++)
8             {
9                 projects.push_back(make_pair(capital[i], profits[i]));
10            }
11        sort(projects.begin(), projects.end());
12
13        priority_queue<int> maxHeap;
14        int index = 0;
15        int n = projects.size();
16
17        for (int i = 0; i < k; i++)
18            {
19                while (index < n && projects[index].first <= w)
20                    {
21                        maxHeap.push(projects[index].second);
22                        index++;
23                    }
24                if (!maxHeap.empty())
25                    {
26                        w += maxHeap.top();
27                        maxHeap.pop();
28                    } else {
29                        break;
30                    }
31            }
32        return w;
33    }
34 };

```