Pseudo code description of the algorithm

Employee class

- **has_sub() show is the employee has subordinates in the class constructor**

- **total_influence() be the sum of the values of from this node to ceo;**

```
def total_influence(){
        result =0;
        Employee this is current_employee;
        while current_employee's id is not equal to 0
         if current.employee has not influenced
           set the reslut to result add current.employee's influence

         set current_employee to current.employee's boss

        return result
}
```

- **set_employee_and_bosses_as_parsed() if any employee has already influenced, set the flag in employee field to be true;**

```
def set_employee_and_bosses_as_parsed(){
        Employee current is this
        while current_employee's id is not equal to 0
          change current.employee has influenced

          set current_employee to current_employee.boss;
}
```

- **find_employee_with_id(id) according to given id find the employee**

```
def find_employee_with_id(id)
 Employee result is null;
  for each Employee in all employees
   if emplyee's id equal given id
     set result to the employee

  return result
```

Graph Class

- **add_employee(id, boss,influence) add the employee with given id, boss_id and its influence**

```
def add_employee(id, boss, influence)
```

```
 Employee boss;
if boss's id euqal to 0
   set boss to the first element of all employee in Graph
else
   set boss to Employee by calling find_employee_with_id(id)
if boss is null
    error
create a new Employee with id, influence, boss
add this Employee to all employees
```

- **optimal_total_influence(k) find optimal solution of total influence if you can influence k employees**

```
def optimal_total_influence(k)
   optimal_reslut =0;
   while k>0
    set the optimal_result to optiaml_total_influence add
        the queue.heap_extract_max() (the update max in queue)

   return optiaml_reslut;
```

- **process_graph() build the graph**

```
def process_graph()
   for each Employee in all employees
     if employee.has_sub is false
        insert the employee to queue
```

- **main function (run the problem by calling main function)**

```
def main function
  Grpah g;
  add each Employee with given id, boss_id, and influence in input field to graph
  g.process_graph();
  print the optimal_total_influence(k) of the g
```