## Part A

Question 1:

1.
On average advertiser “i” receives r(i)*CTR(i) value for each user that sees the ad. This is his truthful valuation. If all advertizers initially bid thruthfully, different slots may end up having different cost-per-click. This encourages bidders to receive the spot with lowest cost-per-click. This entails adjusting the bid so that the bidder receives the spot with the lowest price of ad impression.

In a simple example (lecture slides) with 3 bidders and 2 slots: if first slot has CTR of 100 and is awared to bidder who pays 8$, the CPC is 0.08$/click. The second spot has CTR of 80 and is awared to a bidder who pays 5$. The CPC is 0.0625$/click. If a third bidder enter the auction (with 10 valuation of top spot) his payoff is 200$ [(10$-8$)*100] if he bids thruthfully, and 400$ [(10$-5$)*80] if he bids not thruthfully e.g. 6$. The assumption is of course that this bidder know bids of other bidders so this effect can only be observed when information is public and at least one auction is condcted.

2.
Second price auction is a truth-telling strategy:
In case a bidder wins by bidding his true value v (i.e. v>b2 - second highest bid), he gains nothing from bidding less than v but more than b2. He still wins and gets v-b2 as payoff. If he bids below b2, he losses the auction (so incurs 0 gain).  There is no incentive to do so since when he wins he gets v-b2>0 payoff. As a result, there is no incentive for him to deviate from truthtelling strategy in case he wins.
In case a bidder losses by bidding his true value (i.e. v<b2), bidding any value below b2 has no effect, since he still loses and receives 0 utility. If he bids above b2 he wins, but receives v-b2<0 utility so he is worse off. As a result, if bidder loses by his thruthtelling bid there is no incentive from him to deviate from this truthtelling strategy.
In conclusion, in both cases, win or lose, a bidder has no incentive to deviate from his truthtelling strategy, what means it is his dominant strategy – a strategy that he choses no matter the stategy of other bidders.

Question 2:

To run the code:

```
git clone https://github.com/blackwhitehere/webEcon
cd partA
python DriverSSA.py
```

Below is only the VCG function. Other parts of the code were also modified.
The method calculates “lost CTR” for each slot (i.e. difference in CTR to the next bidder if a bidder occupying the slot was removed) which is multiplied by CPC of each bidder (in fact the CPC of the next bidder, see VCG formula).

```python
def executeVCG(self, slots):
    for key, slot in enumerate(slots):
        # assign slot winners according to sorted order of bidders
        if key < len(self.bids):
            slot.bidder = self.bids[key].name
            slot.win_bid = self.bids[key].value
        # calculate lost ctr for all slots with successive slot
        if key < len(slots) - 1:
            slot.calc_lost_ctr(slots[key + 1].clickThruRate)
    for key, bid in enumerate(self.bids):
        # calculate cpc for all bidders who got a slot:
        if key < len(slots) - 1:
            bid.calc_cpc(slots[key+1].win_bid, slots[key + 1].clickThruRate)
    # in case there are more slots than bids:
    if len(slots) > len(self.bids):
        # set lost_ctr of last bidder to 0, since he did not harm anybody
        slots[len(self.bids)-1].lost_ctr = 0.0
        # last bidder does not have a bidder cpc that can be used to compute vcg price
        self.bids[len(self.bids)-1].cpc = 0.0
    # in case there are more/equal bidders than slots:
    if len(slots) < len(self.bids):
        # set lost_ctr of last slot to ctr of last slot
        slots[len(slots)-1].lost_ctr = slots[len(slots)-1].clickThruRate
        # last successful bidder has cpc that of a first bidder who did no get a spot
        self.bids[len(slots)-1].calc_cpc(self.bids[len(slots)].value, slots[len(slots)-
1].clickThruRate)
    price_pass = [slot.lost_ctr*self.bids[key].cpc for key, slot in enumerate(slots) if key <
len(self.bids)]
    for i in range(len(price_pass)):
        slots[i].set_price(price_pass[i])
    for key, slot in reversed(list(enumerate(slots))):
        # accumulate vcg price from bottom to top slots
        if key < len(slots) - 1:
            slot.price += slots[key + 1].price
    for slot in slots:
        if slot.win_bid >= slot.price:
            slot.profit = slot.price
```