Part 1
Here is the code with the optimal solution and below that I will paste the rest of the code!

```
In [205]: print(LpStatus[prob.status])
          for v in prob.variables():
              if v.varValue > 0:
                  print(f"{v.name} = {v.varValue:.2f}")
          print(f"Cost: {value(prob.objective)}")

          Optimal
          amounts_Celery,_Raw = 48.33
          amounts_Frozen_Broccoli = 0.08
          amounts_Lettuce,Iceberg,Raw = 76.91
          amounts_Oranges = 2.73
          amounts_Poached_Eggs = 1.13
          amounts_Popcorn,Air_Popped = 13.71
          Cost: 4.532695045840001
```

```
In [190]: datalist = data.values.tolist()
          foods = {x[0] for x in datalist}
          price = {x[0]:x[1] for x in datalist}
          calories = {x[0]:x[3] for x in datalist}
          cholesterol = {x[0]:x[4] for x in datalist}
          fat = {x[0]:x[5] for x in datalist}
          sodium = {x[0]:x[6] for x in datalist}
          carbs = {x[0]:x[7] for x in datalist}
          fiber = {x[0]:x[8] for x in datalist}
          protein = {x[0]:x[9] for x in datalist}
          vita = {x[0]:x[10] for x in datalist}
          vitc = {x[0]:x[11] for x in datalist}
          calc = {x[0]:x[12] for x in datalist}
          iron = {x[0]:x[13] for x in datalist}

In [191]: lp_vars = LpVariable.dicts("amounts",foods,0)

In [192]: prob = LpProblem('CandyProblem',LpMinimize)

In [193]: #iron

In [202]: prob += lpSum(price[i]*lp_vars[i] for i in foods)
          prob += 1500 >= lpSum(calories[i]*lp_vars[i] for i in foods) <= 2500
          prob += 30 >= lpSum(cholesterol[i]*lp_vars[i] for i in foods) >= 240
          prob += 20 >= lpSum(fat[i]*lp_vars[i] for i in foods) <= 70
          prob += 800 >= lpSum(sodium[i]*lp_vars[i] for i in foods) <= 2000
          prob += 130 >= lpSum(carbs[i]*lp_vars[i] for i in foods) <= 450
          prob += 125 >= lpSum(fiber[i]*lp_vars[i] for i in foods) <= 250
          prob += 60 >= lpSum(protein[i]*lp_vars[i] for i in foods) <= 100
          prob += 1000 >= lpSum(vita[i]*lp_vars[i] for i in foods) <= 10000
          prob += 400 >= lpSum(vitc[i]*lp_vars[i] for i in foods) <= 5000
          prob += 700 >= lpSum(calc[i]*lp_vars[i] for i in foods) <= 1500
          prob += 10 >= lpSum(iron[i]*lp_vars[i] for i in foods) <= 40

In [203]: soln = prob.solve()
```

Part 2

FOr this part, I basically wrote a line of code to ensure that amount * price was greater than the price *.1. This works because the price of each item is the price per serving size. So if (amount)*(price) is greater than (price)*.1, you are essentially ensuring that you have greater than .1 serving size

Below is the output of the code showing that it works. Below that is the full code. The last line is what adds the new constraint

```
In [34]: print(LpStatus[prob.status])
         for v in prob.variables():
             if v.varValue > 0:
                 print(f"{v.name} = {v.varValue:.2f}")
         print(f"Cost: {value(prob.objective)}")

         Optimal
         amounts_Cheerios = 5.98
         amounts_Grapes = 1.02
         amounts_Poached_Eggs = 1.13
         Cost: 2.094000004
```

```
]: prob += lpSum(price[i]*lp_vars[i] for i in foods)
   prob += 1500 >= lpSum(calories[i]*lp_vars[i] for i in foods) <= 2500
   prob += 30 >= lpSum(cholesterol[i]*lp_vars[i] for i in foods) >= 240
   prob += 20 >= lpSum(fat[i]*lp_vars[i] for i in foods) <= 70
   prob += 800 >= lpSum(sodium[i]*lp_vars[i] for i in foods) <= 2000
   prob += 130 >= lpSum(carbs[i]*lp_vars[i] for i in foods) <= 450
   prob += 125 >= lpSum(fiber[i]*lp_vars[i] for i in foods) <= 250
   prob += 60 >= lpSum(protein[i]*lp_vars[i] for i in foods) <= 100
   prob += 1000 >= lpSum(vita[i]*lp_vars[i] for i in foods) <= 10000
   prob += 400 >= lpSum(vitc[i]*lp_vars[i] for i in foods) <= 5000
   prob += 700 >= lpSum(calc[i]*lp_vars[i] for i in foods) <= 1500
   prob += 10 >= lpSum(iron[i]*lp_vars[i] for i in foods) <= 40
   prob += lpSum(lp_vars[i]*price[i] for i in foods) >= lpSum(price[i]*.1 for i in foods)
```

Part 3

I wrote some code so that the (celery)*(brocoli)=0 and then celery+brocoli >1

```
In [9]: *lp_vars[i] for i in foods)
        calories[i]*lp_vars[i] for i in foods) <= 2500
        olesterol[i]*lp_vars[i] for i in foods) >= 240
        t[i]*lp_vars[i] for i in foods) <= 70
        odium[i]*lp_vars[i] for i in foods) <= 2000
        arbs[i]*lp_vars[i] for i in foods) <= 450
        iber[i]*lp_vars[i] for i in foods) <= 250
        otein[i]*lp_vars[i] for i in foods) <= 100
        vita[i]*lp_vars[i] for i in foods) <= 10000
        itc[i]*lp_vars[i] for i in foods) <= 5000
        alc[i]*lp_vars[i] for i in foods) <= 1500
        on[i]*lp_vars[i] for i in foods) <= 40
        'Frozen Broccoli'])^lpSum(lp_vars['Celery, Raw'])*lpSum(lp_vars['Celery, Raw'])^lpSum(lp_vars['Frozen Broccoli'])==0
```