Top-20 Training Program
(Greedy Thinking)

Apply the greedy thinking discussed in class to solve the following problems.

Problem1: Min-Stop Journey

There are n gas stations s1,s2,….sn along a highway from Hyderabad to Mumbai. On a full tank of gas your car goes for M miles. Gas station s1 is in Pune , each gas stations si($2 \leq i \leq n$) is di $\leq$ M miles after the previous gas station si-1, and gas station sn is in Mumbai. Write an efficient function to find the minimum number of gas stops that must be taken when driving from Pune to Mumbai. What are the time and space complexities of your solution?

Function Prototype:

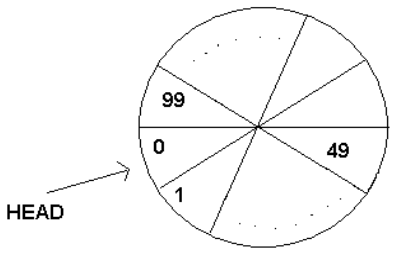
```
int FindMinStops(int d[], int n, int m)
```

Problem2: Minimum distance read of sectors

Assume that there is a circular disk of 100 **sectors** (0 to 99). Given an array of sectors to be traversed, determine the minimum distance that the head should travel in order to traverse all the sectors. The starting position of the head is always assumed to be 0.After beginning traversal, a maximum of one reversal in direction is allowed for the head.

Function Prototype:

```
int minSectorTraversal(int []sector_to_read, int n)
```



Example:

Sectors to be traversed: 50, 80, 70
Return value should be : 50
(i.e. the minimum traversal will be 0 - 80- 70 - 50 which adds up to 50)

Sectors to be traversed: 10, 70, 60
Return value should be : 60
(i.e. the minimum traversal will be 0 -10 - reverse - 70 - 60 which adds up to 60)

Top-20 Training Program
(Greedy Thinking)

Problem3: Minimum Amount spent to buy milk

The Merry Milk Makers company has several farmers from which they may buy milk, and each one has a (potentially) different price at which they sell to the milk packing plant. Moreover, as a cow can only produce so much milk a day, the farmers only have so much milk to sell per day. Each day, Merry Milk Makers can purchase an integral amount of milk from each farmer, less than or equal to the farmer's limit. Given the Merry Milk Makers' daily requirement of milk(totalneed), along with the cost per gallon(p[i]) and amount of available milk for each farmer(a[i]), write an efficient function to calculate the minimum amount of money that it takes to fulfill the Merry Milk Makers' requirements. Assume that the total milk produced per day by the farmers will be sufficient to meet the demands of the Merry Milk Makers and p[i] value will be between 0 and 1000. What are the time and space complexities of your solution?

Function Prototype:

```
int FindMinCost(int totalneed, int nfarmers, int *a, int *p)
```

Example:

Input: totalneed=100 nfarmers=5

```
a[] = 20 40 10 80 30
p[] =  5  9  3  8  6
```

Output: 630

Problem4: Maximize the tip

A new coffee place opened on the campus of ISB. Things work differently at this coffee place. Every customer arrives at time 0, and the owner decides the order in which to serve them. A single customer is served each second, until all the customers have been served. The first customer is served at second 0. If customer i is served at time t seconds, he will pay tips[i] - t. If this number is negative, he will not pay a tip at all. Given an array of tips, write an efficient function to return the maximum amount of money the owner can make in tips. What are the time and space complexities of your solution?

Function Prototype:

```
int getMaxMoney(int tips[], int n)
```

Problem5: Help the student to answer test

A student taking a test uses greedy strategies to maximize the test results. The input to this problem consists of the test time K in minutes, and the (estimated) times to solve each of the n questions on the test, $T[1..n]$, where time $T[i] > 0$ is the time in minutes for solving question i, $1 \le i \le n$. We assume no partial credits will be given; thus a completed test question gets full credits while incomplete answers get 0 credits. We also

Top-20 Training Program
(Greedy Thinking)

assume that $\sum_{i=1}^{n} T[i] > K$ so that there is not enough time to solve all test questions. There are two greedy strategies being considered:

a. Maximize the total number of completed test questions by working on the longest test question first, then the second longest, etc. Give a "small" example (i.e., $n \leq 4$) to show that this strategy is not optimal.
b. Maximize the total amount of time the student uses for the completed test questions, by working on the shortest test question first, then the second shortest, etc. (Thus, a student could be busy but receives no credits if no test questions are completed.) Does this algorithm (always) generate optimal solutions? Use an example to illustrate.