LAB 6

Submitted by:
Sushant Bansal 1410110454
Pragya Chaturvedi 1410110289

Problem Statement:

Suppose you were to drive from point A to point B. Your gas tank with a capacity C, when full, holds enough gas to travel m miles. You have a precise map that gives distances between gas stations along the route. Let d1 < d2 < .... < dn be the locations of all the gas stations along the route where di is the distance from point A to the gas station. You can assume that the distance between neighboring gas stations is at most m miles.

In the case that the rate at which you can fill your tank at a gas station is r (in liters/minute), so if you stop to fill your tank from 2 liters to 8 liters, you would have to stop for 6/r minutes. Give the most efficient greedy solution, where you need to minimize the total time you stop for gas filling?

Input:

Total Distance = 100
No of Stations = 8
Tank Capacity = 10 l
r = 1 l/m
m = 20

Output:

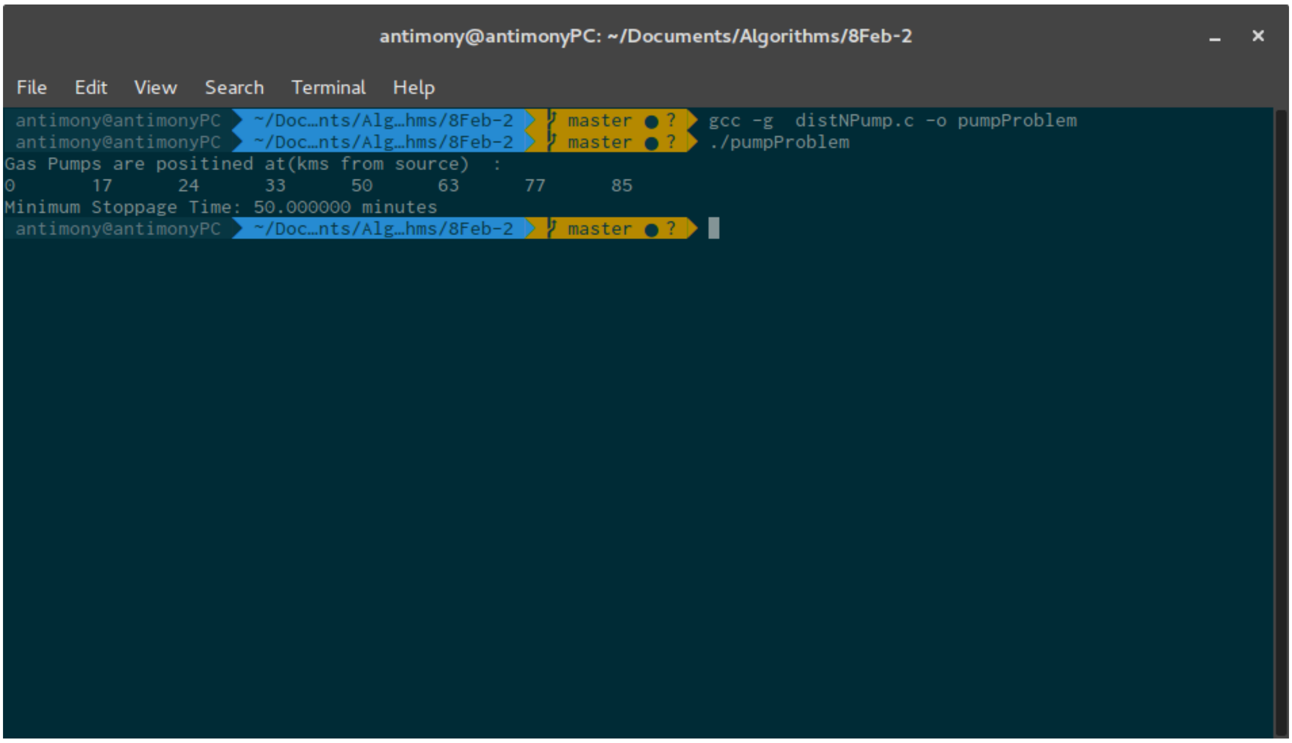
Minimum Stoppage Time: 50 minutes

Code:

```
/*
 * Made by Sushant Bansal 1410110454 and Pragya Chaturvedi 1410110289
 * 8th Feb, LAB 6
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

void initiate(int *arr, int num){
	for(int i=0;i<num;i++){
		arr[i] = -1;
	}
}

int returnSum(int *arr, int end){
	int sum = 0;
```

```
    for(int i=0;i<end;i++){
        sum += arr[i];
    }
    return sum;
}

void randomizePumps(int *arr, int num, int maxDist, int totalDistance){
    srand(time(NULL));
    arr[0]=0;
    int tmp;
    int chck = 1,i;
    while(chck == 1){
        for(i=1;i<num;i++){
            tmp = rand()%(maxDist+1);
            if(tmp == 0)
                tmp++;
            arr[i]= tmp + arr[i-1];
        }
        if(totalDistance-arr[num-1]<maxDist)
            chck = 0;
    }
    if(arr[num-1]>100)
        arr[num-1]=100;
}

void printArray(int *arr, int num){
    for(int i=0;i<num;i++)
        printf("%d \t",arr[i]);
    printf("\n");
}

int main(){
    int distance = 100,
    maxDist=20 ,
    capacity=10 ,
    noOfStations=8 ,
    rateOfFilling=1;
    int pumpDistance[noOfStations];

    initiate(pumpDistance,noOfStations);
    randomizePumps(pumpDistance,noOfStations,maxDist,distance);
    printf("Gas Pumps are positined at(kms from source)  : \n");
    printArray(pumpDistance,noOfStations);

    int k;
    int current_distance=distance - pumpDistance[0];
    int tempD;
    int diffBwPumps; double totalStoppageTime = 0;
    int i=0,j=0;
    while(current_distance>maxDist){
        current_distance = distance - pumpDistance[i];
        // printf(" Current Dist: %d \n",distance-current_distance );
        for(j=i+1;j<noOfStations;j++){
            tempD = pumpDistance[j]-pumpDistance[i];
            if(tempD<=maxDist){
                continue;
            }
            else
                break;
        }
        // printf("B Stoppage Time: %f \n",totalStoppageTime);

        if(pumpDistance[i]+maxDist >= distance){
            totalStoppageTime += (double)(distance -
pumpDistance[i])/2;
            // printf("if Stoppage Time: %f \n",(double)(distance -
pumpDistance[i]));
        }
        else{
            totalStoppageTime += (double)(pumpDistance[j-1]-
pumpDistance[i])/2;
            // printf("else Stoppage Time: %f \n",(double)
(pumpDistance[j-1]-pumpDistance[i])/2);
        }
        i = j-1;
    }

    printf("Maximum Stoppage Time: %f minutes\n",totalStoppageTime );
}
```

Screenshot:



```
antimony@antimonyPC  ~/Doc…nts/Alg…hms/8Feb-2  master ● ?  gcc -g  distNPump.c -o pumpProblem
antimony@antimonyPC  ~/Doc…nts/Alg…hms/8Feb-2  master ● ?  ./pumpProblem
Gas Pumps are positined at(kms from source)  :
0       17      24      33      50      63      77      85
Minimum Stoppage Time: 50.000000 minutes
antimony@antimonyPC  ~/Doc…nts/Alg…hms/8Feb-2  master ● ? 
```