DESIGN ANALYSIS AND ALGORITHM

LAB 4

GREEDY ALGORITHM

NAME: NAOMI GEORGE

SLOT: L25+L26+L33+L34+L13+L14

REGISTRATION NO. : 19BCE7572

COURSE CODE: CSE3004

CODE:

```
( CAR FUEL PROBLEM)
#include <iostream>
#include <vector>
using namespace std;

int MinR( int n, int milesAway, vector<int> Stops, int fulltank) {
   int numR = 0;
   int currentR = 0;
   int lastR = 0;

   if ((Stops[currentR] + fulltank) >= milesAway) {
      return numR;
   }

    while (currentR < n) {
      lastR = currentR;
      while ( ( currentR < n ) && ( (Stops[currentR + 1] - Stops[lastR]) <= fulltank ) )
      {
         currentR = currentR + 1;
      }

      cout << currentR << "  " << Stops[currentR] << "\n"; //printing to check

      if (currentR== lastR)
      {
         return -1;
      }
      numR = numR + 1;

      if ((Stops[currentR] + fulltank) >= milesAway)
```

```
        {
            return numR;
        }
    }
    return -1;
}

int main() {
    int milesAway, fulltank, n, stopValue;
    vector<int> Stops;
    cin >> milesAway;
    cin >> fulltank;
    cin >> n;
    Stops.push_back(0);
    if (n == 4) {
        int stop1, stop2, stop3, stop4;
        cin >> stop1 >> stop2 >> stop3 >> stop4;
        Stops.push_back(stop1);
        Stops.push_back(stop2);
        Stops.push_back(stop3);
        Stops.push_back(stop4);
    }
    else {
      for ( int i = 0; i < n; i++) {
         cin >> stopValue;
         Stops.push_back(stopValue);
      }
    }
    cout << MinR(n, milesAway, Stops, fulltank);
    return 0;
}
```

OUTPUT:

CODE:

(MAXIMUM SALARY)

```
#include <algorithm>
#include <sstream>
#include <iostream>
#include <vector>
#include <string>


using std::vector;
using std::string;

bool IsGreaterOrEqual(string digit, string maxDigit){

if (digit+maxDigit >=maxDigit +digit) {

return true;

}else{

return false;

}

}



string largest_number(vector<string> a) {
```

```
string result;
std::stringstream ret;
while (a.size()) {
string Maxdigit("0");
size_t index = 0;
for (size_t digit = 0; digit < a.size(); digit++) {

if (IsGreaterOrEqual(a[digit], Maxdigit)) {
Maxdigit = a[digit] ;
index = digit;
}
}
ret << Maxdigit;
a.erase(a.begin() + index);
}
ret >> result;
return result;
}


int main() {
int n;
std::cin >> n;
vector<string> a(n);
for (size_t i = 0; i < a.size(); i++) {
std::cin >> a[i];
```

```
}

std::cout << largest_number(a);

}
```

OUTPUT: