DESIGN ANALYSIS OF ALGORITHMS

ASSIGNMENT-4

Name: tarun

Reg.No: 19BCE7578

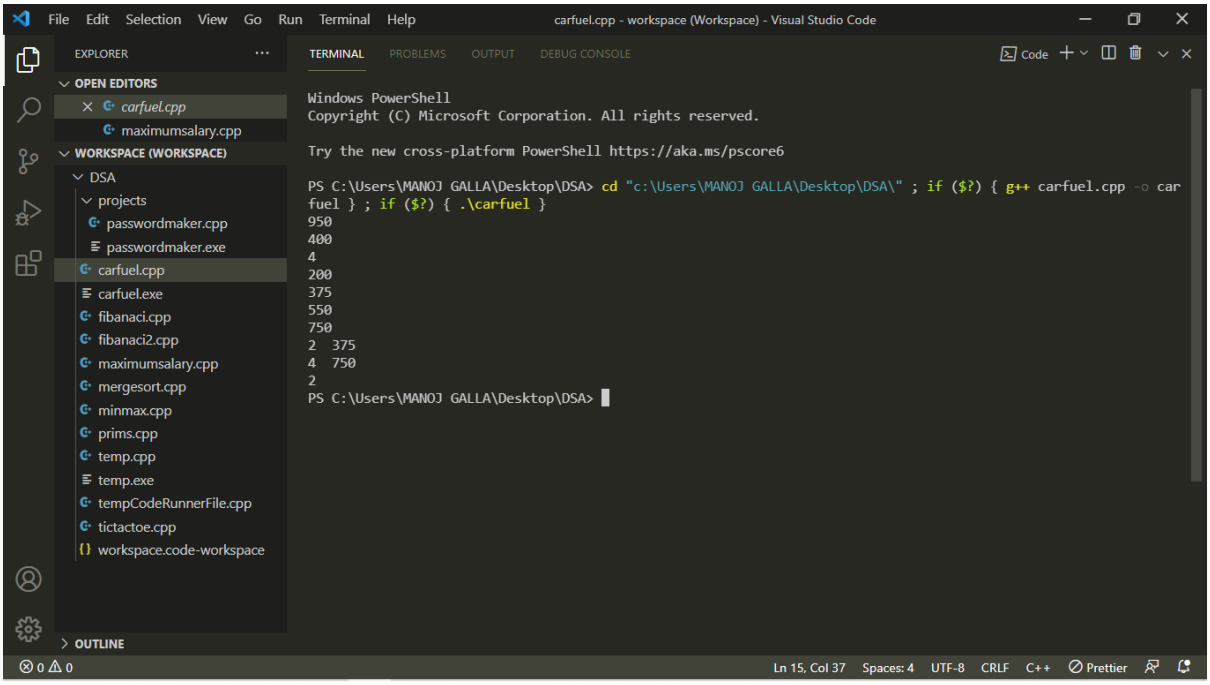
1. Carfuel

CODE:

```
#include <iostream>
#include <vector>
using namespace std;

int MinRefills( int n, int milesAway, vector<int> Stops, int fulltank)
{
    int numRefills = 0;
    int currentRefill = 0;
    int lastRefill = 0;

    if ((Stops[currentRefill] + fulltank) >= milesAway) {
        return numRefills;
    }

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

        cout << currentRefill << "   " << Stops[currentRefill] << "\n";

        if (currentRefill == lastRefill)
        {
```

```
            return -1;
        }
        numRefills = numRefills + 1;

        if ((Stops[currentRefill] + fulltank) >= milesAway)
        {
            return numRefills;
        }
    }
    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 << MinRefills(n, milesAway, Stops, fulltank)<<endl;
    return 0;
}
```

OUTPUT:



2. Maximum Salary

CODE:

```
#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;
```

```cpp
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: