**ITC Project**

#include <iostream>

using namespace std;

int main() {

int plate\_num[100] = { 21000, 10400, 11600, 23200, 33000, 33400, 4000, 4400, 5000, 53500, 6000, 6600, 7000, 733700, 8000, 8800, 9000, 9900, 9990, 9999};

char brand[100] = { 'C', 'A', 'B', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'Z', 'N', 'O', 'P', 'Q', 'R', 'S', 'T' };

float capacity[100] = { 1.3, 1.1, 1.2, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.1, 3.2 };

int price[100] = { 20000, 102000, 11000, 22000, 30000, 33000, 40000, 44000, 500030, 55000, 60000, 66000, 70000, 77000, 80000, 880030, 903000, 993000, 99900, 99990 };

int Number\_of\_cars = 20;

//plate number wise in ascending order

bool continue\_or\_not = false;

do{

cout << "The Menu: " << endl;

cout << "1. Sort and display all the records plate number wise in ascending order" << endl;

cout << "2. Sort and display all the records plate number wise in descending order." << endl;

cout << "3. Sort and display all records in ascending order based on capacity in Brand" << endl;

cout << "4. Sort and display all records in descending order based on capacity in Brand" << endl;

cout << "5. Sort and display all records in ascending order based on price in plate number" << endl;

cout << "6. Sort and display all records in descending order based on price in plate number" << endl;

cout << "7. Add a new entry of a customer." << endl;

cout << "8. Delete a customer record based on his plate number. " << endl;

cout << "9. Display record of all the customers greater than X price (in descending order) The value of X will be entered by the User. " << endl;

cout << "10. Display record of all the customers greater than X price (in ascending order). The value of X will be entered by the User. " << endl;

cout << "11. Display record of all the customers less than or equal to X price (in descending order). The value of X will be entered by the User." << endl;

cout << "12. Display record of all the customers less than or equal to X price (in ascending order). The value of X will be entered by the User." << endl << endl;

cout << "Enter the option number you want to display: " << endl;

int select\_option = 0;

cin >> select\_option;

int helper = 0;

char helper2;

float helper3 = 0.0;

if (select\_option == 1){

cout << "1. Sort and display all the records plate number wise in ascending order" << endl;

for (int index = 0; index < Number\_of\_cars; index++){

for (int loop = 0; loop < Number\_of\_cars; loop++){

if (plate\_num[index] < plate\_num[loop])

{

helper = plate\_num[index];

plate\_num[index] = plate\_num[loop];

plate\_num[loop] = helper;

helper2 = brand[index];

brand[index] = brand[loop];

brand[loop] = helper2;

helper3 = capacity[index];

capacity[index] = capacity[loop];

capacity[loop] = helper3;

helper = price[index];

price[index] = price[loop];

price[loop] = helper;

}

}

}

for (int print = 0; print < Number\_of\_cars; print++){

cout << plate\_num[print] << " " << brand[print] << " " << capacity[print] << " " << price[print] << endl;

}

}

//the records plate number wise in descending order.

else if (select\_option == 2){

cout << "2. Sort and display all the records plate number wise in descending order." << endl;

for (int index = 0; index < Number\_of\_cars; index++){

for (int loop = 0; loop < Number\_of\_cars; loop++){

if (plate\_num[index] > plate\_num[loop])

{

helper = plate\_num[index];

plate\_num[index] = plate\_num[loop];

plate\_num[loop] = helper;

helper2 = brand[index];

brand[index] = brand[loop];

brand[loop] = helper2;

helper3 = capacity[index];

capacity[index] = capacity[loop];

capacity[loop] = helper3;

helper = price[index];

price[index] = price[loop];

price[loop] = helper;

}

}

}

for (int print = 0; print < Number\_of\_cars; print++){

cout << plate\_num[print] << " " << brand[print] << " " << capacity[print] << " " << price[print] << endl;

}

}

//records in ascending order based on capacity in Brand

else if (select\_option == 3){

cout << "3. Sort and display all records in ascending order based on capacity in Brand" << endl;

for (int index = 0; index < Number\_of\_cars; index++){

for (int loop = 0; loop < Number\_of\_cars; loop++){

if (capacity[index] < capacity[loop])

{

helper = plate\_num[index];

plate\_num[index] = plate\_num[loop];

plate\_num[loop] = helper;

helper2 = brand[index];

brand[index] = brand[loop];

brand[loop] = helper2;

helper3 = capacity[index];

capacity[index] = capacity[loop];

capacity[loop] = helper3;

helper = price[index];

price[index] = price[loop];

price[loop] = helper;

}

}

}

for (int print = 0; print < Number\_of\_cars; print++){

cout << plate\_num[print] << " " << brand[print] << " " << capacity[print] << " " << price[print] << endl;

}

}

//all records in descending order based on capacity in Brand

else if (select\_option == 4){

cout << "4. Sort and display all records in descending order based on capacity in Brand" << endl;

for (int index = 0; index < Number\_of\_cars; index++){

for (int loop = 0; loop < Number\_of\_cars; loop++){

if (capacity[index] > capacity[loop])

{

helper = plate\_num[index];

plate\_num[index] = plate\_num[loop];

plate\_num[loop] = helper;

helper2 = brand[index];

brand[index] = brand[loop];

brand[loop] = helper2;

helper3 = capacity[index];

capacity[index] = capacity[loop];

capacity[loop] = helper3;

helper = price[index];

price[index] = price[loop];

price[loop] = helper;

}

}

}

for (int print = 0; print < Number\_of\_cars; print++){

cout << plate\_num[print] << " " << brand[print] << " " << capacity[print] << " " << price[print] << endl;

}

}

//all records in ascending order based on price in plate number

else if (select\_option == 5){

cout << "5. Sort and display all records in ascending order based on price in plate number" << endl;

for (int index = 0; index < Number\_of\_cars; index++){

for (int loop = 0; loop < Number\_of\_cars; loop++){

if (price[index] < price[loop])

{

helper = plate\_num[index];

plate\_num[index] = plate\_num[loop];

plate\_num[loop] = helper;

helper2 = brand[index];

brand[index] = brand[loop];

brand[loop] = helper2;

helper3 = capacity[index];

capacity[index] = capacity[loop];

capacity[loop] = helper3;

helper = price[index];

price[index] = price[loop];

price[loop] = helper;

}

}

}

for (int print = 0; print < Number\_of\_cars; print++){

cout << plate\_num[print] << " " << brand[print] << " " << capacity[print] << " " << price[print] << endl;

}

}

//Sort and display all records in descending order based on price in plate number

else if (select\_option == 6){

cout << "6. Sort and display all records in descending order based on price in plate number" << endl;

for (int index = 0; index < Number\_of\_cars; index++){

for (int loop = 0; loop < Number\_of\_cars; loop++){

if (price[index] > price[loop])

{

helper = plate\_num[index];

plate\_num[index] = plate\_num[loop];

plate\_num[loop] = helper;

helper2 = brand[index];

brand[index] = brand[loop];

brand[loop] = helper2;

helper3 = capacity[index];

capacity[index] = capacity[loop];

capacity[loop] = helper3;

helper = price[index];

price[index] = price[loop];

price[loop] = helper;

}

}

}

for (int print = 0; print < Number\_of\_cars; print++){

cout << plate\_num[print] << " " << brand[print] << " " << capacity[print] << " " << price[print] << endl;

}

}

//Add a new entry of a customer.

else if (select\_option == 7){

cout << "7. Add a new entry of a customer." << endl;

for (int print = 0; print < Number\_of\_cars; print++){

cout << plate\_num[print] << " " << brand[print] << " " << capacity[print] << " " << price[print] << endl;

}

cout << "\nEnter Plate Number:";

cin >> plate\_num[Number\_of\_cars];

cout << "\nEnter Brand:";

cin >> brand[Number\_of\_cars];

cout << "\nEnter Capacity:";

cin >> capacity[Number\_of\_cars];

cout << "\nEnter Price:";

cin >> price[Number\_of\_cars];

Number\_of\_cars++;

for (int print = 0; print < Number\_of\_cars; print++){

cout << plate\_num[print] << " " << brand[print] << " " << capacity[print] << " " << price[print] << endl;

}

}

//Delete a customer record based on his plate number.

else if (select\_option == 8){

for (int print = 0; print < Number\_of\_cars; print++){

cout << plate\_num[print] << " " << brand[print] << " " << capacity[print] << " " << price[print] << endl;

}

cout << "\nEnter Plate Number to Delete: ";

int plate\_num\_input;

cin >> plate\_num\_input;

bool found = false;

for (int index = 0; index < Number\_of\_cars; index++){

if (plate\_num[index] == plate\_num\_input){

for (int loop = index; loop < (Number\_of\_cars - 1); loop++){

plate\_num[loop] = plate\_num[loop + 1];

brand[loop] = brand[loop + 1];

capacity[loop] = capacity[loop + 1];

price[loop] = price[loop + 1];

found = true;

}

break;

}

}if (found){

cout << "\nCars Detail Deleted Successfully!";

}

else{

cout << "\nNot Found";

}

plate\_num[Number\_of\_cars] = 0;

brand[Number\_of\_cars] = '0';

capacity[Number\_of\_cars] = 0.0;

price[Number\_of\_cars] = 0;

Number\_of\_cars--;

}

//Display record of all the customers greater than X price (in descending order) The value of X will be entered by the User.

else if (select\_option == 9){

cout << "9. Display record of all the customers greater than X price (in descending order) The value of X will be entered by the User. " << endl;

for (int index = 0; index < Number\_of\_cars; index++){

for (int loop = 0; loop < Number\_of\_cars; loop++){

if (price[index] > price[loop])

{

helper = plate\_num[index];

plate\_num[index] = plate\_num[loop];

plate\_num[loop] = helper;

helper2 = brand[index];

brand[index] = brand[loop];

brand[loop] = helper2;

helper3 = capacity[index];

capacity[index] = capacity[loop];

capacity[loop] = helper3;

helper = price[index];

price[index] = price[loop];

price[loop] = helper;

}

}

}

int enter\_price = 0;

do{

cout << "\nEnter the price for print record greater then Enter price:";

cin >> enter\_price;

} while (enter\_price <= 0);

for (int index = 0; index < Number\_of\_cars; index++){

if (enter\_price < price[index]){

cout << plate\_num[index] << " " << brand[index] << " " << capacity[index] << " " << price[index] << endl;

}

}

}

//Display record of all the customers greater than X price (in ascending order). The value of X will be entered by the User.

else if (select\_option == 10){

cout << "10. Display record of all the customers greater than X price (in ascending order). The value of X will be entered by the User. " << endl;

for (int index = 0; index < Number\_of\_cars; index++){

for (int loop = 0; loop < Number\_of\_cars; loop++){

if (price[index] < price[loop])

{

helper = plate\_num[index];

plate\_num[index] = plate\_num[loop];

plate\_num[loop] = helper;

helper2 = brand[index];

brand[index] = brand[loop];

brand[loop] = helper2;

helper3 = capacity[index];

capacity[index] = capacity[loop];

capacity[loop] = helper3;

helper = price[index];

price[index] = price[loop];

price[loop] = helper;

}

}

}

int enter\_price = 0;

do{

cout << "\nEnter the price for print record greater then Enter price:";

cin >> enter\_price;

} while (enter\_price <= 0);

for (int index = 0; index < Number\_of\_cars; index++){

if (enter\_price < price[index]){

cout << plate\_num[index] << " " << brand[index] << " " << capacity[index] << " " << price[index] << endl;

}

}

}

//Display record of all the customers less than or equal to X price (in descending order). The value of X will be entered by the User

else if (select\_option == 11){

cout << "11. Display record of all the customers less than or equal to X price (in descending order). The value of X will be entered by the User." << endl;

for (int index = 0; index < Number\_of\_cars; index++){

for (int loop = 0; loop < Number\_of\_cars; loop++){

if (price[index] > price[loop])

{

helper = plate\_num[index];

plate\_num[index] = plate\_num[loop];

plate\_num[loop] = helper;

helper2 = brand[index];

brand[index] = brand[loop];

brand[loop] = helper2;

helper3 = capacity[index];

capacity[index] = capacity[loop];

capacity[loop] = helper3;

helper = price[index];

price[index] = price[loop];

price[loop] = helper;

}

}

}

int enter\_price = 0;

do{

cout << "\nEnter the price for print record greater then Enter price:";

cin >> enter\_price;

} while (enter\_price <= 0);

for (int index = 0; index < Number\_of\_cars; index++){

if (enter\_price >= price[index]){

cout << plate\_num[index] << " " << brand[index] << " " << capacity[index] << " " << price[index] << endl;

}

}

}

//Display record of all the customers less than or equal to X price (in ascending order). The value of X will be entered by the User

else if (select\_option == 12){

cout << "12. Display record of all the customers less than or equal to X price (in ascending order). The value of X will be entered by the User." << endl << endl;

for (int index = 0; index < Number\_of\_cars; index++){

for (int loop = 0; loop < Number\_of\_cars; loop++){

if (price[index] < price[loop])

{

helper = plate\_num[index];

plate\_num[index] = plate\_num[loop];

plate\_num[loop] = helper;

helper2 = brand[index];

brand[index] = brand[loop];

brand[loop] = helper2;

helper3 = capacity[index];

capacity[index] = capacity[loop];

capacity[loop] = helper3;

helper = price[index];

price[index] = price[loop];

price[loop] = helper;

}

}

}

int enter\_price = 0;

do{

cout << "\nEnter the price for print record greater then Enter price:";

cin >> enter\_price;

} while (enter\_price <= 0);

for (int index = 0; index < Number\_of\_cars; index++){

if (enter\_price >= price[index]){

cout << plate\_num[index] << " " << brand[index] << " " << capacity[index] << " " << price[index] << endl;

}

}

}

else{

cout << "\nWrong Option!";

}

cout << "\nPress Any Key for End Except '0' and '1' and 'Enter Botten':";

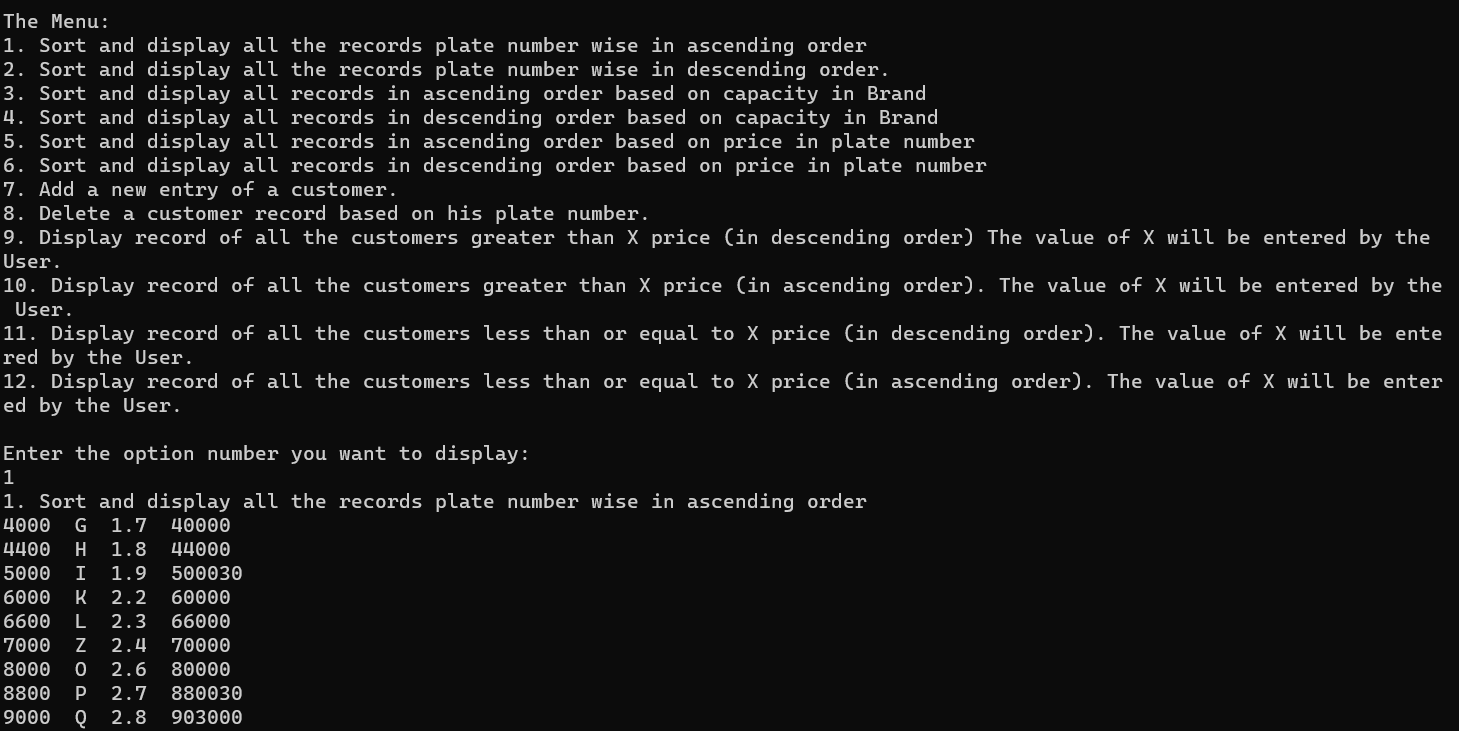
}while (cin >> continue\_or\_not);

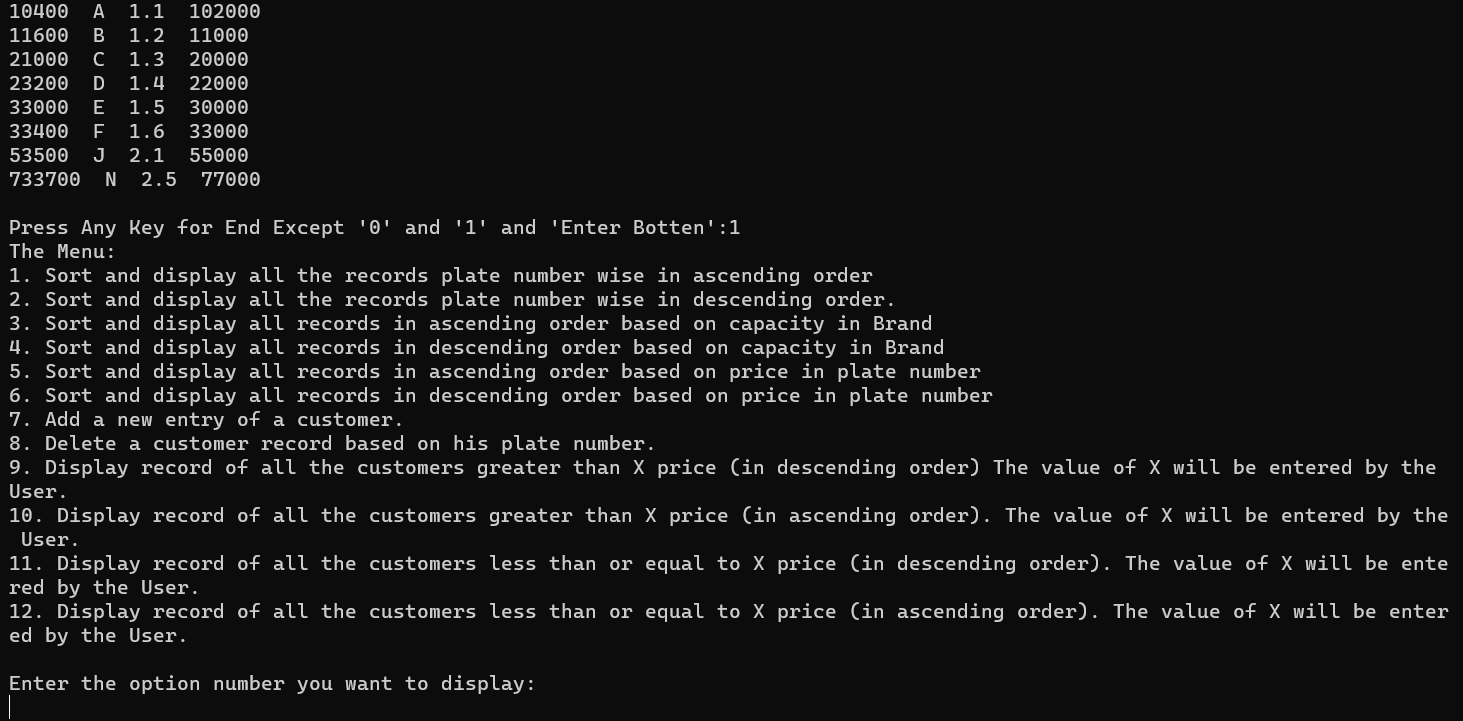
cout << '\n';

return 0;

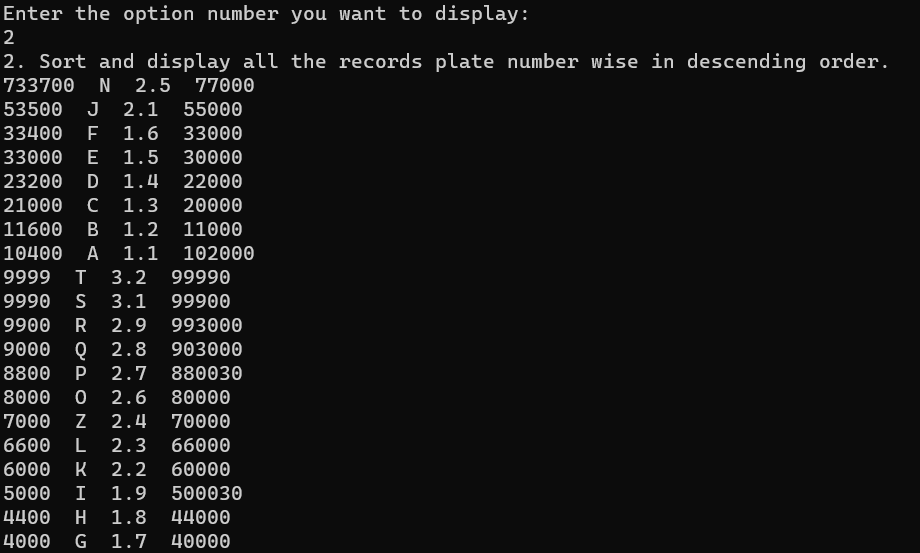
}

Output 1:

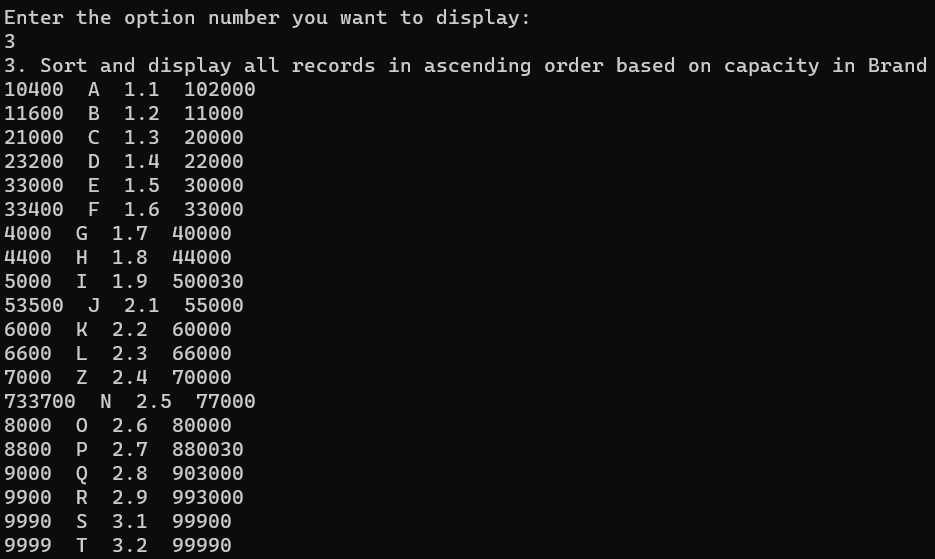




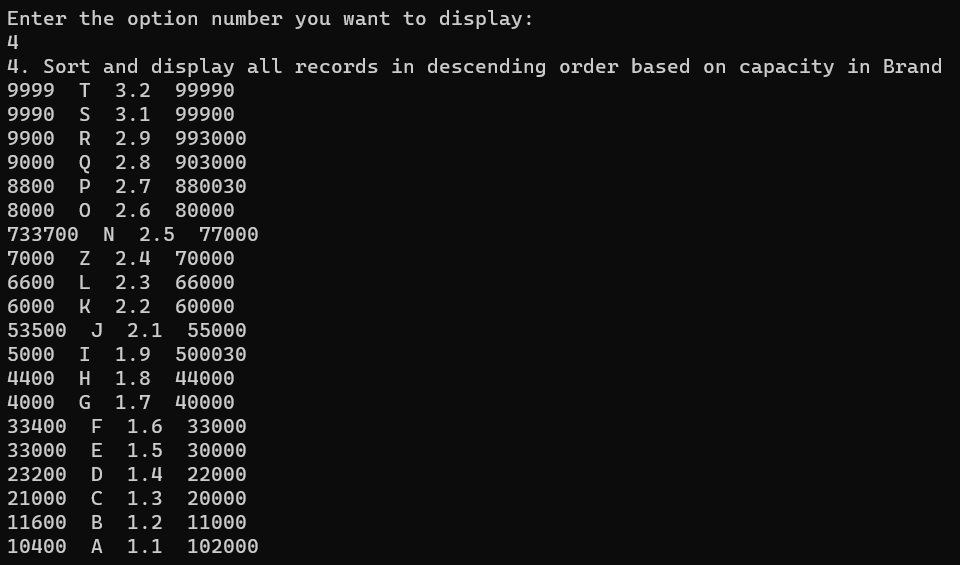
Output 2:



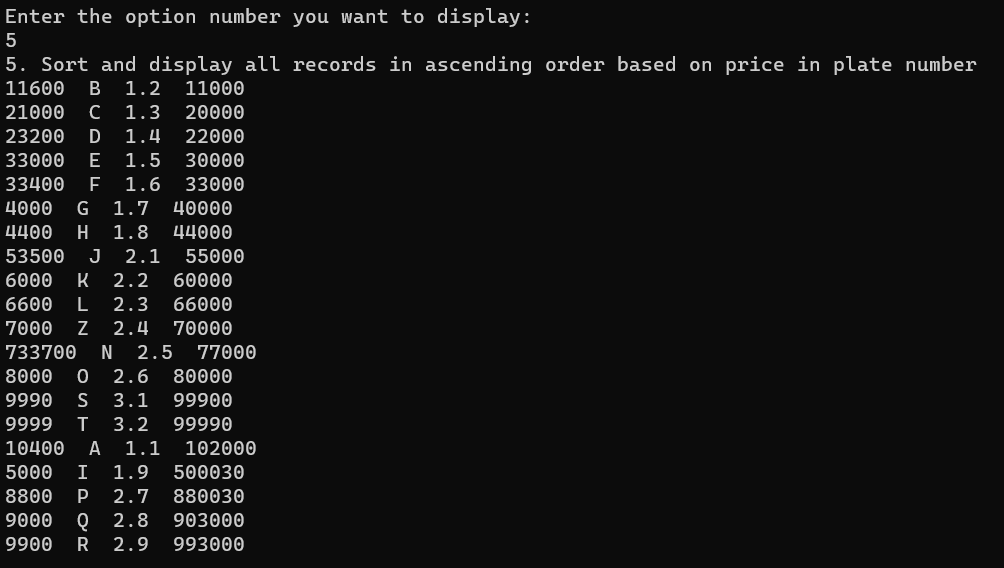
Output 3:



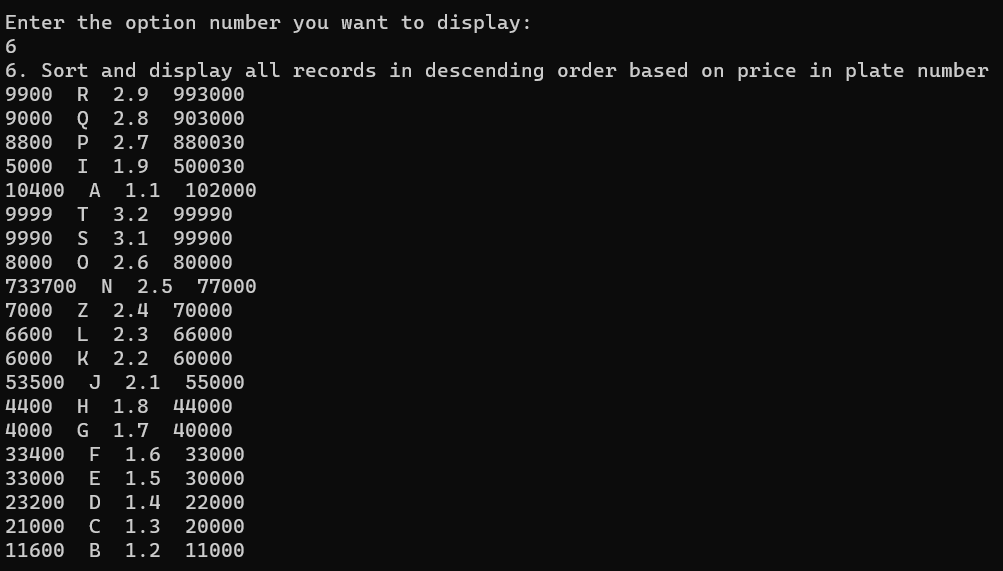
Output 4:



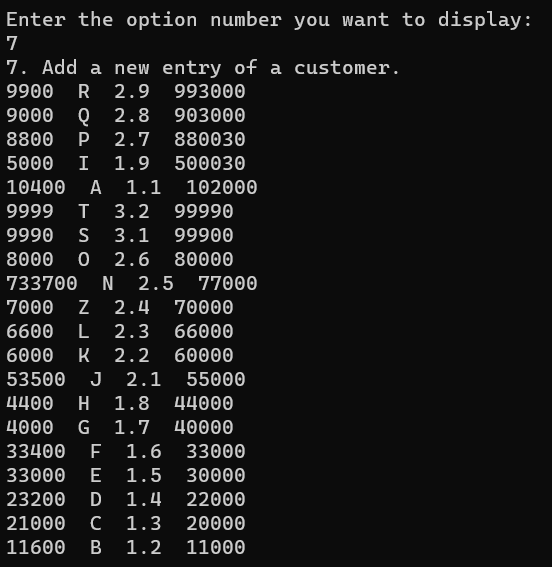
Output 5:

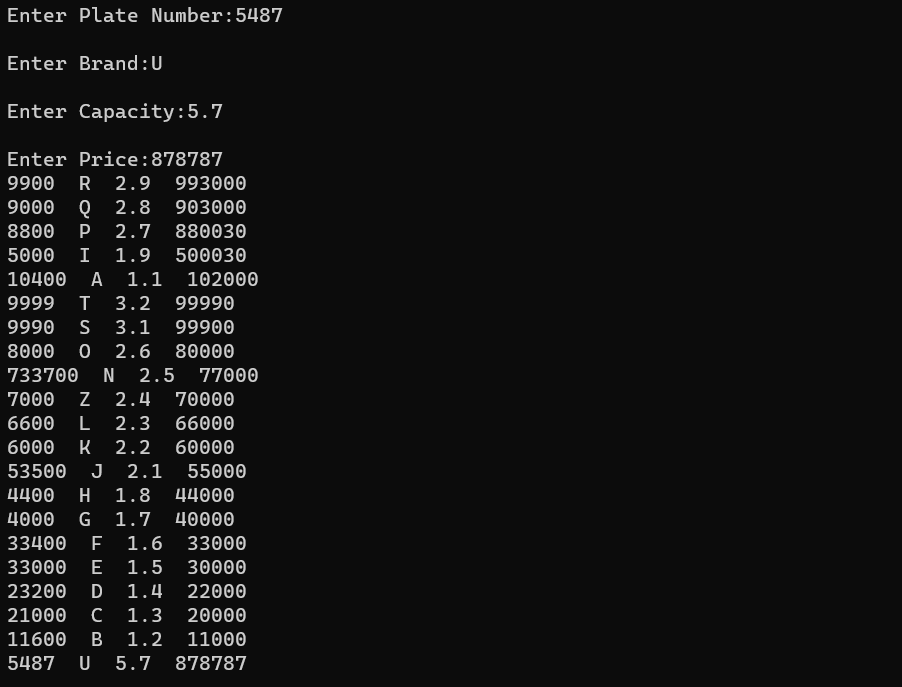


Output 6:

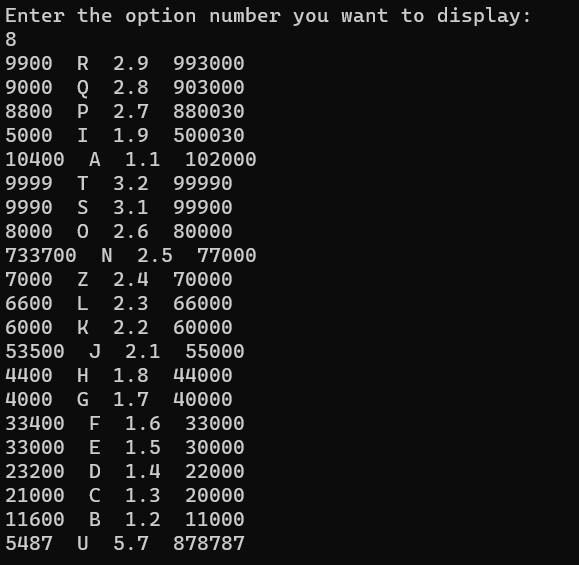


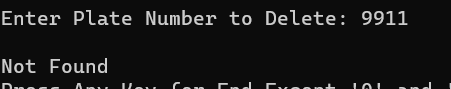
Output 7:

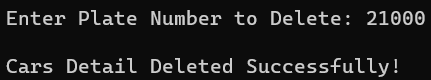




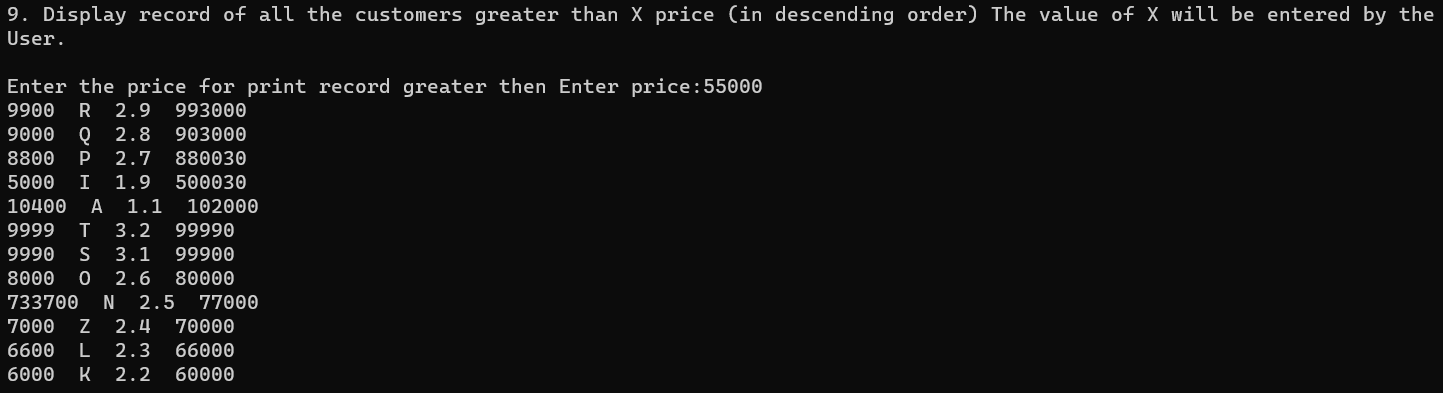
Output 8:



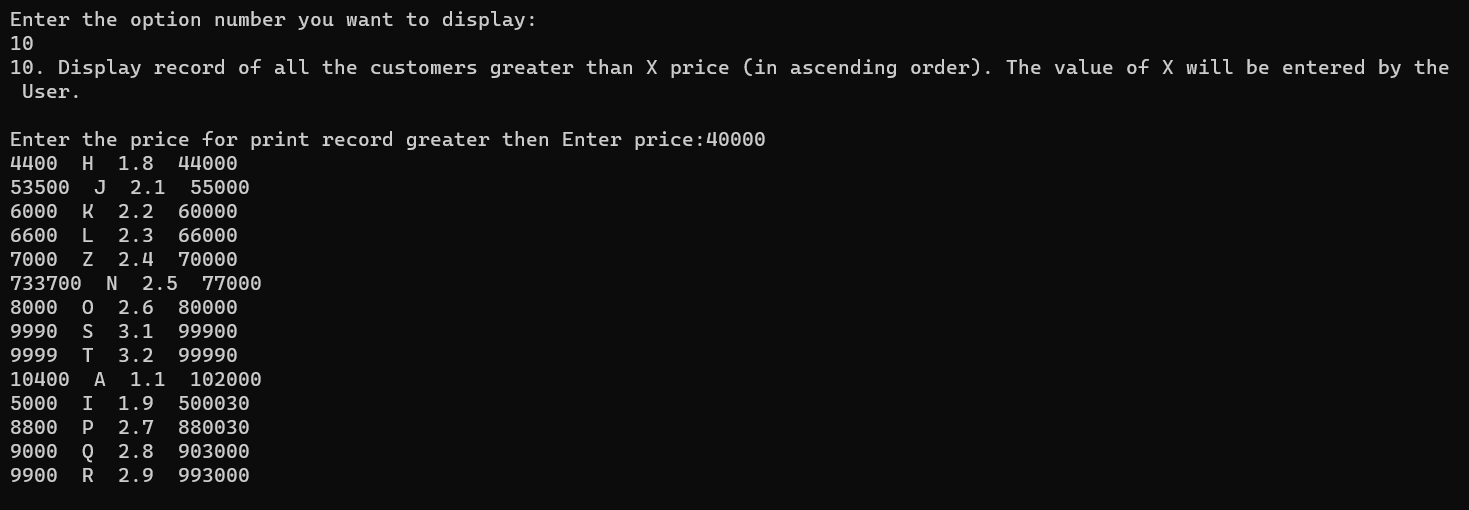




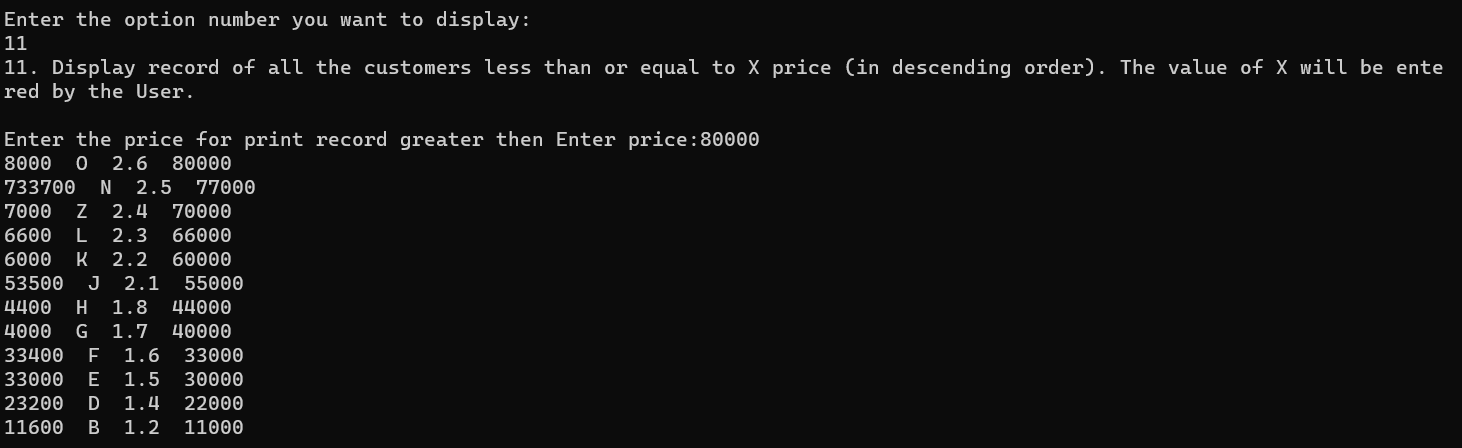
Output 9:



Output 10:



Output 11:



Output 12:

