**Structures and Pointers**

1. Implementation of Structures (Define a structure named Friends with members: name, pet name, phone number, and a nested structure named Type with members: Type of friend (School, College or Area friend), name of common friends and places visited together. Write a C program to input at least 3 friends data and display the details in proper time format.)

Code:

#include <stdio.h>

struct Type {

char typeOfFriend[20];

char commonFriends[50];

char placesVisited[50];

};

struct Friends {

char name[50];

char petName[20];

long phoneNumber;

struct Type friendType;

};

int main() {

struct Friends friends[3];

for (int i = 0; i < 3; i++) {

printf("Enter details for Friend %d:\n", i + 1);

printf("Name: ");

scanf("%s", friends[i].name);

printf("Pet Name: ");

scanf("%s", friends[i].petName);

printf("Phone Number: ");

scanf("%ld", &friends[i].phoneNumber);

printf("Type of Friend (School/College/Area): ");

scanf("%s", friends[i].friendType.typeOfFriend);

printf("Common Friends: ");

scanf("%s", friends[i].friendType.commonFriends);

printf("Places Visited Together: ");

scanf("%s", friends[i].friendType.placesVisited);

}

printf("\n--- Friends Details ---\n");

printf("Name\t\tPet Name\tPhone Number\tType\tCommon Friends\tPlaces Visited\n");

for (int i = 0; i < 3; i++) {

printf("%s\t%s\t%ld\t%s\t%s\t%s\n",friends[i].name,friends[i].petName,friends[i].phoneNumber,friends[i].friendType.typeOfFriend,friends[i].

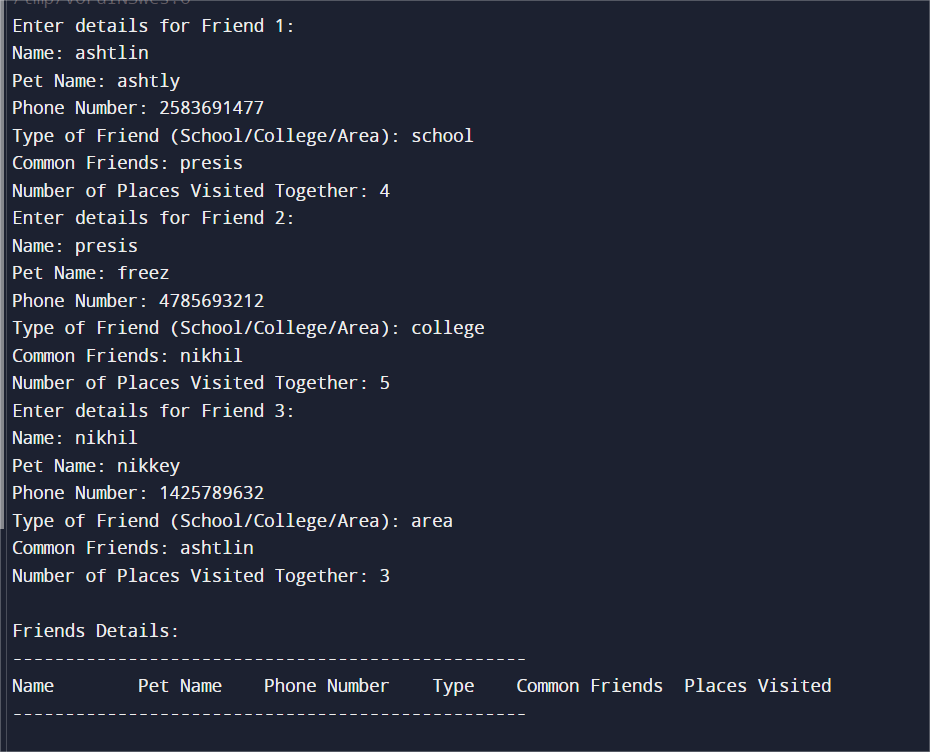
friendType.commonFriends, friends[i].friendType.placesVisited);

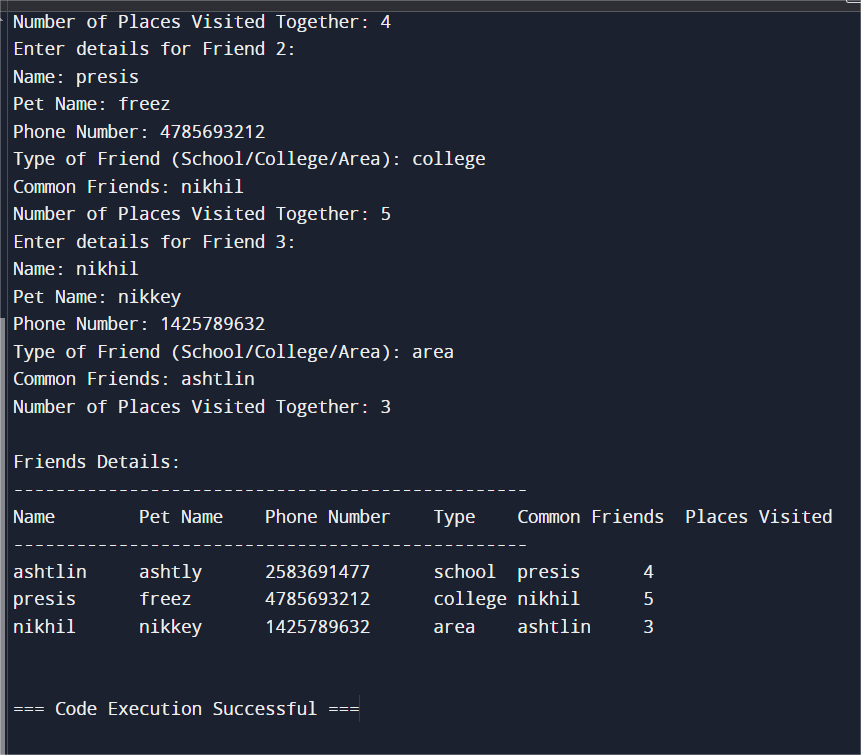
}

return 0;

}

Output:





1. Implementation of Structures using Pointers (Create a structure named Product to store details of the product like name, ID and price. Write a C program to input details for at least 5 products, find the Total cost of all products, the most expensive and the lowest priced product, and display their information.)

Code:

#include <stdio.h>

#include <stdlib.h>

struct Product

{

char name[50];

char id[10];

int price;

};

int main() {

int numProducts, totalCost = 0;

struct Product \*products;

printf("Enter the number of products: ");

scanf("%d", &numProducts);

products = (struct Product \*)malloc(numProducts \* sizeof(struct Product));

for (int i = 0; i < numProducts; i++) {

printf("Enter details for product %d:\n", i + 1);

printf("Product Name: ");

scanf("%s", products[i].name);

printf("Product ID: ");

scanf("%s", products[i].id);

printf("Price: ");

scanf("%d", &products[i].price);

totalCost += products[i].price;

}

printf("Product Details:\n");

for (int i = 0; i < numProducts; i++) {

printf("Product Name: %s, Product ID: %s, Price: %d\n", products[i].name, products[i].id, products[i].price);

}

int maxPrice = products[0].price, minPrice = products[0].price;

int maxIndex = 0, minIndex = 0;

for (int i = 1; i < numProducts; i++) {

if (products[i].price > maxPrice) {

maxPrice = products[i].price;

maxIndex = i;

}

if (products[i].price < minPrice) {

minPrice = products[i].price;

minIndex = i;

}

}

printf("Most Expensive Product: %s, Product ID: %s, Price: %d\n", products[maxIndex].name, products[maxIndex].id, products[maxIndex].price);

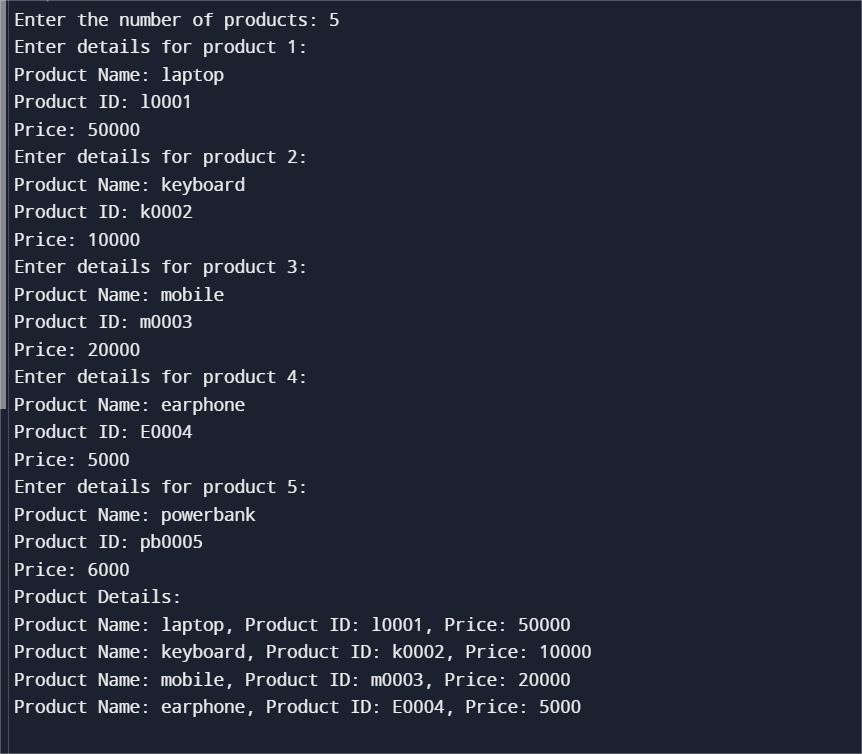
printf("Least Expensive Product: %s, Product ID: %s, Price: %d\n", products[minIndex].name, products[minIndex].id, products[minIndex].price);

printf("Total Cost of All Products: %d\n", totalCost);

free(products);

return 0;

}

Output:

My Github profile link: https://github.com/presisashtlin