**DATA STRUCTURES AND ALGORITHMS ASSIGNMENT**

**Name:sivagovindan**

**Reg no:RA2311003050377**

**Year & Branch: II B.Tech CSE - F Section**

**Assignment No: 1**

**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 atleast 3 friends data and display the

details in proper time format.)

**ANSWER:**

#include <stdio.h>

#include <string.h>

struct friends {

char name[50];

char pet\_name[50];

int phone\_number;

};

struct type {

char type\_of\_friends[50];

char common\_friends[50];

char places\_visited\_together[150];

struct friends fr;

};

int main() {

int n;

printf("Number of friends: ");

scanf("%d", &n);

struct type t[n];

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

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

printf("Enter your name: ");

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

printf("Enter your pet's name: ");

scanf("%s", t[i].fr.pet\_name);

printf("Enter your phone number: ");

scanf("%d", &t[i].fr.phone\_number);

printf("Enter common friends: ");

scanf("%s", t[i].common\_friends);

printf("Enter type of friends: ");

scanf("%s", t[i].type\_of\_friends);

printf("Enter places visited together: ");

scanf("%s", t[i].places\_visited\_together);

printf("\n");

}

printf("Details of friends:\n");

printf("%-20s %-20s %-15s %-30s %-20s %-30s\n",

"Name", "Pet Name", "Phone Number", "Common Friends", "Type of Friends", "Places Visited Together");

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

printf("%-20s %-20s %-15d %-30s %-20s %-30s\n",

t[i].fr.name,

t[i].fr.pet\_name,

t[i].fr.phone\_number,

t[i].common\_friends,

t[i].type\_of\_friends,

t[i].places\_visited\_together);

}

FILE \*file = fopen("friends\_data.csv", "w");

if (file == NULL) {

printf("\n");

return 1;

}

fprintf(file, "Name,Pet Name,Phone Number,Common Friends,Type of Friends,Places Visited Together\n");

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

fprintf(file, "%s,%s,%d,%s,%s,%s\n",

t[i].fr.name,

t[i].fr.pet\_name,

t[i].fr.phone\_number,

t[i].common\_friends,

t[i].type\_of\_friends,

t[i].places\_visited\_together);

}

fclose(file);

printf("Data successfully written to friends\_data.csv\n");

return 0;

}

**OUTPUT:**

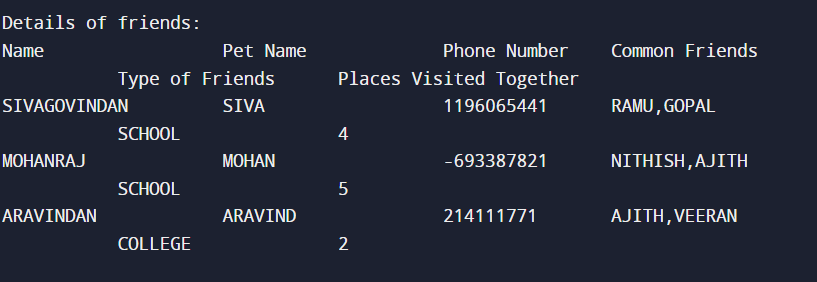
Details of friends:

Name Pet Name Phone Number Common Friends Type of Friends Places Visited Together

SIVAGOVINDAN SIVA 1196065441 RAMU,GOPAL SCHOOL 4

MOHANRAJ MOHAN -693387821 NITHISH,AJITH SCHOOL 5

ARAVINDAN ARAVIND 214111771 AJITH,VEERAN COLLEGE 2



2. 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.)

**ANSWER:**

#include <stdio.h>

#include <string.h>

struct product {

char name[150];

char id[150];

float price;

};

int main() {

int n;

float totalcost = 0.0;

struct product\* mostexpensive;

struct product\* lowestpriced;

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

scanf("%d", &n);

struct product p[n];

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

printf("Product %d:\n", i + 1);

printf("Product name: ");

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

printf("Product ID: ");

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

printf("Price: ");

scanf("%f", &p[i].price);

totalcost += p[i].price;

if (i == 0) {

mostexpensive = &p[i];

lowestpriced = &p[i];

} else {

if (p[i].price > mostexpensive->price) {

mostexpensive = &p[i];

}

if (p[i].price < lowestpriced->price) {

lowestpriced = &p[i];

}

}

printf("\n");

}

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

printf("Product %d details:\n", i + 1);

printf("Product name: %s\n", p[i].name);

printf("Product ID: %s\n", p[i].id);

printf("Price: %.2f\n", p[i].price);

printf("\n");

}

printf("Total cost: %.2f\n", totalcost);

printf("Most expensive product: %s (%.2f)\n", mostexpensive->name, mostexpensive->price);

printf("Lowest priced product: %s (%.2f)\n", lowestpriced->name, lowestpriced->price);

return 0;

}

**OUTPUT:**

Product 1 details:

Product name: LAPTOP

Product ID: P001

Price: 66000.00

Product 2 details:

Product name: MOBILE

Product ID: PH02

Price: 20000.00

Product 3 details:

Product name: IPAD

Product ID: IP03

Price: 33000.00

Product 4 details:

Product name: CHARGER

Product ID: CH04

Price: 1000.00

Product 5 details:

Product name: HEADPHONES

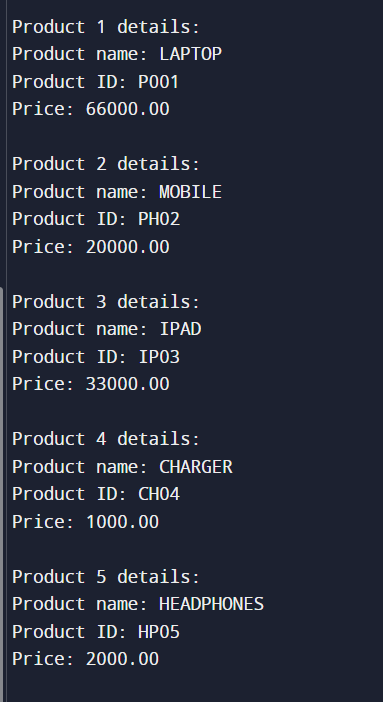
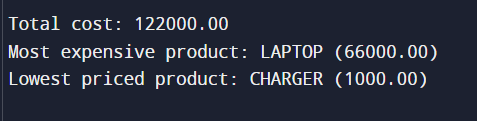
Product ID: HP05

Price: 2000.00

Total cost: 122000.00

Most expensive product: LAPTOP (66000.00)

Lowest priced product: CHARGER (1000.00)

**  
**