8/9/2024

DATA STRUCTURE AND ALGORITHM

ASSIGNMENT - 1

STRUCTURES AND POINTERS:

QUESTION 1:

```c
#include <stdio.h>

struct Friends {
    char name[50];
    char petName[50];
    char phoneNumber[15];
    struct Type {
        char friendType[20];
        char commonFriends[100];
        int placesVisited;
    } type;
};
void main() {
    int num;
    printf("Enter the number of friends to add: ");
    scanf("%d", &num);

    struct Friends friends[num];

    for (int i = 0; i < num; i++) {
        printf("\nEntering details for friend %d:\n", i + 1);

        printf("Enter friend's name: ");
        scanf(" %[^\n]%*c", friends[i].name);

        printf("Enter pet name: ");
        scanf(" %[^\n]%*c", friends[i].petName);

        printf("Enter phone number: ");
        scanf(" %[^\n]%*c", friends[i].phoneNumber);

        printf("Enter type of friend (School/College/Area): ");
        scanf(" %[^\n]%*c", friends[i].type.friendType);

        printf("Enter names of common friends: ");
        scanf(" %[^\n]%*c", friends[i].type.commonFriends);

        printf("Enter number of places visited together: ");
        scanf("%d", &friends[i].type.placesVisited);
    }
```

```c

    printf("\nFriends List:\n");
    printf("S.No.   Name          Pet Name     Phone Number  Type of Friend  Name of Common Friends
              No. of Places Visited\n");
    printf("-----------------------------------------------------------------------------------------
    ----------------------------------------------------------------------------------------------\n");
    for (int i = 0; i < num; i++) {
        printf("%-7d %-15s %-12s %-14s %-15s %-30s %d\n",
               i + 1,
               friends[i].name,
               friends[i].petName,
               friends[i].phoneNumber,
               friends[i].type.friendType,
               friends[i].type.commonFriends,
               friends[i].type.placesVisited);
    }
}
```

OUTPUT:

```
Enter the number of friends to add: 3

Entering details for friend 1:
Enter friend's name: Manu
Enter pet name: Unam
Enter phone number: 9876543210
Enter type of friend (School/College/Area): College
Enter names of common friends: aswin, manu, manu
Enter number of places visited together: 5

Entering details for friend 2:
Enter friend's name: Aswin
Enter pet name: ubihs
Enter phone number: 12345678901
Enter type of friend (School/College/Area): College
Enter names of common friends: manu, manu, Manu
Enter number of places visited together: 1

Entering details for friend 3:
Enter friend's name: Vishwagosh
Enter pet name: wish
Enter phone number: 6789054321
Enter type of friend (School/College/Area): School
Enter names of common friends: 0
Enter number of places visited together: 2
```

```
Friends List:
S.No.   Name          Pet Name     Phone Number  Type of Friend  Name of Common Friends
                No. of Places Visited
-------------------------------------------------------------------------------------
    ---------------------------------------------------------------------------
    -----------
1       Manu            Unam         9876543210     College         aswin, manu, manu
              5
2       Aswin           ubihs        12345678901    College         manu, manu, Manu
              1
3       Vishwagosh      wish         6789054321     School          0
              2
```

QUESTION 2:

```c
#include <stdio.h>

struct Product {
    char name[50];
    int id;
    float price;
};

int main() {
    int num;
    printf("Enter the number of Products to add: ");
    scanf("%d", &num);

    struct Product product[num];
    struct Product* ptr = product;
    float totalCost = 0;
    struct Product *mostExpensive = &product[0];
    struct Product *leastExpensive = &product[0];

    for (int i = 0; i < num; i++) {
        printf("\nEntering details for Product %d:\n", i + 1);

        printf("Enter product name: ");
        scanf(" %[^\n]%*c", ptr[i].name);

        printf("Enter product ID: ");
        scanf("%d", &ptr[i].id);

        printf("Enter product price: ");
        scanf("%f", &ptr[i].price);

        totalCost += ptr[i].price;
        if (ptr[i].price > mostExpensive->price) {
            mostExpensive = &ptr[i];
        }
        if (ptr[i].price < leastExpensive->price) {
            leastExpensive = &ptr[i];
        }
    }

    printf("\nProduct Details:\n");
    for (int i = 0; i < num; i++) {
        printf("Name: %s, ID: %d, Price: %.2f\n", ptr[i].name, ptr[i].id, ptr[i].price);
    }

    printf("\nMost Expensive Product:\n");
    printf("Name: %s, ID: %d, Price: %.2f\n", mostExpensive->name, mostExpensive->id, mostExpensive->price);

    printf("Least Expensive Product:\n");
    printf("Name: %s, ID: %d, Price: %.2f\n", leastExpensive->name, leastExpensive->id, leastExpensive->price);

    printf("Total Cost: %.2f\n", totalCost);

    return 0;
}
```

OUTPUT:

```
Enter the number of Products to add: 5

Entering details for Product 1:
Enter product name: Pen
Enter product ID: 1
Enter product price: 10

Entering details for Product 2:
Enter product name: Laptop
Enter product ID: 2
Enter product price: 60000

Entering details for Product 3:
Enter product name: Television
Enter product ID: 3
Enter product price: 32000

Entering details for Product 4:
Enter product name: Bag
Enter product ID: 4
Enter product price: 4000

Entering details for Product 5:
Enter product name: watch
Enter product ID: 5
Enter product price: 1000
```

```
Product Details:
Name: Pen, ID: 1, Price: 10.00
Name: Laptop, ID: 2, Price: 60000.00
Name: Television, ID: 3, Price: 32000.00
Name: Bag, ID: 4, Price: 4000.00
Name: watch, ID: 5, Price: 1000.00

Most Expensive Product:
Name: Laptop, ID: 2, Price: 60000.00
Least Expensive Product:
Name: Pen, ID: 1, Price: 10.00
Total Cost: 97010.00
```