DSA Assignment

Structures and Pointers

1.

```
#include <stdio.h>
#include <string.h>
//declaring a structure Type
struct Type {
   char type[20]; // School, College or Area friend
   char common_friends[50];// common friends name
   int places_visited; //no. of places visited
};

//declaring another structure Friends
struct Friends{
   char name[50];                    //name of friend
   char pet_name[20];                  // his/her pet name
   char phone_number[15];                // his / her ph.no.
   struct Type type;                  //nesting Type into Friends by creating object inside Friends
} ;

int main() {                          //declaring main()
   int n;
   printf("Enter the number of friends to add: ");
   scanf("%d", &n);                    // getting no. of friends
```

```
struct Friends friends[n];              // creatibg n no. of objects for Friends


for (int i = 0; i < n; i++) {           //getting inputs using fir loop dynamically
   printf("Enter Friend Name: ");
   scanf("%s", friends[i].name);        //dot operator is used to access variables of objects
   printf("Enter Pet Name: ");
   scanf("%s", friends[i].pet_name);
   printf("Enter Phone Number: ");
   scanf("%s", friends[i].phone_number);
   printf("Enter Type of friend: ");
   scanf("%s", friends[i].type.type);     //using dot operator twice to access variables in nested structure
   printf("Enter Name of Common Friends: ");
   scanf("%s", friends[i].type.common_friends);
   printf("Enter No. of Places Visited Together: ");
   scanf("%d", &friends[i].type.places_visited);
}



printf("\nFriend Details:\n");             //printing details obtained using for loop
printf("Friends Name\tPet Name\tPhone number\tType os Friend\tPlaces visited together");
for (int i = 0; i < n; i++) {
   printf("\n%s\t\t", friends[i].name);
   printf("%s\t\t", friends[i].pet_name);
   printf("%s\t\t", friends[i].phone_number);
   printf("%s\t\t", friends[i].type.type);
   printf("%s\t\t", friends[i].type.common_friends);
   printf("%d\n\n", friends[i].type.places_visited);
```

```
    }


    return 0;

}
```

main.c

Run

```c
#include <stdio.h>
#include <string.h>
//declaring a structure Type
struct Type {
    char type[20]; // School, College or Area friend
    char common_friends[50];// common friends name
    int places_visited; //no. of places visited
};

//declaring another structure Friends
struct Friends{
    char name[50];
                //name of friend
    char pet_name[20];
                // his/her pet name
    char phone_number[15];
                // his / her ph.no.
    struct Type type;
                //nesting Type into Friends by creating object inside Friends
} ;

int main() {
                //declaring main()
    int n;
    printf("Enter the number of friends to add: ");
    scanf("%d", &n);
                // getting no. of friends
```

Output

Clear

```
/tmp/cUI5rYKrhA.o
Enter the number of friends to add: 2
Enter Friend Name: aravind
Enter Pet Name: ara
Enter Phone Number: 98765432
Enter Type of friend: school
Enter Name of Common Friends: haris
Enter No. of Places Visited Together: 4
Enter Friend Name: haris
Enter Pet Name: hari
Enter Phone Number: 987564321
Enter Type of friend: school
Enter Name of Common Friends: aravind
Enter No. of Places Visited Together: 4

Friend Details:
Friends Name    Pet Name    Phone number
    Type os Friend  Places visited together
aravind     ara     98765432        school
        haris       4


haris       hari        987564321
    school      aravind     4



=== Code Execution Successful ===
```

2.

```
#include <stdio.h>
#include <string.h>

// Define the Product structure
struct Product {
   char name[50];        //declaring variables inside the structure
   char id[10];
   float price;
};

int main() {                  //main function
   int n, i;
   float totalCost = 0;
   struct Product products[5];       // creating objects for struct Product
   struct Product *mostExpensive, *leastExpensive; //creating pointers for structuree

   printf("Enter the number of products: ");
   scanf("%d", &n);                     //getting no. of products as input daynamically
```

```
// Input details for each product
for(i = 0; i < n; i++) {                //getting inputs for all the object using variable dynamically
   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("%f", &products[i].price);
}

// Initializing mostExpensive and leastExpensive pointers
mostExpensive = &products[0];
leastExpensive = &products[0];

// Calculating total cost and find most/least expensive products using linear sorting algorithm
for(i = 0; i < n; i++) {
   totalCost += products[i].price;

   if(products[i].price > mostExpensive->price) {          //using linear sorting algorithm
      mostExpensive = &products[i];                    //complexity O(n)
   }
   if(products[i].price < leastExpensive->price) {
      leastExpensive = &products[i];
```

```
    }
  }

  // printing all product details
  printf("\nProduct Details:\n");
  for(i = 0; i < n; i++) {
    printf("Product Name: %s, Product ID: %s, Price: %.2f\n", products[i].name, products[i].id, products[i].price);
  }

  // printing the most expensive product
  printf("\nMost Expensive Product: ");
  printf("Product Name: %s, Product ID: %s, Price: %.2f\n", mostExpensive->name, mostExpensive->id, mostExpensive->price);

  // printing the least expensive product
  printf("Least Expensive Product: ");
  printf("Product Name: %s, Product ID: %s, Price: %.2f\n", leastExpensive->name, leastExpensive->id, leastExpensive->price);

  // printing the total cost of all products
  printf("Total Cost of All Products: %.2f\n", totalCost);

  return 0;
}
```

main.c

Run

```c
#include <stdio.h>
#include <string.h>

// Define the Product structure
struct Product {
    char name[50];
        //declaring variables inside
        the structure
    char id[10];
    float price;
};

int main() {
    //main function
    int n, i;
    float totalCost = 0;
    struct Product products[5];
         // creating objects for
        struct Product
    struct Product *mostExpensive,
        *leastExpensive; //creating
        pointers for structuree

    printf("Enter the number of
        products: ");
    scanf("%d", &n);
                    //getting no. of
        products as input daynamically

    // Input details for each product
    for(i = 0; i < n; i++) {
                     //getting inputs
        for all the object using
        variable dynamically
```

Output

Clear

```
/tmp/up42dLLwcw.o
Enter the number of products: 2
Enter details for product 1:
Product Name: laptop
Product ID: 1
Price: 1000
Enter details for product 2:
Product Name: ipad
Product ID: 2
Price: 900

Product Details:
Product Name: laptop, Product ID: 1, Price:
    1000.00
Product Name: ipad, Product ID: 2, Price:
    900.00

Most Expensive Product: Product Name: laptop
    , Product ID: 1, Price: 1000.00
Least Expensive Product: Product Name: ipad,
    Product ID: 2, Price: 900.00
Total Cost of All Products: 1900.00


=== Code Execution Successful ===
```

GITHUB PROFILE LINK: **GitHub - AnishIdhayan-1412/C-programs**