

C Programs for Practice (Structures)

1. C Program to store information of students using Structure

```
// C Program to Store Information of Students Using Structure
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Create the student structure
struct Student {
    char* name;
    int roll_number;
    int age;
    double total_marks;
};

int main() {

    // Create an array of student structure variable with
    // 5 Student's records
    struct Student students[5];
    int n = sizeof(students)/sizeof(struct Student);

    // Get the students data
    students[0].roll_number = 1;
    students[0].name = "Geeks1";
    students[0].age = 12;
    students[0].total_marks = 78.50;

    students[1].roll_number = 5;
    students[1].name = "Geeks5";
    students[1].age = 10;
    students[1].total_marks = 56.84;

    students[2].roll_number = 2;
    students[2].name = "Geeks2";
    students[2].age = 11;
    students[2].total_marks = 87.94;

    students[3].roll_number = 4;
    students[3].name = "Geeks4";
    students[3].age = 12;
    students[3].total_marks = 89.78;
```

```

students[4].roll_number = 3;
students[4].name = "Geeks3";
students[4].age = 13;
students[4].total_marks = 78.55;

// Print the Students information
printf("=====\n");
printf("      Student Records      \n");
printf("=====\n");

for (int i = 0; i < n; i++) {
    printf("\nStudent %d:\n", i + 1);
    printf("  Name      : %s\n", students[i].name);
    printf("  Roll Number : %d\n", students[i].roll_number);
    printf("  Age       : %d\n", students[i].age);
    printf("  Total Marks : %.2f\n", students[i].total_marks);
}

printf("=====\n");

return 0;
}

```

2. C Program to store student records as Structures and sort them by name

```

// C program to read Student records
// like id, name and age,
// and display them in sorted order by Name

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

// struct person with 3 fields
struct Student {
    char* name;
    int id;
    char age;
};

// setting up rules for comparison
// to sort the students based on names
int comparator(const void* p, const void* q)

```

```
{
    return strcmp(((struct Student*)p)->name,
                  ((struct Student*)q)->name);
}
```

// Driver program

int main()

```
{
    int i = 0, n = 5;

    struct Student arr[n];

    // Get the students data
    arr[0].id = 1;
    arr[0].name = "bd";
    arr[0].age = 12;

    arr[1].id = 2;
    arr[1].name = "ba";
    arr[1].age = 10;

    arr[2].id = 3;
    arr[2].name = "bc";
    arr[2].age = 8;

    arr[3].id = 4;
    arr[3].name = "aaz";
    arr[3].age = 9;

    arr[4].id = 5;
    arr[4].name = "az";
    arr[4].age = 10;

    // Print the Unsorted Structure
    printf("Unsorted Student Records:\n");
    for (i = 0; i < n; i++) {
        printf("Id = %d, Name = %s, Age = %d \n",
              arr[i].id, arr[i].name, arr[i].age);
    }
    // Sort the structure
    // based on the specified comparator
    qsort(arr, n, sizeof(struct Student), comparator);

    // Print the Sorted Structure
```

```

        printf("\n\nStudent Records sorted by Name:\n");
        for (i = 0; i < n; i++) {
            printf("Id = %d, Name = %s, Age = %d \n",
                arr[i].id, arr[i].name, arr[i].age);
        }

        return 0;
    }

```

3. C Program to write Structure to a File

```

// C program for writing
// struct to file
#include <stdio.h>
#include <stdlib.h>

// a struct to be read and written
struct person {
    int id;
    char fname[20];
    char lname[20];
};

int main()
{
    FILE* outfile;

    // open file for writing
    outfile = fopen("person.bin", "wb");
    if (outfile == NULL) {
        fprintf(stderr, "\nError opened file\n");
        exit(1);
    }

    struct person input1 = { 1, "rohan", "sharma" };

    // write struct to file
    int flag = 0;
    flag = fwrite(&input1, sizeof(struct person), 1,
        outfile);
    if (flag) {
        printf("Contents of the structure written "
            "successfully");
    }
    else

```

```

        printf("Error Writing to File!");

    // close file
    fclose(outfile);

    return 0;
}

```

4. C Program to read structure from a File

```

// C program for reading
// struct from a file
#include <stdio.h>
#include <stdlib.h>

// struct person with 3 fields
struct person {
    int id;
    char fname[20];
    char lname[20];
};

// Driver program
int main()
{
    FILE* infile;

    // Open person.dat for reading
    infile = fopen("person1.dat", "wb+");
    if (infile == NULL) {
        fprintf(stderr, "\nError opening file\n");
        exit(1);
    }

    struct person write_struct = { 1, "Rohan", "Sharma" };

    // writing to file
    fwrite(&write_struct, sizeof(write_struct), 1, infile);

    struct person read_struct;

    // setting pointer to start of the file
    rewind(infile);

    // reading to read_struct

```

```
fread(&read_struct, sizeof(read_struct), 1, infile);

printf("Name: %s %s \nID: %d", read_struct.fname,
      read_struct.lname, read_struct.id);

// close file
fclose(infile);

return 0;
}
```