

Pgm 7.2.1

```
struct student
{
    int regno;
    int contactno;
    char * name;
    struct address Address;
}
```

Declare a pointer to this structure and allocate memory dynamically for 5 students. Implement functions(read and display) to read data for 5 students and print the same.

Code:

```
#include <stdio.h>
#include <stdlib.h>
struct address
{
    int dno;
    char street[10];
    char area[10];
    char city[10];
};
struct student
{
    int regno;
    int contactno;
    char * name;
    struct address ad;
};

void write(struct student *ptri)
{
    printf(" \n name \n");
    char* cptr = malloc(20);
    scanf("%s",cptr);
```

```

    ptri->name = cptr;
    printf(" \n regno \n");
    scanf("%d",&(ptri->regno));
    printf(" \n contact No \n");
    scanf("%d",&(ptri->contactno));
    printf(" \n d.no \n");
    scanf("%d",&(ptri->ad.dno));
    printf(" \n street \n");
    scanf("%s",(ptri->ad.street));
    printf(" \n area \n");
    scanf("%s",(ptri->ad.area));
    printf(" \n city \n");
    scanf("%s",(ptri->ad.city));
}

void read(struct student *ptri)
{
    printf(" \n name: %s\n",(ptri->name));
    printf(" \n regno: %d \n",(ptri->regno));
    printf(" \n contact: %d\n",(ptri->contactno));
    printf(" \n dno: %d\n",(ptri->ad.dno));
    printf(" \n street : %s\n",(ptri->ad.street));
    printf(" \n area : %s\n",(ptri->ad.area));
    printf(" \n city : %s\n",(ptri->ad.city));
}

int main()
{
    struct student *ptr;
    ptr = malloc(5*sizeof(struct student));
    for (struct student* i = ptr; i < (ptr+5); i++) {
        printf("\n writing data for Student %d\n", (int)(i-ptr+1));
        write(i);
    }
    for (struct student* i = ptr; i < (ptr+5); i++) {
        printf("\n reading data from Student %d\n", (int)(i-ptr+1));
        read(i);
    }
    return 0;
}

```

Output and code screenshots:

```
navin@DESKTOP-D400C38: /mnt/d/ds lab
reading data from Student 3
name: swami
regno: 556
contact: 667889
dno: 6
street : kai
area : chung
city : ha
reading data from Student 4
name: nathan
regno: 455
contact: 6678
dno: 6
street : maisan
area : chinatown
city : singapore
reading data from Student 5
name: swaminathan
regno: 115126
contact: 997766
dno: 6
street : seoultown
area : tokyo
city : japan
navin@DESKTOP-D400C38: /mnt/d/ds lab$
```

```
navin@DESKTOP-D400C38: /mnt/d/ds lab
contact llo
997766
d.no
6
street
seoultown
area
tokyo
city
japan
reading data from Student 1
name: Navin
regno: 2019
contact: 5566
dno: 5
street : indira
area : adayar
city : chennai
reading data from Student 2
name: ashok
regno: 566
contact: 5667
dno: 6
street : thai
area : tnagar
city : chennai
reading data from Student 3
name: swami
```

```
navin@DESKTOP-D400C38: /mnt/d/ds lab
city
writing data for Student 4
name
nathan
regno
455
contact No
6678
d.no
6
street
maisan
area
chinatown
city
singapore
writing data for Student 5
name
swaminathan
regno
115126
contact No
997766
d.no
6
street
seculitown
area
tokyo
city
japan
reading data from Student 1
```

```
navin@DESKTOP-D400C38: /mnt/d/ds lab
regno
566
contact No
5667
d.no
6
street
thai
area
tnagar
city
chennai
writing data for Student 3
name
swami
regno
556
contact No
667889
d.no
6
street
kai
area
chung ha
city
writing data for Student 4
name
nathan
regno
455
contact No
6678
```

```
navin@DESKTOP-D400C38: /mnt/d/ds lab$ ./st
writing data for Student 1
name
Navin
regno
2019
contact No
5566
d.no
5
street
indira
area
adayar
city
chennai
writing data for Student 2
name
ashok
regno
566
contact No
5667
d.no
6
street
thai
area
tnagar
city
chennai
writing data for Student 3
```

```
navin@DESKTOP-D400C38: /mnt/d/ds lab$ ./st
navin@DESKTOP-D400C38:~$ cd /mnt
navin@DESKTOP-D400C38:/mnt$ cd d
navin@DESKTOP-D400C38:/mnt/d$ cd 'ds lab'
navin@DESKTOP-D400C38:/mnt/d/ds lab$ gcc st.c -o st
navin@DESKTOP-D400C38:/mnt/d/ds lab$ ./st
writing data for Student 1
name
Navin
regno
2019
contact No
5566
d.no
5
street
indira
area
adayar
city
chennai
writing data for Student 2
name
ashok
regno
566
contact No
5667
d.no
6
street
thai
area
tnagar
city
```

The screenshot shows the Visual Studio Code editor with a C program in a file named `student.c`. The program defines a `struct student` with fields for `regno`, `contactno`, `name`, and a pointer to a `struct address`. The `struct address` contains `dno`, `street`, `area`, and `city`. The `write` function takes a pointer to a `struct student` and prompts the user to enter data for each field, including dynamic memory allocation for the `name` field.

```
d: > ds lab > C stc > student
1 #include <stdio.h>
2 #include <stdlib.h>
3 struct address
4 {
5     int dno;
6     char street[10];
7     char area[10];
8     char city[10];
9 };
10 struct student
11 {
12     int regno;
13     int contactno;
14     char * name;
15     struct address ad;
16 };
17
18
19
20 void write(struct student *ptr1)
21 {
22     printf(" \n name \n");
23     char* cptr = malloc(20);
24     scanf("%s",cptr);
25     ptr1->name = cptr;
26     printf(" \n regno \n");
27     scanf("%d",&(ptr1->regno));
28     printf(" \n contact No \n");
29     scanf("%d",&(ptr1->contactno));
30     printf(" \n d.no \n");
31     scanf("%d",&(ptr1->ad.dno));
32     printf(" \n street \n");
33     scanf("%s",(ptr1->ad.street));
34     printf(" \n area \n");
35     scanf("%s",(ptr1->ad.area));
36     printf(" \n city \n");
37     scanf("%s",(ptr1->ad.city));
38 }
```

The screenshot shows the continuation of the C program. It includes a `read` function that displays the stored data for a student. The `main` function allocates an array of `struct student` pointers, iterates through them to write and then read data, and finally returns 0.

```
30 printf(" \n d.no \n");
31 scanf("%d",&(ptr1->ad.dno));
32 printf(" \n street \n");
33 scanf("%s",(ptr1->ad.street));
34 printf(" \n area \n");
35 scanf("%s",(ptr1->ad.area));
36 printf(" \n city \n");
37 scanf("%s",(ptr1->ad.city));
38
39
40
41 void read(struct student *ptr1)
42 {
43     printf(" \n name: %s\n",(ptr1->name));
44     printf(" \n regno: %d \n",(ptr1->regno));
45     printf(" \n contact: %d\n",(ptr1->contactno));
46     printf(" \n dno: %d\n",(ptr1->ad.dno));
47     printf(" \n street : %s\n",(ptr1->ad.street));
48     printf(" \n area : %s\n",(ptr1->ad.area));
49     printf(" \n city : %s\n",(ptr1->ad.city));
50 }
51
52 int main()
53 {
54     struct student *ptr;
55     ptr = malloc(5*sizeof(struct student));
56     for (struct student* i = ptr; i < (ptr+5); i++) {
57         printf("\n writing data for Student %d\n", (int)(i-ptr+1));
58         write(i);
59     }
60     for (struct student* i = ptr; i < (ptr+5); i++) {
61         printf("\n reading data from Student %d\n", (int)(i-ptr+1));
62         read(i);
63     }
64     return 0;
65 }
66
```