```c
#include <stdio.h>

// DEFINING STRUCT
struct MyData
{
    int i;
    float f;
    double d;
    char c;
};

int main(void)
{
    //variable declarations

    // 35 will be assigned to 'i' of 'data_one'
    // 3.9 will be assigned to 'f' of 'data_one'
    // 1.23765 will be assigned to 'd' of 'data_one'
    // 'A' will be assigned to 'c' of 'data_one'
    struct MyData data_one = { 35, 3.9f, 1.23765, 'A'}; //Inline Initialization

    // 'P' will be assigned to 'i' of 'data_two' ... but 'P' is a character (char) and 'i' is an integer ... so 'P' is converted into it decimal integer ASCII value (80) and 80 is assigned to 'i' of data_two
    // 6.2 will be assigned to 'f' of 'data_two'
    // 12.199523 will be assigned to 'd' of 'data_two'
    // 68 will be assigned to 'c' of 'data_two' ... but 68 is an integer (int) and 'c' is a 'char' ... so 68 is considered as a decimal ASCII value and its corressponding character ( 'D' ) is assigned to 'c' of data_two
    struct MyData data_two = { 'P', 6.2f, 12.199523, 68 }; //Inline Initialization

    // 36 will be assigned to 'i' of 'data_three'
    // 'G' is 'char', but 'f' of 'data_three' is 'float'...hence, 'G' is converted to its decimal integer ASCII value (71) and this in turn is converted to 'float' (71.000000) and then it will be assigned to 'f' of 'data_three'
    // 0.0000000 will be assigned to 'd' of 'data_three'
    // No character will be assigned to 'c' of 'data_three'
    struct MyData data_three = { 36, 'G' }; //Inline Initialization

    // 79 will be assigned to 'i' of 'data_four'
    // 0.000000 will be assigned to 'f' of 'data_four'
    // 0.000000 will be assigned to 'd' of 'data_four'
    // No character will be assigned to 'c' of 'data_four'
    struct MyData data_four = { 79 }; //Inline Initialization

    //code
    //Displaying Values Of The Data Members Of 'struct MyData'
    printf("\n\n");
    printf("DATA MEMBERS OF 'struct MyData data_one' ARE : \n\n");
    printf("i = %d\n", data_one.i);
    printf("f = %f\n", data_one.f);
    printf("d = %lf\n", data_one.d);
```

```c
    printf("c = %c\n", data_one.c);

    printf("\n\n");
    printf("DATA MEMBERS OF 'struct MyData data_two' ARE : \n\n");
    printf("i = %d\n", data_two.i);
    printf("f = %f\n", data_two.f);
    printf("d = %lf\n", data_two.d);
    printf("c = %c\n", data_two.c);

    printf("\n\n");
    printf("DATA MEMBERS OF 'struct MyData data_three' ARE : \n\n");
    printf("i = %d\n", data_three.i);
    printf("f = %f\n", data_three.f);
    printf("d = %lf\n", data_three.d);
    printf("c = %c\n", data_three.c);

    printf("\n\n");
    printf("DATA MEMBERS OF 'struct MyData data_four' ARE : \n\n");
    printf("i = %d\n", data_four.i);
    printf("f = %f\n", data_four.f);
    printf("d = %lf\n", data_four.d);
    printf("c = %c\n", data_four.c);

    return(0);
}
```