```c
#include <stdio.h>

// DEFINING STRUCT
struct MyData
{
    int i;
    float f;
    double d;
    char c;
} data; //Declaring a single struct variable of type 'struct MyData' globally ...

int main(void)
{
    //variable declarations
    int i_size;
    int f_size;
    int d_size;
    int struct_MyData_size;

    //code
    //Assigning Data Values To The Data Members Of 'struct MyData'
    data.i = 30;
    data.f = 11.45f;
    data.d = 1.2995;

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

    //Calculating Sizes (In Bytes) Of The Data Members Of 'struct MyData'
    i_size = sizeof(data.i);
    f_size = sizeof(data.f);
    d_size = sizeof(data.d);

    //Displaying Sizes (In Bytes) Of The Data Members Of 'struct MyData'
    printf("\n\n");
    printf("SIZES (in bytes) OF DATA MEMBERS OF 'struct MyData' ARE : \n\n");
    printf("Size of 'i' = %d bytes\n", i_size);
    printf("Size of 'f' = %d bytes\n", f_size);
    printf("Size of 'd' = %d bytes\n", d_size);

    //Calculating Size (In Bytes) Of the entire 'struct Mydata'
    struct_MyData_size = sizeof(struct MyData); //can also give struct name ->
      sizeof(MyData)

   //Displaying Sizes (In Bytes) Of the entire 'struct Mydata'
    printf("\n\n");
    printf("Size of 'struct MyData' : %d bytes\n\n", struct_MyData_size);

```

```
52      return(0);
53  }
54
```