

```
1  #include <stdio.h>
2
3  // DEFINING STRUCT
4  struct MyData
5  {
6      int i;
7      float f;
8      double d;
9  };
10
11 int main(void)
12 {
13     //variable declarations
14     int i_size;
15     int f_size;
16     int d_size;
17     int struct_MyData_size;
18     int pointer_to_struct_MyData_size;
19
20     struct MyData *pData = NULL;
21
22     //code
23     printf("\n\n");
24
25     pData = (struct MyData *)malloc(sizeof(struct MyData));
26     if (pData == NULL)
27     {
28         printf("FAILED TO ALLOCATE MEMORY TO 'struct MyData' !!! EXITTING NOW ... ↗
29             \n\n");
30         exit(0);
31     }
32     else
33         printf("SUCCESSFULLY ALLOCATED MEMORY TO 'struct MyData' !!!\n\n");
34
35     //Assigning Data Values To The Data Members Of 'struct MyData'
36     (*pData).i = 30;
37     (*pData).f = 11.45f;
38     (*pData).d = 1.2995;
39
40     //Displaying Values Of The Data Members Of 'struct MyData'
41     printf("\n\n");
42     printf("DATA MEMBERS OF 'struct MyData' ARE : \n\n");
43     printf("i = %d\n", (*pData).i);
44     printf("f = %f\n", (*pData).f);
45     printf("d = %lf\n", (*pData).d);
46
47     //Calculating Sizes (In Bytes) Of The Data Members Of 'struct MyData'
48     i_size = sizeof((*pData).i);
49     f_size = sizeof((*pData).f);
50     d_size = sizeof((*pData).d);
51
```

```
52 //Displaying Sizes (In Bytes) Of The Data Members Of 'struct MyData'
53 printf("\n\n");
54 printf("SIZES (in bytes) OF DATA MEMBERS OF 'struct MyData' ARE : \n\n");
55 printf("Size of 'i' = %d bytes\n", i_size);
56 printf("Size of 'f' = %d bytes\n", f_size);
57 printf("Size of 'd' = %d bytes\n", d_size);
58
59 //Calculating Size (In Bytes) Of the entire 'struct Mydata'
60 struct_MyData_size = sizeof(struct MyData);
61 pointer_to_struct_MyData_size = sizeof(struct MyData *);
62
63 //Displaying Sizes (In Bytes) Of the entire 'struct Mydata'
64 printf("\n\n");
65 printf("Size of 'struct MyData' : %d bytes\n\n", struct_MyData_size);
66 printf("Size of pointer to 'struct MyData' : %d bytes\n\n",
        pointer_to_struct_MyData_size);
67
68 if (pData)
69 {
70     free(pData);
71     pData = NULL;
72     printf("MEMORY ALLOCATED TO 'struct MyData' HAS BEEN SUCCESSFULLY
        FREED !!!\n\n");
73 }
74
75 return(0);
76 }
77
78
```