

```
1  #include <stdio.h>
2
3  // DEFINING STRUCT
4  struct MyData
5  {
6      int *ptr_i;
7      int i;
8
9      float *ptr_f;
10     float f;
11
12     double *ptr_d;
13     double d;
14 };
15
16 int main(void)
17 {
18     //variable declarations
19     struct MyData *pData = NULL;
20
21     //code
22     printf("\n\n");
23     pData = (struct MyData *)malloc(sizeof(struct MyData));
24     if (pData == NULL)
25     {
26         printf("FAILED TO ALLOCATE MEMORY TO 'struct MyData' !!! EXITTING NOW ... ↗
27             \n\n");
28         exit(0);
29     }
30     else
31         printf("SUCCESSFULLY ALLOCATED MEMORY TO 'struct MyData' !!!\n\n");
32
33     pData->i = 9;
34     pData->ptr_i = &(pData->i);
35
36     pData->f = 11.45f;
37     pData->ptr_f = &(pData->f);
38
39     pData->d = 30.121995;
40     pData->ptr_d = &(pData->d);
41
42     printf("\n\n");
43     printf("i = %d\n", *(pData->ptr_i));
44     printf("Address Of 'i' = %p\n", pData->ptr_i);
45
46     printf("\n\n");
47     printf("f = %f\n", *(pData->ptr_f));
48     printf("Address Of 'f' = %p\n", pData->ptr_f);
49
50     printf("\n\n");
51     printf("d = %lf\n", *(pData->ptr_d));
52     printf("Address Of 'd' = %p\n", pData->ptr_d);
```

```
52
53     if (pData)
54     {
55         free(pData);
56         pData = NULL;
57         printf("MEMORY ALLOCATED TO 'struct MyData' HAS BEEN SUCCESSFULLY
58             FREED !!!\n\n");
59     }
60     return(0);
61 }
62
63
64
```