```c
#include <stdio.h>

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

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

    typedef struct MyData* MyDataPtr;

    MyDataPtr pData;

    //code
    printf("\n\n");

    pData = (MyDataPtr)malloc(sizeof(struct MyData));
    if (pData == NULL)
    {
        printf("FAILED TO ALLOCATE MEMORY TO 'sturct MyData' !!! EXITTING NOW ... \n\n");
        exit(0);
    }
    else
        printf("SUCCESSFULLY ALLOCATED MEMORY TO 'sturct MyData' !!!\n\n");


    //Assigning Data Values To The Data Members Of 'struct MyData'
    pData->i = 30;
    pData->f = 11.45f;
    pData->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", pData->i);
    printf("f = %f\n", pData->f);
    printf("d = %lf\n", pData->d);

    //Calculating Sizes (In Bytes) Of The Data Members Of 'struct MyData'
    i_size = sizeof(pData->i);
    f_size = sizeof(pData->f);
```

```
52      d_size = sizeof(pData->d);
53
54      //Displaying Sizes (In Bytes) Of The Data Members Of 'struct MyData'
55      printf("\n\n");
56      printf("SIZES (in bytes) OF DATA MEMBERS OF 'struct MyData' ARE : \n\n");
57      printf("Size of 'i' = %d bytes\n", i_size);
58      printf("Size of 'f' = %d bytes\n", f_size);
59      printf("Size of 'd' = %d bytes\n", d_size);
60
61      //Calculating Size (In Bytes) Of the entire 'struct Mydata'
62      struct_MyData_size = sizeof(struct MyData);
63      pointer_to_struct_MyData_size = sizeof(MyDataPtr);
64
65     //Displaying Sizes (In Bytes) Of the entire 'struct Mydata'
66      printf("\n\n");
67      printf("Size of 'struct MyData' : %d bytes\n\n", struct_MyData_size);
68      printf("Size of pointer to 'struct MyData' : %d bytes\n\n",                    ➥
          pointer_to_struct_MyData_size);
69
70      if (pData)
71      {
72          free(pData);
73          pData = NULL;
74          printf("MEMORY ALLOCATED TO 'struct MyData' HAS BEEN SUCCESSFULLY          ➥
            FREED !!!\n\n");
75      }
76
77      return(0);
78  }
79
80
```