```
1 #include <stdio.h>
 2 #include <stdlib.h>
 3
 4 int main(void)
 5 {
 6
        void MyAlloc(int **ptr, unsigned int numberOfElements);
 7
        //variable declarations
 8
        int *piArray = NULL;
 9
10
        unsigned int num_elements;
11
        int i;
12
        // code
13
14
        printf("\n\n");
15
        printf("How Many Elements You Want In Integer Array ?\n\n");
16
        scanf("%u", &num_elements);
17
18
        printf("\n\n");
19
        MyAlloc(&piArray, num_elements);
20
21
        printf("Enter %u Elements To Fill Up Your Integer Array : \n\n",
                                                                                          P
          num_elements);
22
        for (i = 0; i < num elements; i++)</pre>
23
            scanf("%d", &piArray[i]);
24
25
        printf("\n\n");
26
        printf("The %u Elements Entered By You In The Integer Array : \n\n",
          num_elements);
27
        for (i = 0; i < num elements; i++)</pre>
28
            printf("%u\n", piArray[i]);
29
30
        printf("\n\n");
31
        if (piArray)
32
        {
33
            free(piArray);
34
            piArray = NULL;
35
            printf("Memory Allocated Has Now Been Successfully Freed !!!\n\n");
36
        }
37
38
        return(0);
39 }
40
41 void MyAlloc(int **ptr, unsigned int numberOfElements)
42 {
        // code
43
44
        *ptr = (int *)malloc(numberOfElements * sizeof(int));
45
        if (*ptr == NULL)
46
        {
47
            printf("Could Not Allocate Memory !!! Exitting Now ...\n\n");
48
            exit(0);
49
        }
50
```

```
...8-PointerToPointerFor1DArray\PointerToPointerFor1DArray.c 2
51 printf("MyAlloc() Has Successfully Allocated %lu Bytes For Integer Array !!!\n >
             \n", (numberOfElements * sizeof(int)));
52 }
53
```