

```

1  /*****
2  *  reallocate: learn how to allocate and resize arrays dynamically *
3  *****/
4  #include <stdio.h>
5  #include <stdlib.h>
6
7  /*****
8  *  Function prototypes *
9  *****/
10
11 int* new_int_array(size_t size);
12
13 void set_values_int_array(int* array, size_t size);
14
15 void print_int_array(int* array, size_t size);
16
17 int* resize_int_array(int* array, size_t new_size);
18
19 // driver program
20 int main(){
21     size_t size = 10;
22     int* my_array = new_int_array(size);
23     set_values_int_array(my_array, size);
24     print_int_array(my_array, size);
25
26     size_t new_size = 20;
27     my_array = resize_int_array(my_array, new_size);
28     print_int_array(my_array, new_size);
29     set_values_int_array(my_array, new_size);
30     print_int_array(my_array, new_size);
31
32     free(my_array);
33     return 0;
34 }
35
36 /*****
37 *  Function definitions *
38 *****/
39
40 int* new_int_array(size_t size){
41     int* array = (int*) malloc(size * sizeof(int));
42     if(array == NULL){
43         printf("Catastrophic error, aborting!\n");
44         exit(1);
45     }
46     return array;
47 }
48
49 void set_values_int_array(int* array, size_t size){
50     for(size_t i = 0; i < size; i++){
51         *(array + i) = 5 * i;
52     }
53 }
54
55 void print_int_array(int* array, size_t size){
56     for(size_t i = 0; i < size; i++){
57         printf("%d ", array[i]);
58     }
59     printf("\n");
60 }
61
62 int* resize_int_array(int* array, size_t new_size){
63     int* tmp = realloc(array, new_size * sizeof(int));
64     if(tmp == NULL){
65         printf("Not enough space available.\n");
66         exit(1);
67     }
68     return tmp;
69 }

```