

main.c



Share

Run

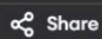
Output

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void staticAllocation() {
5     int arr[5] = {1, 2, 3, 4, 5};
6     printf("Static Allocation: ");
7     for (int i = 0; i < 5; i++) {
8         printf("%d ", arr[i]);
9     }
10    printf("\n");
11 }
12
13 void stackAllocation() {
14     void stackFunction() {
15         int arr[5] = {6, 7, 8, 9, 10};
16         printf("Stack Allocation: ");
17         for (int i = 0; i < 5; i++) {
18             printf("%d ", arr[i]);
19         }
20         printf("\n");
21     }
22     stackFunction();
23 }
24
25 void dynamicAllocation() {
26     int *arr = (int *)malloc(5 * sizeof(int));
27     for (int i = 0; i < 5; i++) {
28         arr[i] = i + 11;
29     }
30     printf("Dynamic Allocation: ");
31     for (int i = 0; i < 5; i++) {
32         printf("%d ", arr[i]);
33     }
34     printf("\n");
35     free(arr);
36 }
37
38 int main() {
39     staticAllocation();
40     stackAllocation();
41     dynamicAllocation();
42     return 0;
43 }
44
```

Static Allocation: 1 2 3 4 5  
Stack Allocation: 6 7 8 9 10  
Dynamic Allocation: 11 12 13 14 15

=== Code Execution Successful ===

main.c



Run

Output

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <dirent.h>
4 #include <string.h>
5
6 int compare(const void *a, const void *b) {
7     return strcmp(*(const char **)a, *(const char **)b);
8 }
9
10 int main() {
11     struct dirent **namelist;
12     int n;
13
14     n = scandir(".", &namelist, NULL, alphasort);
15     if (n < 0) perror("scandir");
16     else {
17         for (int i = 0; i < n; i++) {
18             printf("%s\n", namelist[i]->d_name);
19             free(namelist[i]);
20         }
21         free(namelist);
22     }
23     return 0;
24 }
25
```

```
.\
..\
.bash_logout
.bashrc
.profile
```

=== Code Execution Successful ===