

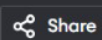
```
main.c
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void staticAllocation() {
5     int arr[5] = {1, 2, 3, 4, 5};
6     printf("Static Allocation: ");
7     for (int i = 0; i < 5; i++) printf("%d ", arr[i]);
8     printf("\n");
9 }
10 void stackAllocation() {
11     void stackFunction() {
12         int arr[5] = {6, 7, 8, 9, 10};
13         printf("Stack Allocation: ");
14         for (int i = 0; i < 5; i++) printf("%d ", arr[i]);
15         printf("\n");
16     }
17     stackFunction();
18 }
19 void dynamicAllocation() {
20     int *arr = (int *)malloc(5 * sizeof(int));
21     for (int i = 0; i < 5; i++) arr[i] = i + 11;
22     printf("Dynamic Allocation: ");
23     for (int i = 0; i < 5; i++) printf("%d ", arr[i]);
24     printf("\n");
25     free(arr);
26 }
27 int main() {
```

Output

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 #include <sys/stat.h>
6 #include <unistd.h>
7 void organize_files(const char *dir_path) {
8     struct dirent *entry;
9     DIR *dp = opendir(dir_path);
10    if (dp == NULL) {
11        perror("opendir");
12        return;
13    }
14    while ((entry = readdir(dp))) {
15        if (entry->d_type == DT_REG) {
16            char *ext = strrchr(entry->d_name, '.');
17            if (ext) {
18                char folder[256];
19                snprintf(folder, sizeof(folder), "%s/%s", dir_path, ext + 1);
20                struct stat st = {0};
21                if (stat(folder, &st) == -1) {
22                    if (mkdir(folder, 0755) == -1) {
23                        perror("mkdir");
24                        continue;
25                    }
26                }
27            }
28        }
29    }
30 }

```

Usage: /tmp/Va7T2Mrub6/main.o <directory_path>

=== Code Exited With Errors ===