Programiz C Online Compiler

main.c

Share Run

```c
#include <stdio.h>
#include <stdlib.h>

void staticAllocation() {
    int arr[5] = {1, 2, 3, 4, 5};
    printf("Static Allocation: ");
    for (int i = 0; i < 5; i++) printf("%d ", arr[i]);
    printf("\n");
}
void stackAllocation() {
    void stackFunction() {
        int arr[5] = {6, 7, 8, 9, 10};
        printf("Stack Allocation: ");
        for (int i = 0; i < 5; i++) printf("%d ", arr[i]);
        printf("\n");
    }
    stackFunction();
}
void dynamicAllocation() {
    int *arr = (int *)malloc(5 * sizeof(int));
    for (int i = 0; i < 5; i++) arr[i] = i + 11;
    printf("Dynamic Allocation: ");
    for (int i = 0; i < 5; i++) printf("%d ", arr[i]);
    printf("\n");
    free(arr);
}
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 ===
```

Programiz C Online Compiler

main.c

Share Run

```c
#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
#include <string.h>
#include <sys/stat.h>
#include <unistd.h>
void organize_files(const char *dir_path) {
    struct dirent *entry;
    DIR *dp = opendir(dir_path);
    if (dp == NULL) {
        perror("opendir");
        return;
    }
    while ((entry = readdir(dp))) {
        if (entry->d_type == DT_REG) {
            char *ext = strrchr(entry->d_name, '.');
            if (ext) {
                char folder[256];
                snprintf(folder, sizeof(folder), "%s/%s", dir_path, ext + 1);
                struct stat st = {0};
                if (stat(folder, &st) == -1) {
                    if (mkdir(folder, 0755) == -1) {
                        perror("mkdir");
                        continue;
                    }
                }

```

Output

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


=== Code Exited With Errors ===
```