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() {
    staticAllocation();
    stackAllocation();
    dynamicAllocation();
    return 0;
}
```

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

Share Run

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

int compare(const void *a, const void *b) {
    return strcmp(*(const char **)a, *(const char **)b);
}

int main() {
    struct dirent **namelist;
    int n;

    n = scandir(".", &namelist, NULL, alphasort);
    if (n < 0) perror("scandir");
    else {
        for (int i = 0; i < n; i++) {
            printf("%s\n", namelist[i]->d_name);
            free(namelist[i]);
        }
        free(namelist);
    }
    return 0;
}
```

Output

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


=== Code Execution Successful ===
```