```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>

int main() {
    int arr[] = {10, 5, 8, 2, 1};
    int n = sizeof(arr) / sizeof(arr[0]);

    // Create a child process
    pid_t child_pid = fork();

    if (child_pid < 0) {
        perror("Fork failed");
        exit(1);
    }

    if (child_pid == 0) {
        // Child process
        char *args[] = {"child_program", NULL};
        execve("child_program", args, NULL);
        perror("Execve failed");
        exit(1);
    } else {
        // Parent process
        wait(NULL);

        // Sort the array (in ascending order)
```

```
        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (arr[j] > arr[j + 1]) {
                    // Swap arr[j] and arr[j + 1]
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }

        printf("Sorted array: ");
        for (int i = 0; i < n; i++) {
            printf("%d ", arr[i]);
        }
        printf("\n");
    }

    return 0;
}

#include <stdio.h>

int main() {
    printf("Child process displaying the sorted array in reverse order:\n");
    int arr[] = {1, 2, 5, 8, 10};
    int n = sizeof(arr) / sizeof(arr[0]);

    for (int i = n - 1; i >= 0; i--) {
```

```
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```

OUTPUT:

Sorted array: 1 2 5 8 10