```
#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/wait.h>


int main(int argc, char *argv[]) {

    if (argc < 2) {

        printf("Usage: %s <number_of_elements> <elements...>\n", argv[0]);

        return 1;

    }


    int n = atoi(argv[1]);


    if (argc != n + 2) {

        printf("Error: Number of elements does not match the arguments provided.\n");

        printf("Usage: %s <number_of_elements> <elements...>\n", argv[0]);

        return 1;

    }


    int arr[n];


    for (int i = 0; i < n; i++) {

        arr[i] = atoi(argv[i + 2]);

    }


    pid_t pid = fork();
```

```c
    if (pid < 0) {
        perror("Fork failed");
        return 1;
    }
    else if (pid == 0) {

        printf("Child process (PID: %d): Reversed array:\n", getpid());
        for (int i = n - 1; i >= 0; i--) {
            printf("%d ", arr[i]);
        }
        printf("\n");
    }
    else {

        wait(NULL);
        printf("Parent process (PID: %d): Child process finished.\n", getpid());
    }

    return 0;
}
```

OUTPUT:

```
avcoe@avcoe-Vostro-3710:~$ gcc CLA.c
avcoe@avcoe-Vostro-3710:~$ ./a.out 5 10 20 30 40 50
Child process (PID: 5806): Reversed array:50 40 30 20 10
Parent process (PID: 5805): Child process finished.
```