Language: C

Open in editor

```c
#include <assert.h>
#include <ctype.h>
#include <limits.h>
#include <math.h>
#include <stdbool.h>
#include <stddef.h>
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

char* readline();
char* ltrim(char*);
char* rtrim(char*);
char** split_string(char*);

int parse_int(char*);

int* reverseArray(int a_count, int* a, int* result_count) {
    *result_count = a_count;
    int* reversed = malloc(a_count * sizeof(int));
    for (int i = 0; i < a_count; i++) {
        reversed[i] = a[a_count - 1 - i];
    }
    return reversed;
}

int main() {
    FILE* fptr = fopen(getenv("OUTPUT_PATH"), "w");

    int arr_count = parse_int(ltrim(rtrim(readline())));

    char** arr_temp = split_string(rtrim(readline()));

    int* arr = malloc(arr_count * sizeof(int));

    for (int i = 0; i < arr_count; i++) {
        int arr_item = parse_int(*(arr_temp + i));
        *(arr + i) = arr_item;
    }

    int res_count;
    int* res = reverseArray(arr_count, arr, &res_count);

    for (int i = 0; i < res_count; i++) {
        fprintf(fptr, "%d", *(res + i));
        if (i != res_count - 1) {
            fprintf(fptr, " ");
        }
    }

    fprintf(fptr, "\n");

    fclose(fptr);
    free(arr);
    free(res);
    return 0;
}

char* readline() {
    size_t alloc_length = 1024;
    size_t data_length = 0;
    char* data = malloc(alloc_length);
```

```c

    while (true) {
        char* cursor = data + data_length;
        char* line = fgets(cursor, alloc_length - data_length, stdin);

        if (!line) {
            break;
        }

        data_length += strlen(cursor);

        if (data_length < alloc_length - 1 || data[data_length - 1] == '\n') {
            break;
        }

        alloc_length <<= 1;
        data = realloc(data, alloc_length);

        if (!data) {
            data = '\0';
            break;
        }
    }

    if (data[data_length - 1] == '\n') {
        data[data_length - 1] = '\0';
        data = realloc(data, data_length);
    } else {
        data = realloc(data, data_length + 1);
        if (data) {
            data[data_length] = '\0';
        }
    }

    return data;
}

char* ltrim(char* str) {
    if (!str) return '\0';
    while (*str != '\0' && isspace(*str)) {
        str++;
    }
    return str;
}

char* rtrim(char* str) {
    if (!str) return '\0';
    char* end = str + strlen(str) - 1;

    while (end >= str && isspace(*end)) {
        end--;
    }

    *(end + 1) = '\0';
    return str;
}

char** split_string(char* str) {
    char** splits = NULL;
    char* token = strtok(str, " ");
    int spaces = 0;

    while (token) {
        splits = realloc(splits, sizeof(char*) * ++spaces);
        if (!splits) return splits;

        splits[spaces - 1] = token;
        token = strtok(NULL, " ");
    }

    return splits;
}

int parse_int(char* str) {
    char* endptr;
    int value = strtol(str, &endptr, 10);
    if (endptr == str || *endptr != '\0') {
        exit(EXIT_FAILURE);
    }
    return value;
}


```