Documentation

-CED18I051

Introduction

There are 4 programs in the src folder. The folder tree is as follows:

```
.
├── Makefile
├── progprepassignment.pdf
├── Documentation.pdf
└── src
    ├── cp.c
    ├── multisort.cpp
    ├── mysort.c
    └── smartsort.cpp

1 directory, 6 files
```

cp.c: This program refers to Q1. This program is used to copy and remove files. Soft Links were used to have both the capabilities in one source code.
mysort.c: This program refers to Q2.
multisort.cpp: This program refers to Q3.
smartsort.cpp: This program refers to Q4.

Compile

Run the following command on the terminal:

```
make
```

After running the above command, the folder tree should look as follows:

```
.
├── cp
├── Makefile
├── multisort
├── mysort
├── progprepassignment.pdf
├── Documentation.pdf
├── rm
├── smartsort
└── src
    ├── cp.c
    ├── multisort.cpp
    ├── mysort.c
    └── smartsort.cpp
```

```
    1 directory, 11 files
```

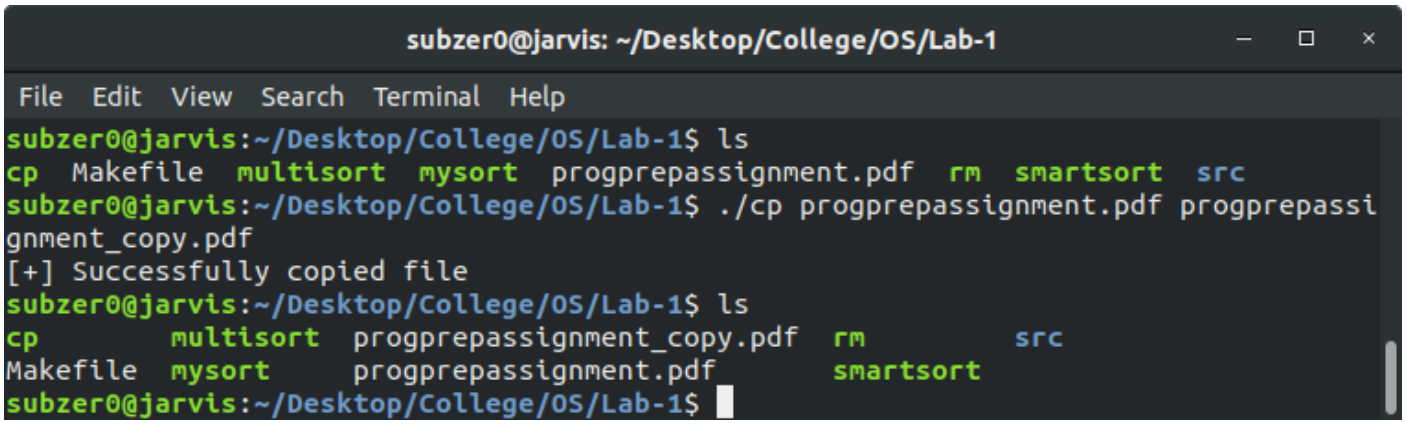
Running

Question 1:

./cp (Copy)

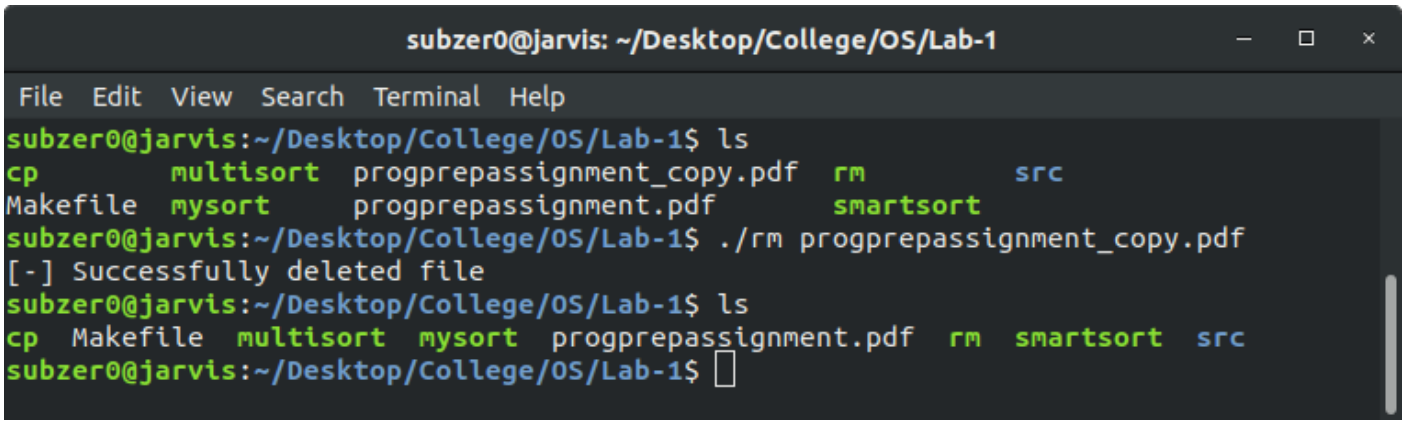
```
Syntax:

    ./cp <path-to-source-file> <path-to-source-file>
```

Since the program opens the files in binary mode, any type of file can be copied.



./rm (Delete)

```
Syntax:

    ./rm <path-to-source-file>
```

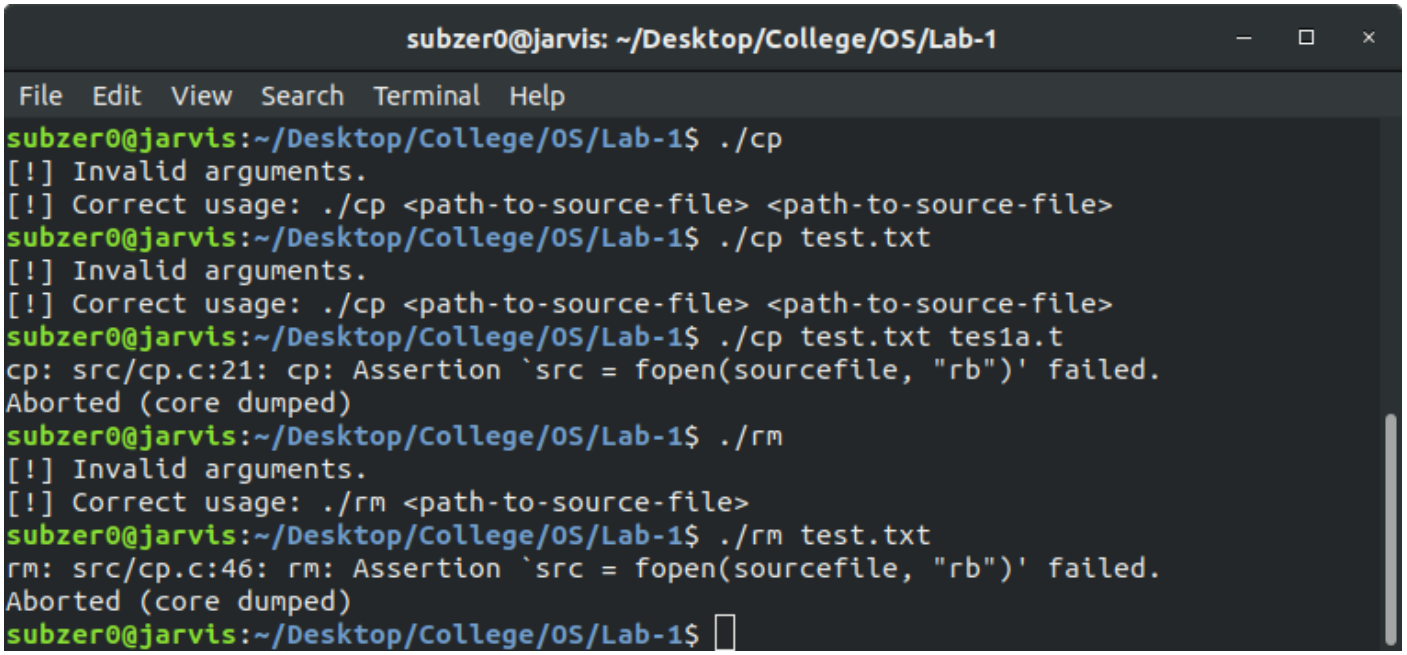


Since copy and delete are file operations, both the source codes were combined in the same source code named cp.c.

It can be observed from the above screenshots that ./cp and ./rm perform what they are meant to. *Soft links* were used to link ./rm to ./cp.

Assert statements were used to avoid the program from facing a segmentation fault runtime error.

The following are some of the validation errors outputted by the code:



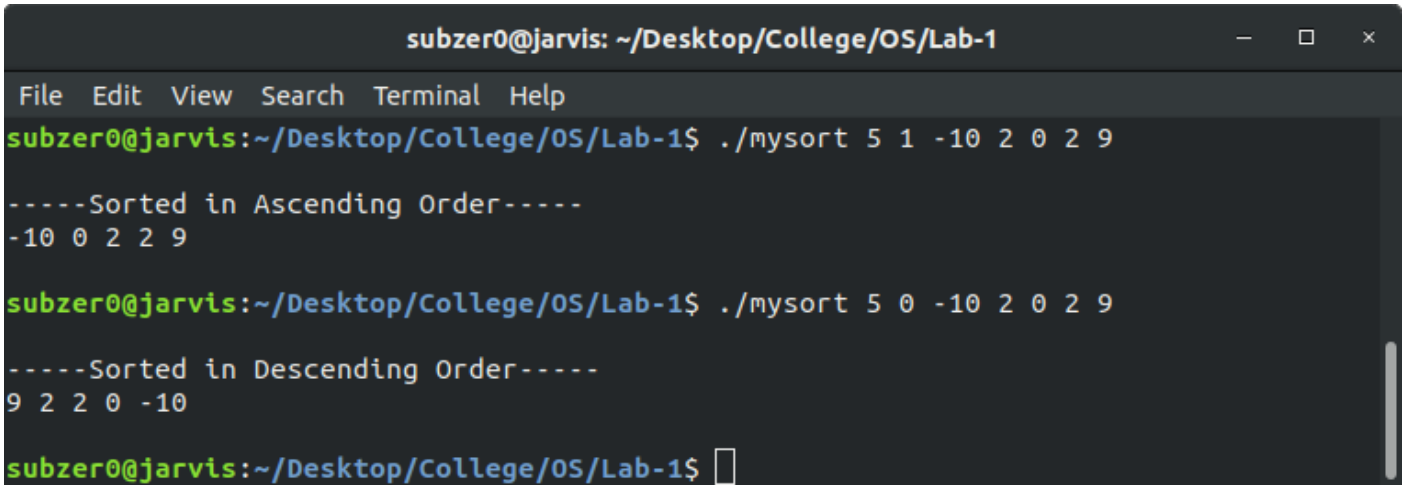
Question 2:

./mysort

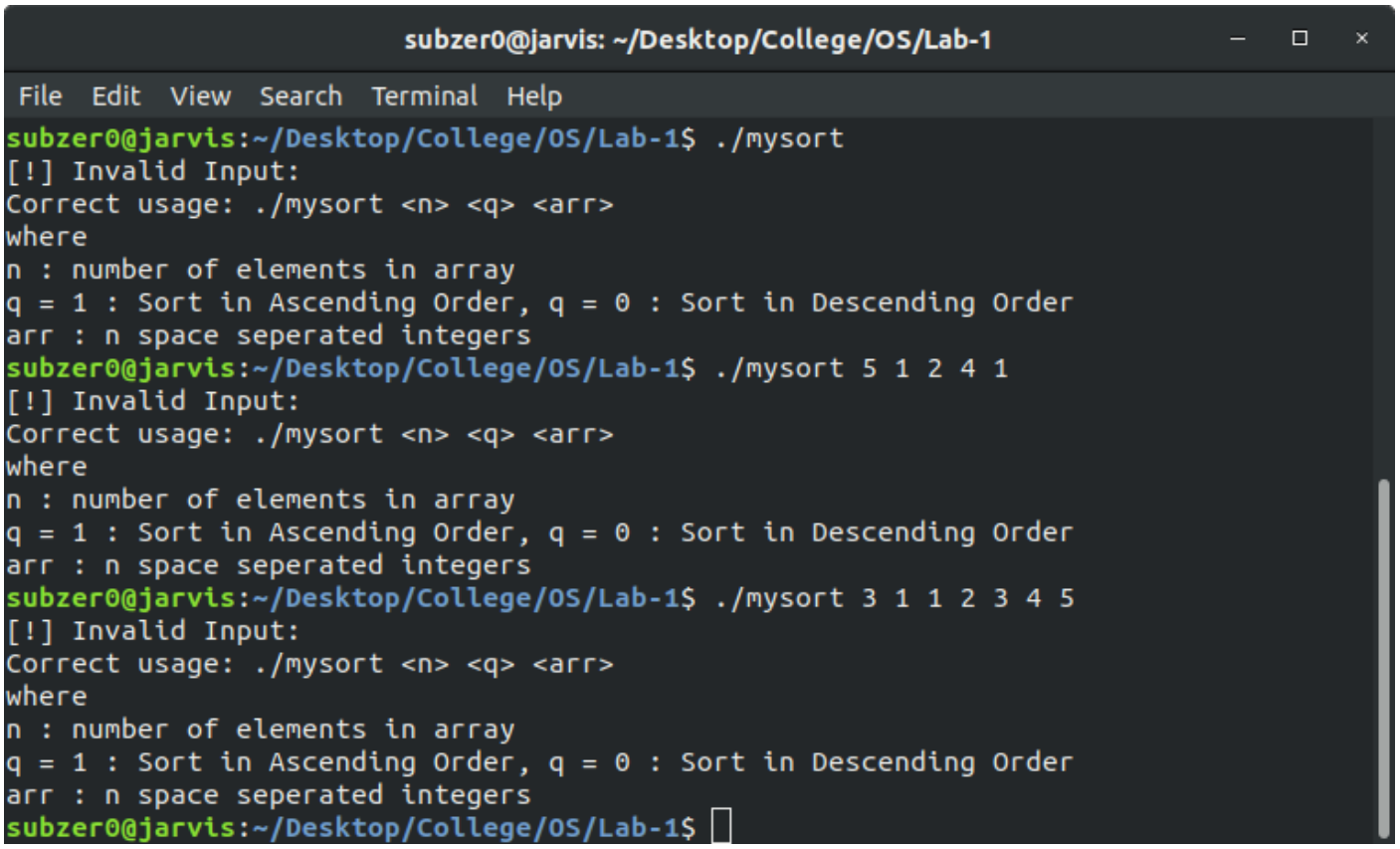
```
Syntax:

      ./mysort <n> <q> <arr>
      where
      n : number of elements in array
      q = 1 : Sort in Ascending Order, q = 0 : Sort in Descending Order
      arr : n space separated integers
```

Function pointers were used to implement the sorting functions.

The output for a valid input is as follows:



The following are some of the validation errors outputted by the code:



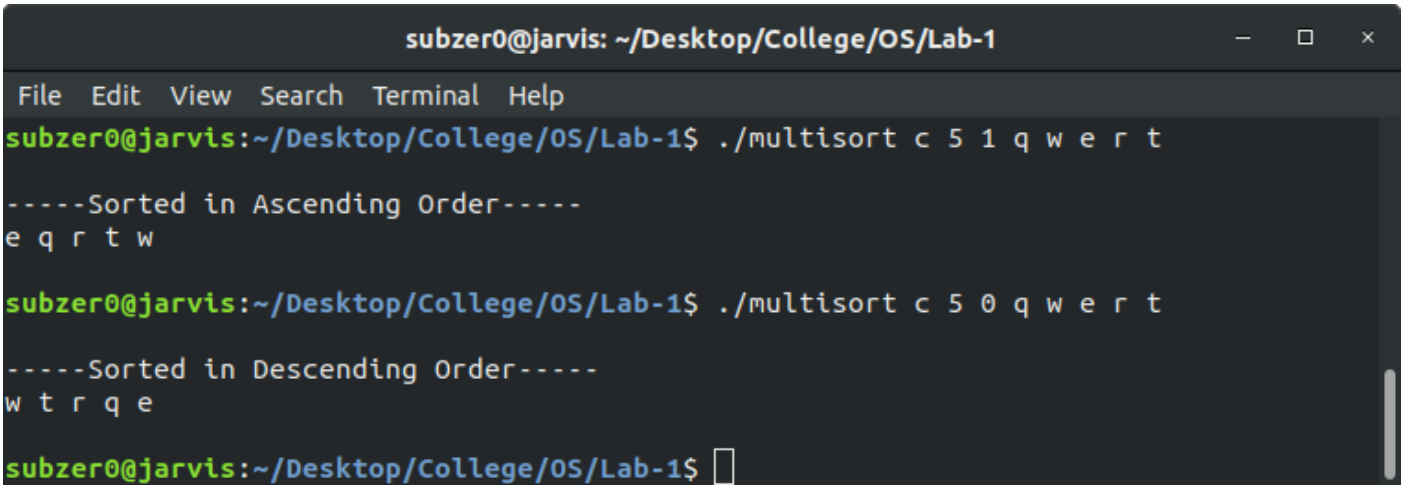
Question 3:

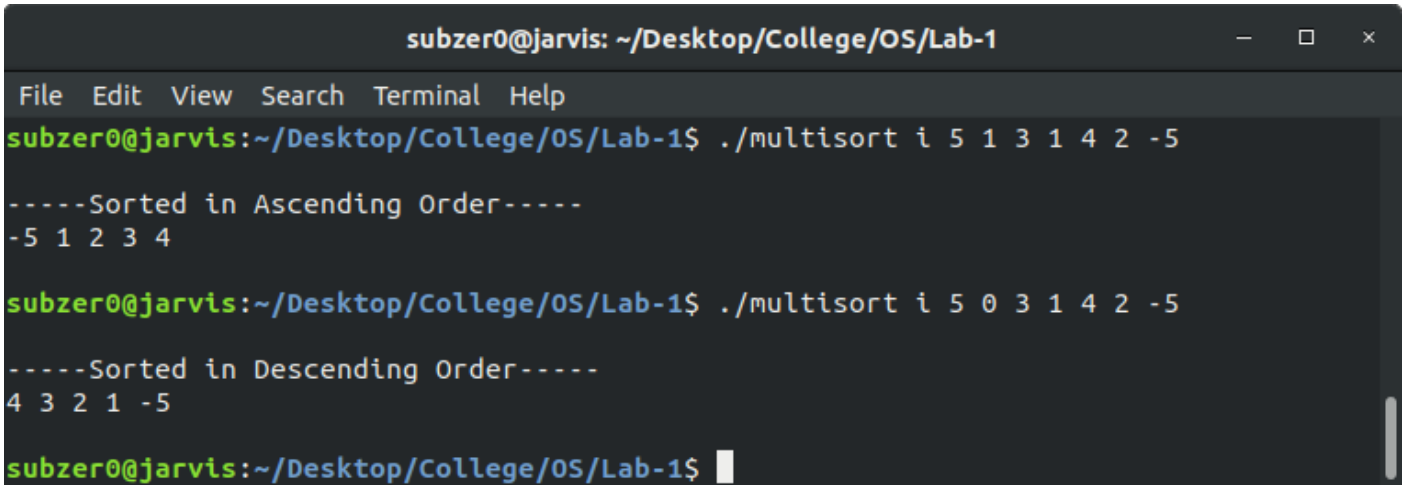
./multisort

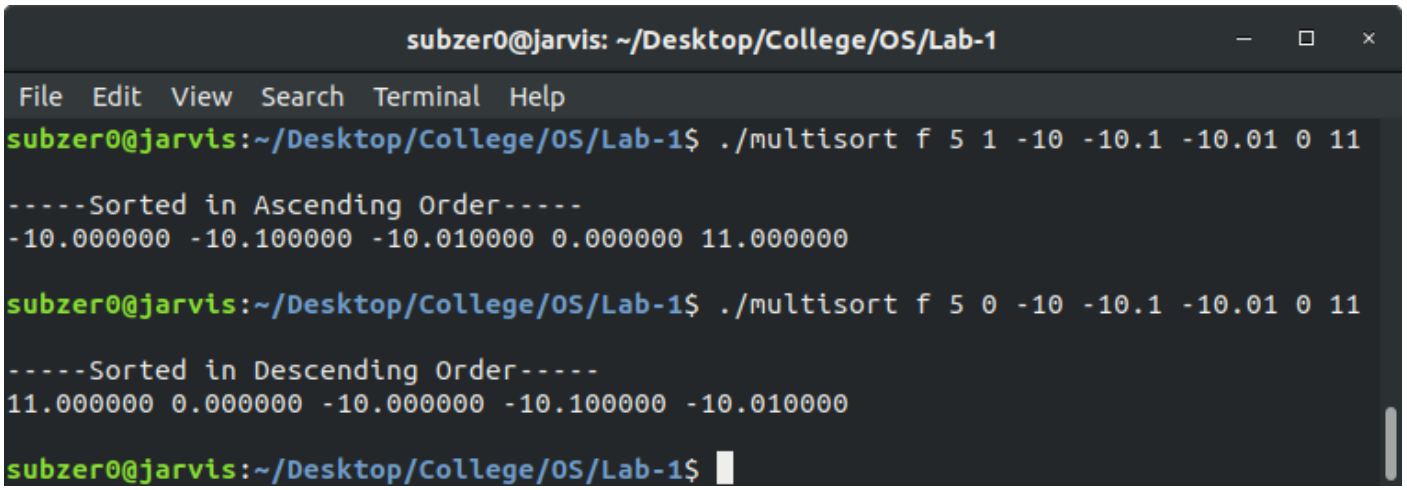
```
Syntax:

      ./mysort <t> <n> <q> <arr>
      where
      t : type of array : c - character, i - integer, f - float
      n : number of elements in array
      q = 1 : Sort in Ascending Order, q = 0 : Sort in Descending Order
      arr : n space separated integers
```

This program uses function overloading to sort arrays of different data types.

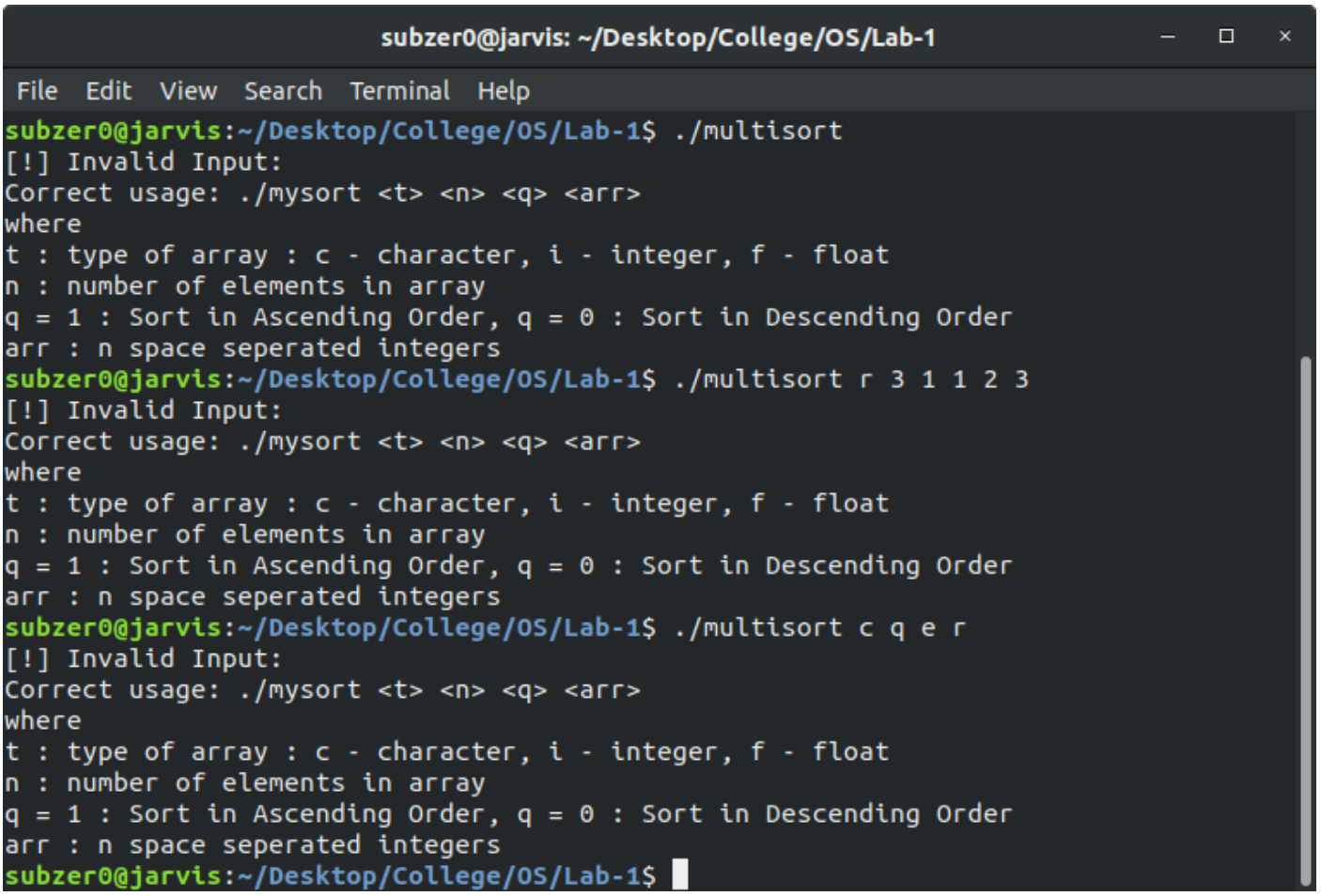
The output for some valid inputs is as follows:







The following are some of the validation errors outputted by the code:



Question 4:

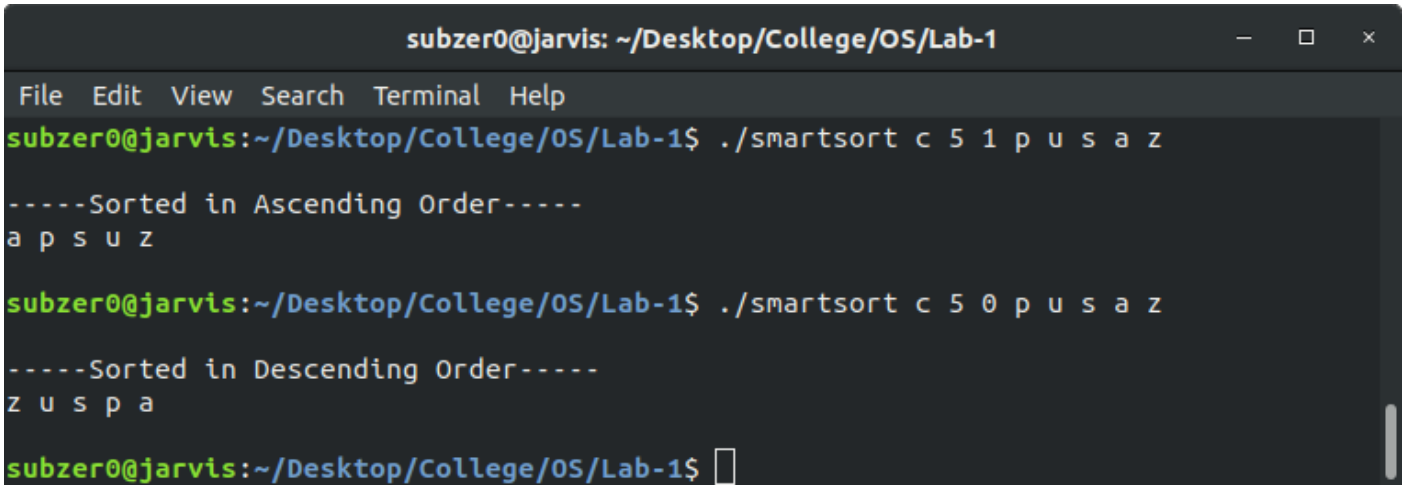
./smartsort

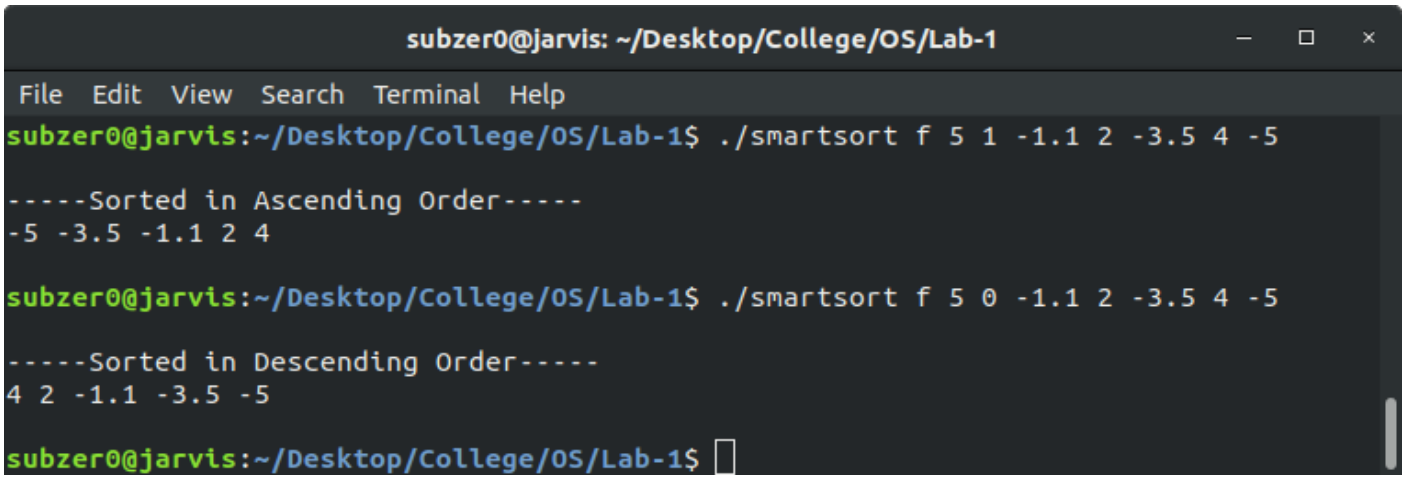
This program uses function templates to sort arrays of different data types.

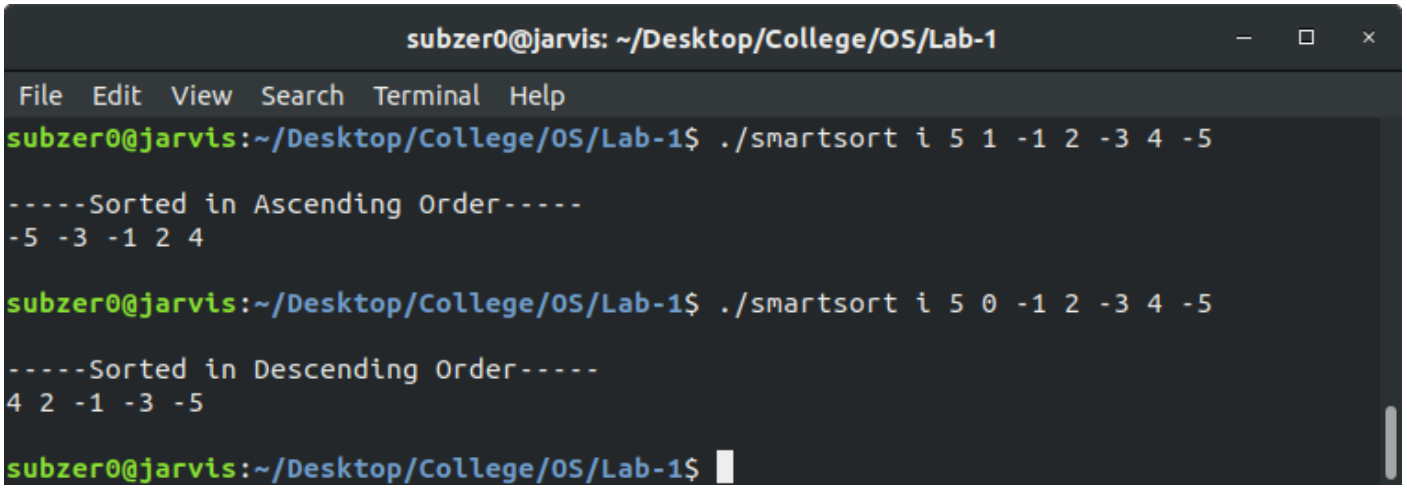
Syntax:

```
./smartsort <t> <n> <q> <arr>
where
t : type of array : c - character, i - integer, f - float
n : number of elements in array
q = 1 : Sort in Ascending Order, q = 0 : Sort in Descending Order
arr : n space separated integers
```

The output for some valid inputs is as follows:







The following are some of the validation errors outputted by the code: