CSE 344 HW1 REPORT
serdar genc 210104004023

Command formats:

gtuStudentGrades “filename<string>”
addStudentGrade “Name Surname<string>” “Grade<string>”
searchStudent “Name Surname<string>”
sortAll “filename<string>”
showAll “filename<string>”
listGrades “filename<string>”
listSome “numofEntries<int>” “pageNumber<int>” “filename<string>”
gtuStudentGrades
defaultTest
quit
Syntax Rules: Space between quotes is crucial! Space between quotes and command names are crucial! For the last 2 commands I have written, there are no whitespace characters after command names. Command names are case sensitive. Student names and grades are not case sensitive.
Accepted grades: AA, BA, BB, CB, CC, DC, DD, FF, NA, VF

gtuStudentGrades “filename<string>” sets the filename to entered parameter so to add student grade or to search a student, a filename must have been set by using gtuStudentGrades “filename<string>” command. So, the last file opened by this command will be the active file, in which you can add students or search for students.

searchStudent will print the first student that matches the name parameter with their grade.

Usage: USER MANUAL

```
Enter command: gtuStudentGrades
***USER MANUAL***
------------------------------------------------
gtuStudentGrades "filename"
This command is used to create or open a file.
NOTE!!: To add students to a file or to search a student in file first you need to open a file using this command. Changes will be done into last opened file.
------------------------------------------------
addStudentGrade "Name Surname" "Grade"
This command is used to add a student and their grade into the file.
NOTE!!: Only aa, ba, bb, cb, cc, dc, dd, ff, na, vf grades are allowed
------------------------------------------------
searchStudent "Name Surname"
This command is used to find a student and their grade from the file.
------------------------------------------------
sortAll "filename"
This command is used to sort all students and their grades inside the chosen file.
The sorting options are: 1)Name ascending       2)Name descending       3)Grade ascending       4)Grade descending
------------------------------------------------
showAll "filename"
This command is used to print all students and their grades inside the file.
------------------------------------------------
listGrades "filename"
This command is used to print first 5 students and their grades inside the file.
------------------------------------------------
listSome "numberOfEntries" "pageNumber" "filename"
This command is used to divide all students into pages each with numberOfEntries. pageNumber variable is for printing the page with entered pageNumber.
EXAMPLE!!: listSome "5" "2" "grades.txt" will list entries from 6th to 10th.
        listSome "4" "3" "grades.txt" will list entries from 9th to 12th.
------------------------------------------------
defaultTest
Enter this command to begin with default test demo.
------------------------------------------------
All error and success records are printed into the file 'event.log'
```

CREATE FILE, OPEN FILE, ADD STUDENT, SEARCH STUDENT

```
[2]+  Stopped                 ./test
serdar@serdar-VirtualBox:~/Desktop/system1$ gcc main.c -o test
serdar@serdar-VirtualBox:~/Desktop/system1$ ./test

Enter command: gtuStudentGrades "test1.txt"
        File 'test1.txt' opened successfully
Child process exited with status: 0

Enter command: addStudentGrade "Serdar genc" "ab"
        Grade is not valid.

Enter command: addStudentGrade "serdar genc" "aa"
        Student grade added successfully.
Child process exited with status: 0

Enter command: gtuStudentGrades "test2.txt"
        File 'test2.txt' opened successfully
Child process exited with status: 0

Enter command: addStudentGrade "maymun" "ff"
        Student grade added successfully.
Child process exited with status: 0

Enter command: searchStudent "serdar genc"
        Student not found
Child process exited with status: 0

Enter command: searchStudent "maymun"
        Student: maymun
        Grade:  FF
Child process exited with status: 0

Enter command: gtuStudentGrades "test1.txt"
        File 'test1.txt' opened successfully
Child process exited with status: 0

Enter command: searchStudent "serdar genc"
        Student: serdar genc
        Grade:  AA
Child process exited with status: 0

Enter command: searchStudent "maymun"
        Student not found
Child process exited with status: 0

Enter command:
```

```
 1 INPUT: gtuStudentGrades "test1.txt"
 2 File 'test1.txt' opened successfully
 3 INPUT: addStudentGrade "Serdar genc" "ab"
 4 Grade is not valid.
 5 INPUT: addStudentGrade "serdar genc" "aa"
 6 Student grade added succesfully: 'serdar genc', 'AA'.
 7 INPUT: gtuStudentGrades "test2.txt"
 8 File 'test2.txt' opened successfully
 9 INPUT: addStudentGrade "maymun" "ff"
10 Student grade added succesfully: 'maymun', 'FF'.
11 INPUT: searchStudent "serdar genc"
12 Student named 'serdar genc' is not found inside file: test2.txt.
13 INPUT: searchStudent "maymun"
14 Student named 'maymun' is found inside file: test2.txt.
15 INPUT: gtuStudentGrades "test1.txt"
16 File 'test1.txt' opened successfully
17 INPUT: searchStudent "serdar genc"
18 Student named 'serdar genc' is found inside file: test1.txt.
19 INPUT: searchStudent "maymun"
20 Student named 'maymun' is not found inside file: test1.txt.
```

As you can see by using gtuStudentGrades “filename<string>” you can create files and switch active files as well.

SORTALL FOR ALL SORT CHOICES

Name ascending:

```
Select Sort Type:
        1)Name ascending        2)Name descending
        3)Grade ascending       4)Grade descending
1
0:  student p,  DD
1:  student o,  AA
2:  student n,  BB
3:  student m,  BA
4:  student l,  FF
5:  student k,  CA
6:  student j,  CC
7:  student i,  AA
8:  student h,  BA
9:  student g,  VF
10:  student f,  NA
11:  student e,  FF
12:  student d,  BB
13:  student c,  BA
14:  student b,  CB
15:  student a,  CC
Child process exited with status: 0
```

Name descending:

```
Select Sort Type:
        1)Name ascending        2)Name descending
        3)Grade ascending       4)Grade descending
2
0:  student a,  CC
1:  student b,  CB
2:  student c,  BA
3:  student d,  BB
4:  student e,  FF
5:  student f,  NA
6:  student g,  VF
7:  student h,  BA
8:  student i,  AA
9:  student j,  CC
10:  student k,  CA
11:  student l,  FF
12:  student m,  BA
13:  student n,  BB
14:  student o,  AA
15:  student p,  DD
Child process exited with status: 0
```

Grade ascending:

```
Select Sort Type:
        1)Name ascending        2)Name descending
        3)Grade ascending       4)Grade descending
3
0:  student g,  VF
1:  student f,  NA
2:  student e,  FF
3:  student l,  FF
4:  student p,  DD
5:  student a,  CC
6:  student j,  CC
7:  student b,  CB
8:  student k,  CA
9:  student d,  BB
10:  student n,  BB
11:  student c,  BA
12:  student h,  BA
13:  student m,  BA
14:  student i,  AA
15:  student o,  AA
Child process exited with status: 0
```

Grade descending:

```
Child process exited with status: 0
Select Sort Type:
        1)Name ascending        2)Name descending
        3)Grade ascending       4)Grade descending
4
0:  student i,  AA
1:  student o,  AA
2:  student c,  BA
3:  student h,  BA
4:  student m,  BA
5:  student d,  BB
6:  student n,  BB
7:  student k,  CA
8:  student b,  CB
9:  student a,  CC
10:  student j,  CC
11:  student p,  DD
12:  student e,  FF
13:  student l,  FF
14:  student f,  NA
15:  student g,  VF
Child process exited with status: 0
```

SHOWALL, LISTGRADES, LISTSOME

SHOWALL:

```
student a, CC
student b, CB
student c, BA
student d, BB
student e, FF
student f, NA
student g, VF
student h, BA
student i, AA
student j, CC
student k, CA
student l, FF
student m, BA
student n, BB
student o, AA
student p, DD
Child process exited with status: 0
```

LISTGRADES:

```
List first 5 grades
--------------------------------------------------
student a, CC
student b, CB
student c, BA
student d, BB
student e, FF
Child process exited with status: 0
```

LISTSOME for inputs 5 1, 8 2, 4, 3

5 1 means every page has 5 index and we want the 1st page. This means that we want to print from 1st student to 5th student. Which are the students a, b, c, d and e

```
student a, CC
student b, CB
student c, BA
student d, BB
student e, FF
Child process exited with status: 0
```

8 2 means every page has 8 index and we want the 2nd page. This means that we want to print from 9th student to 16th student. Which are the students i, j, k, l, m, n, o and p

```
student i, AA
student j, CC
student k, CA
student l, FF
student m, BA
student n, BB
student o, AA
student p, DD
Child process exited with status: 0
```

4 3 means every page has 4 index and we want the 3rd page. Which means that we want to print from 9th student to 12th student. Which are the students i, j, k and l

```
student i, AA
student j, CC
student k, CA
student l, FF
Child process exited with status: 0
```

These outputs are generated by my test demo:

```
} else if (strcmp(token, "defaultTest") == 0) { //default test demo
  printf("Create file 'default.txt'\n------------------------------------------------\n");
  sleep(2);
  createFile("default.txt");
  printf("Add 16 students with grades\n------------------------------------------------\n");
  sleep(2);
  addStudentGrade("default.txt", "student a", "CC");
  addStudentGrade("default.txt", "student b", "CB");
  addStudentGrade("default.txt", "student c", "BA");
  addStudentGrade("default.txt", "student d", "BB");
  addStudentGrade("default.txt", "student e", "FF");
  addStudentGrade("default.txt", "student f", "NA");
  addStudentGrade("default.txt", "student g", "VF");
  addStudentGrade("default.txt", "student h", "BA");
  addStudentGrade("default.txt", "student i", "AA");
  addStudentGrade("default.txt", "student j", "CC");
  addStudentGrade("default.txt", "student k", "CA");
  addStudentGrade("default.txt", "student l", "FF");
  addStudentGrade("default.txt", "student m", "BA");
  addStudentGrade("default.txt", "student n", "BB");
  addStudentGrade("default.txt", "student o", "AA");
  addStudentGrade("default.txt", "student p", "DD");
  printf("Search for students: a h and p\n------------------------------------------------\n");
  sleep(2);
  searchStudent("default.txt", "student a");
  searchStudent("default.txt", "student h");
  searchStudent("default.txt", "student p");
  printf("Sort students, choose every option.\n------------------------------------------------\n");
  sleep(2);
  sortAll("default.txt");
  sortAll("default.txt");
  sortAll("default.txt");
  sortAll("default.txt");
  showAll("default.txt");
  printf("List first 5 grades\n------------------------------------------------\n");
  sleep(2);
  listGrades("default.txt");
  printf("listSome 5 1, listSome 8 2, listSome 4 3\n------------------------------------------------\n");
  sleep(2);
  listSome(5, 1, "default.txt");
  listSome(8, 2, "default.txt");
  listSome(4, 3, "default.txt");
  logFile("DEMO FINISHED");
```

Log File

```
 1 INPUT: defaultTest
 2 File 'default.txt' opened successfully
 3 Student grade added succesfully: 'student a', 'CC'.
 4 Student grade added succesfully: 'student b', 'CB'.
 5 Student grade added succesfully: 'student c', 'BA'.
 6 Student grade added succesfully: 'student d', 'BB'.
 7 Student grade added succesfully: 'student e', 'FF'.
 8 Student grade added succesfully: 'student f', 'NA'.
 9 Student grade added succesfully: 'student g', 'VF'.
10 Student grade added succesfully: 'student h', 'BA'.
11 Student grade added succesfully: 'student i', 'AA'.
12 Student grade added succesfully: 'student j', 'CC'.
13 Student grade added succesfully: 'student k', 'CA'.
14 Student grade added succesfully: 'student l', 'FF'.
15 Student grade added succesfully: 'student m', 'BA'.
16 Student grade added succesfully: 'student n', 'BB'.
17 Student grade added succesfully: 'student o', 'AA'.
18 Student grade added succesfully: 'student p', 'DD'.
19 Student named 'student a' is found inside file: default.txt.
20 Student named 'student h' is found inside file: default.txt.
21 Student named 'student p' is found inside file: default.txt.
22 Getting sort type from user...
23 Sorted by 'Name' 'Ascending'.
24 Getting sort type from user...
25 Sorted by 'Name' 'Descending'.
26 Getting sort type from user...
27 Sorted by 'Grade' 'Ascending'.
28 Getting sort type from user...
29 Sorted by 'Grade' 'Descending'.
30 All students insde 'default.txt' are listed succesfully.
31 First 5 students in 'default.txt' are listed succesfully.
32 In 'default.txt' students from 1th to 5th entries are listed.
33 In 'default.txt' students from 9th to 16th entries are listed.
34 In 'default.txt' students from 9th to 12th entries are listed.
35 DEMO FINISHED
```

Log file shows errors, successful operations, any signals interrupting the program and user inputs.