CSE 344

System Programming

1st Assignment

Report

Author

Mert Emir ŞEKER

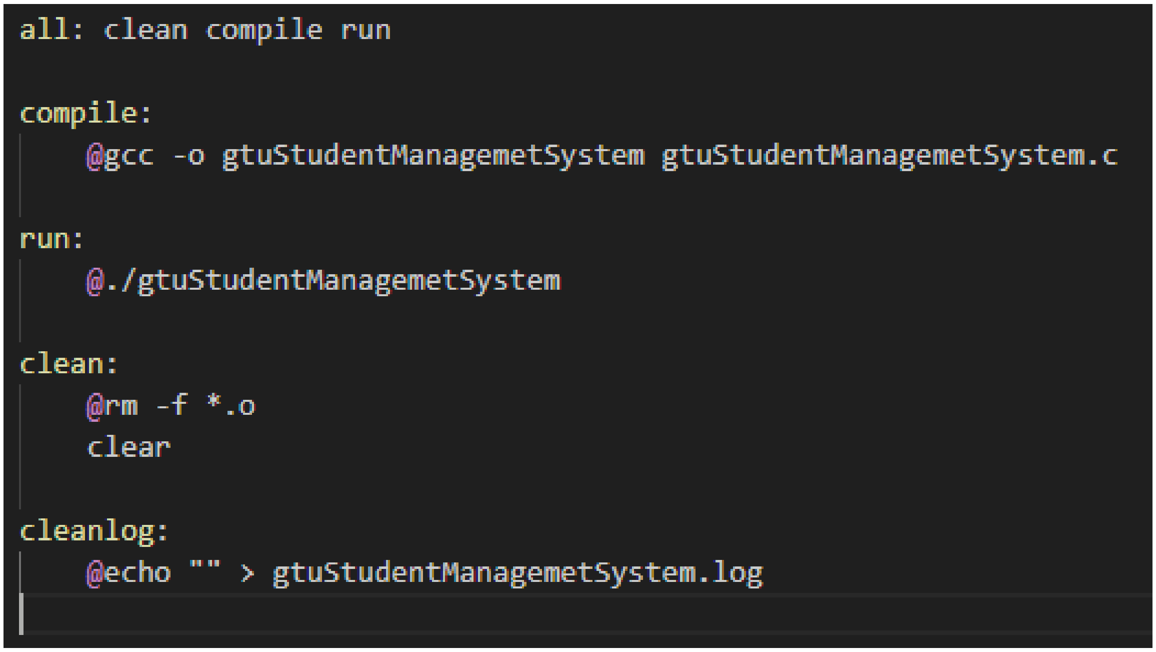
200104004085

Date

22.03.2024

Table Of Contents

Makefile



```
all: clean compile run

compile:
    @gcc -o gtuStudentManagemetSystem gtuStudentManagemetSystem.c

run:
    @./gtuStudentManagemetSystem

clean:
    @rm -f *.o
    clear

cleanlog:
    @echo "" > gtuStudentManagemetSystem.log
```

Makefile commands:

- **all**: Cleans, compiles, and runs the program.
- **compile**: Compiles the source file into an executable.
- **run**: Executes the compiled program.
- **clean**: Clears the terminal.
- **cleanlog**: Empties the log file.

Available Commands

a) q

Exits the program.

b) help

help command is a command that allows you to see all available commands.

c) gtuStudentGrades “filename”

gtuStudentGrades command takes the filename parameter between the two quotes and creates a file with that name

d) addStudentGrade “filename” “Name Surname” “Grade”

addStudentGrade command takes the parameters filename, name, surname and grade between two quotes and writes the name, surname and grade to the file in the filename. If the name already in the file is entered, it updates the grade.

e) searchStudent “filename” “Name Surname”

searchStudent command takes the parameters filename, name, surname between the two quotes and searches for name, surname in the filename file. if found, it prints the name, surname and grade.

f) sortAll “filename”

sortAll command takes the parameter filename between two quotes and gives four options for sorting the names, surnames and degrees in this file. When the user selects one of them, the sorting according to the selected option is printed to the terminal.

Sort Menu

1. Sort by Name Ascending
2. Sort by Name Descending
3. Sort by Grade Ascending
4. Sort by Grade Descending

g) showAll “filename”

showAll command takes the filename parameter between two quotes and prints all first names, surnames and notes in the file to the terminal.

h) listGrades “filename”

listGrades command takes the filename parameter between two quotes and prints all first names, surnames and notes in the first 5 lines of the file to the terminal.

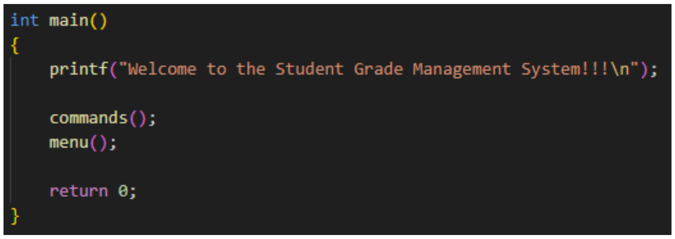
i) listSome “filename” “numOfEntries” “pageNumber”

The listSome command takes the filename numOfEntries and pageNumber parameters and splits the contents of a specified file into pages, displaying a selected number of entries from a given page number.

Functions

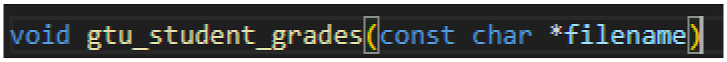
a) main

First, main function displays a welcome message and suppresses the available commands with the commands() function and waits for the user to enter commands with the menu() function.



```
int main()
{
    printf("Welcome to the Student Grade Management System!!!\n");

    commands();
    menu();

    return 0;
}
```

b) gtu_student_grades

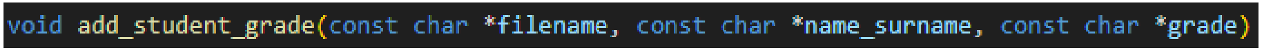


```
void gtu_student_grades(const char *filename)
```

gtu_student_grades function, takes one parameter (filename) and aims to create a new file for recording student grades. It initiates this process by generating a new child process through fork(). If fork() fails, an error is logged and the operation halts with an EXIT_FAILURE status. In the successfully created child process, the function attempts to open the specified file using the open() system call with flags set for writing, creation, and exclusive access. Failure to open the file triggers logging of the incident and termination of the process. Upon successful file creation, the child process concludes with an EXIT_SUCCESS status. Concurrently, the parent process employs waitpid() to monitor the child's completion, logging the outcome based on whether the file creation was successful or not, thereby ensuring proper synchronization and handling of the file creation task within the system.

c) add_student_grades

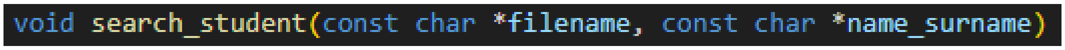
add_student_grade function, takes three parameters (filename, name_surname, and grade) and aims to insert or modify a student's grade in a file. It begins by spawning a new process using fork(). If fork() encounters an issue, it logs the error and terminates the execution with EXIT_FAILURE. Within the child process, the function opens the specified file in read-only mode using the open() system call to scan for the student's existing record. If the student's entry is located, the grade is updated; if not, a new record is appended. This modification involves reading the file content into a buffer, updating this buffer as necessary, and then writing the updated content back to the file with the write() system call. Should file opening or writing fail at any point, the function records the error and exits. After updating the file, it closes the file descriptor using close() and the child process exits successfully. Meanwhile, the parent process employs wait or waitpid() to ensure the child process completes its task before proceeding, ensuring data integrity and synchronicity between processes.



```
void add_student_grade(const char *filename, const char *name_surname, const char *grade)
```

d) search_student

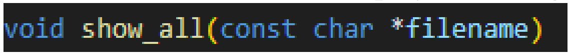
search_student function, takes two parameters (filename and name_surname) and searches for a student's name in a file. It initiates this process by creating a new child process with the fork() system call. If fork() fails, it logs an error and exits with EXIT_FAILURE. In the child process, it opens the file in read-only mode using the open() system call. On failure to open the file, it logs the error and exits. The function then uses the read system call to read() the file byte by byte, assembling lines to search for the specified name_surname. If found, it prints the line, logs the success, and exits. If the student's name isn't found after reading the entire file, it logs this outcome. Finally, it closes the file with the close system call and exits cleanly. Meanwhile, the parent process waits for the child to complete using waitpid(), ensuring proper synchronization and error handling.



```
void search_student(const char *filename, const char *name_surname)
```

e) show_all

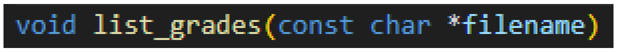
show_all function, takes one parameter (filename) and displays the contents of the specified file. It begins by creating a new child process using fork(). If fork() fails, it logs the error and exits the program to prevent further execution. In the child process, the file is opened in read-only mode with the open() system call. If opening the file fails, an error is logged, and the process exits. The function then reads the file byte by byte using read() system call, outputting its contents to standard output. It also checks if the file is empty and logs a message if so. After reading, the file descriptor is closed, and the child process exits successfully. The parent process waits for the child process to complete, ensuring that the file's contents are displayed and logs a message indicating successful execution.



```
void show_all(const char *filename)
```

f) list_grades

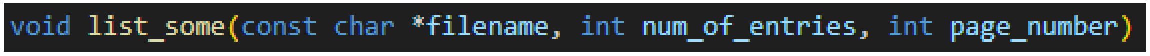
list_grades function takes one parameter (filename) and displays the first five lines from a file, starting by creating a new process with fork(). If fork() fails, it logs the error and exits. In the child process, the file is opened in read-only mode with open() system call. Failure to open triggers an error log and an exit. It reads the file byte by byte with read() system call, printing characters to stdout, stopping after five lines or if the file ends. An empty file check is performed, logging if the file is empty. After reading, it closes the file with close() and exits. The parent process waits for the child with wait(), logging completion once done.



```
void list_grades(const char *filename)
```

g) list_some

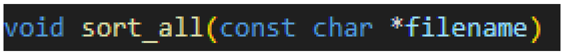
list_some function, takes three parameters (filename, num_of_entries, page_number) and displays a specific range of lines. It starts with fork() to create a new process, logging and exiting if this fails. In the child process, it opens the file in read-only mode with open(). Failure to open leads to error logging and exit. It calculates the range of lines to display based on num_of_entries and page_number, then reads the file byte by byte with read(), printing the characters within the targeted range to stdout. It flags an empty file or a request beyond the file's content, logging accordingly. After processing, it closes the file with close() and exits. The parent process waits with wait(), ensuring orderly execution and logs upon successful display or when attempting to access beyond content.



```
void list_some(const char *filename, int num_of_entries, int page_number)
```

h) sort_all

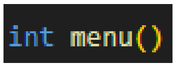
sort_all function takes one parameter (filename) and displays a sorting menu.The user's selection, captured with scanf(), determines which sorting function is called: ascending or descending by name or grade. If the choice is invalid, it alerts the user and returns to the main menu.



```
void sort_all(const char *filename)
```

i) menu

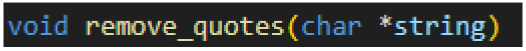
The menu function serves as the main interaction point for users, offering various commands to manage student grades within a file. Upon entering a command, the function compares it to known commands like "q" for quitting, "help" for listing commands, and several actions like "gtuStudentGrades", "addStudentGrade", "searchStudent", "showAll", "listGrades", "listSome", and "sortAll".



```
int menu()
```

j) remove_quotes

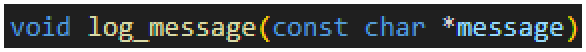
remove_quotes function processes strings, specifically targeting those inputs from the terminal like filenames, names, surnames, and grades, and removes any double quote (") characters they contain.



```
void remove_quotes(char *string)
```

k) log_message

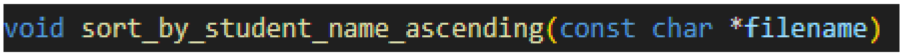
log_message function writes a message to a log file, creating a new process for this task. If the fork() fails, it exits with an error. The child process opens the log file in append mode, writes the message, and then exits successfully, while the parent waits for the child to ensure sequential logging. Errors in file operations lead to immediate exit with failure in the child process.



```
void log_message(const char *message)
```

l) sort_by_student_name_ascending

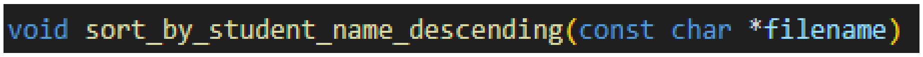
sort_by_student_name_ascending function takes one parameter (filename) and sorts a file's contents by student names in ascending order, leveraging a new process created with fork(). If fork() fails, it logs this via log_message and exits. The child process opens the file open() system call in read-only mode, reads its content into a buffer with read() system call, and null-terminates it. It parses the buffer into lines, sorting them alphabetically with a bubble sort algorithm, specifically comparing student names using strcmp. After sorting, it logs the success and prints the sorted lines. The child then exits, and the parent process waits wait() for completion, ensuring a tidy execution flow.



```
void sort_by_student_name_ascending(const char *filename)
```

m) sort_by_student_name_descending

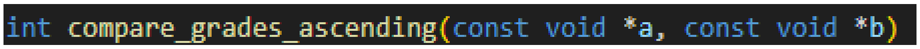
sort_by_student_name_descending function takes one parameter (filename) and sorts a file's contents by student names in descending order, starting with a fork() to create a child process. If fork() fails, it logs the error and exits. In the child process, the file is opened for reading with open() system call. If opening fails, it logs the issue and exits. The function reads the file into a buffer using read() system call, ensuring it's null-terminated. It then splits the buffer into lines, sorting them in descending order by comparing student names with strcmp and swapping as necessary. After sorting, it logs the operation's success and prints the sorted lines before exiting. The parent process waits for the child's completion with wait(), ensuring the process flow is managed efficiently.



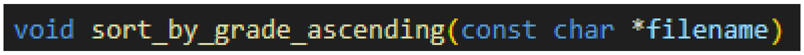
```
void sort_by_student_name_descending(const char *filename)
```

n) compare_grades_ascending and sort_by_grade_ascending

compare_grades_ascending function, designed for use with qsort(), takes two parameters (a,b) and compares student grades in ascending order by mapping each grade to a predefined order from "NA" to "AA". It extracts the grade part from strings using strrchr to find the last occurrence of a space, determining the grade's index within the order array. sort_by_grade_ascending function takes one parameter (filename) and sorts the contents of a file specified by filename based on these grades. It opens the file with open() system call, reads the content into a buffer with read(), ensuring the read content is null-terminated, then closes the file with close(). It splits the content into lines, sorts them using qsort with the compare_grades_ascending comparison function, and prints the sorted lines, effectively organizing the grades in ascending order while logging the operation's success.



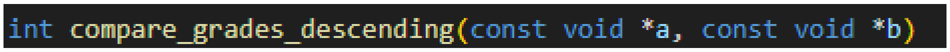
```
int compare_grades_ascending(const void *a, const void *b)
```



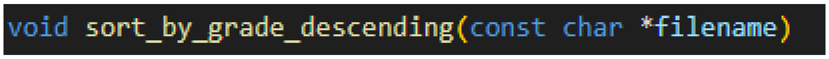
```
void sort_by_grade_ascending(const char *filename)
```

o) compare_grades_ascending and sort_by_grade_descending

compare_grades_descending function, tailored for integration with qsort, takes two parameters (a,b) and compares student grades for sorting in descending order, utilizing a reversed grade order from "AA" to "NA". It identifies each grade's position in this array to determine their comparative ranks. sort_by_grade_descending function takes one parameter (filename) and sorts a file's content by grades in descending order, beginning with opening the file using open(), reading its content into a buffer with read(), and ensuring the buffer is null-terminated before closing the file with close(). It divides the buffer into lines, sorts them using qsort with the compare_grades_descending comparator, logs the successful sort operation, and prints the sorted lines, efficiently organizing the grades from highest to lowest.



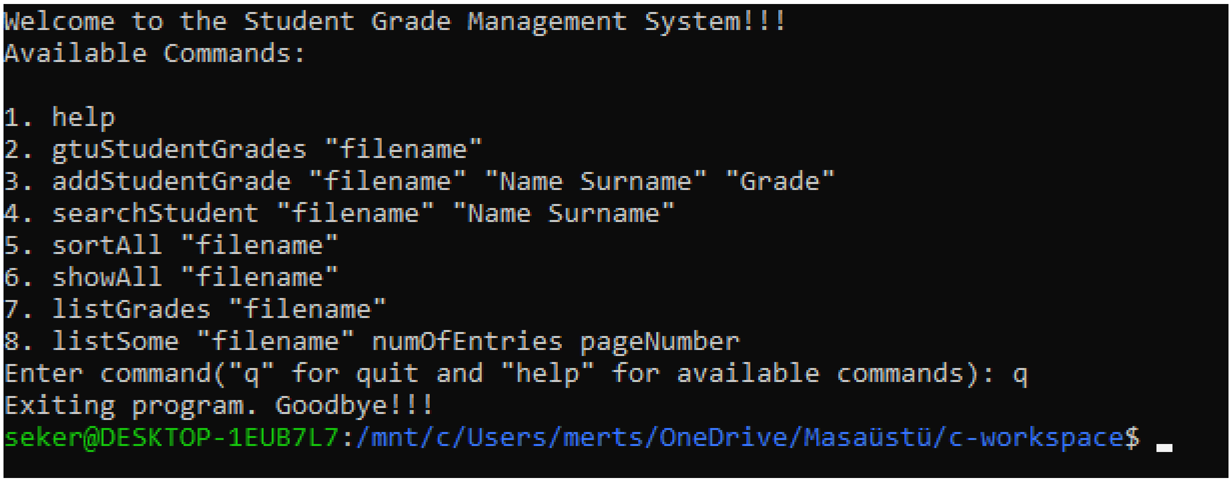
```
int compare_grades_descending(const void *a, const void *b)
```



```
void sort_by_grade_descending(const char *filename)
```

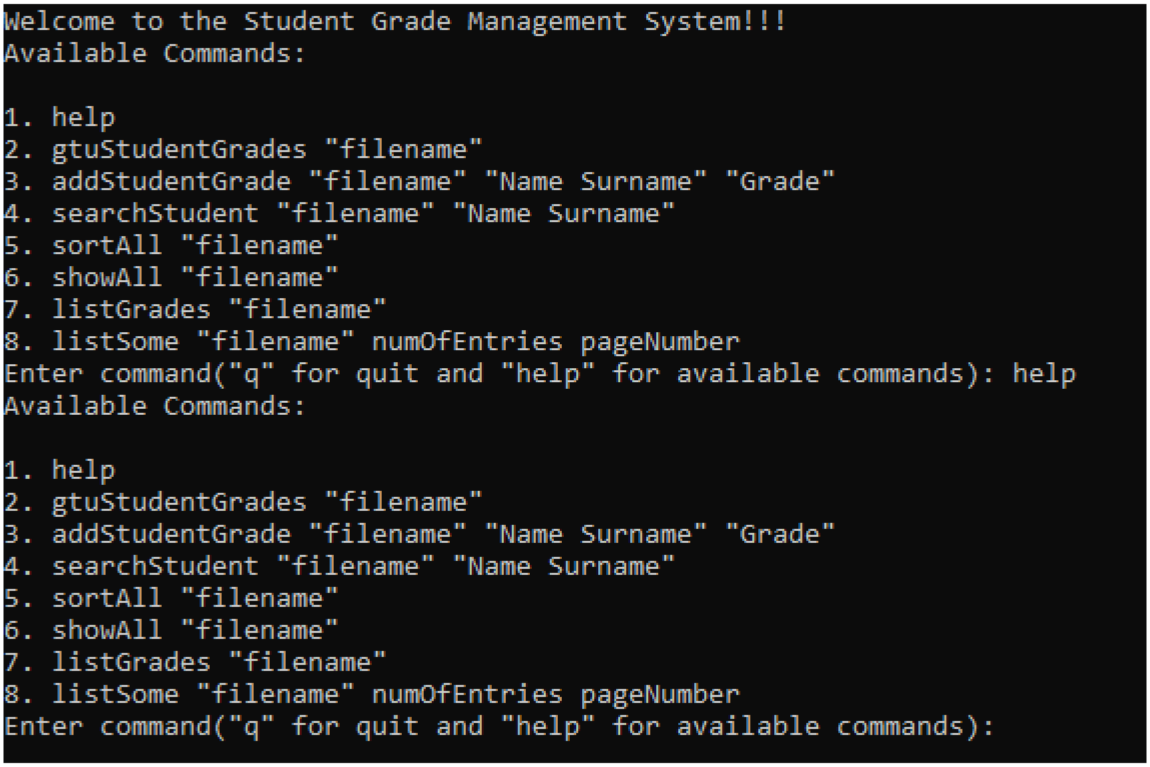
Testing Commands

1) q



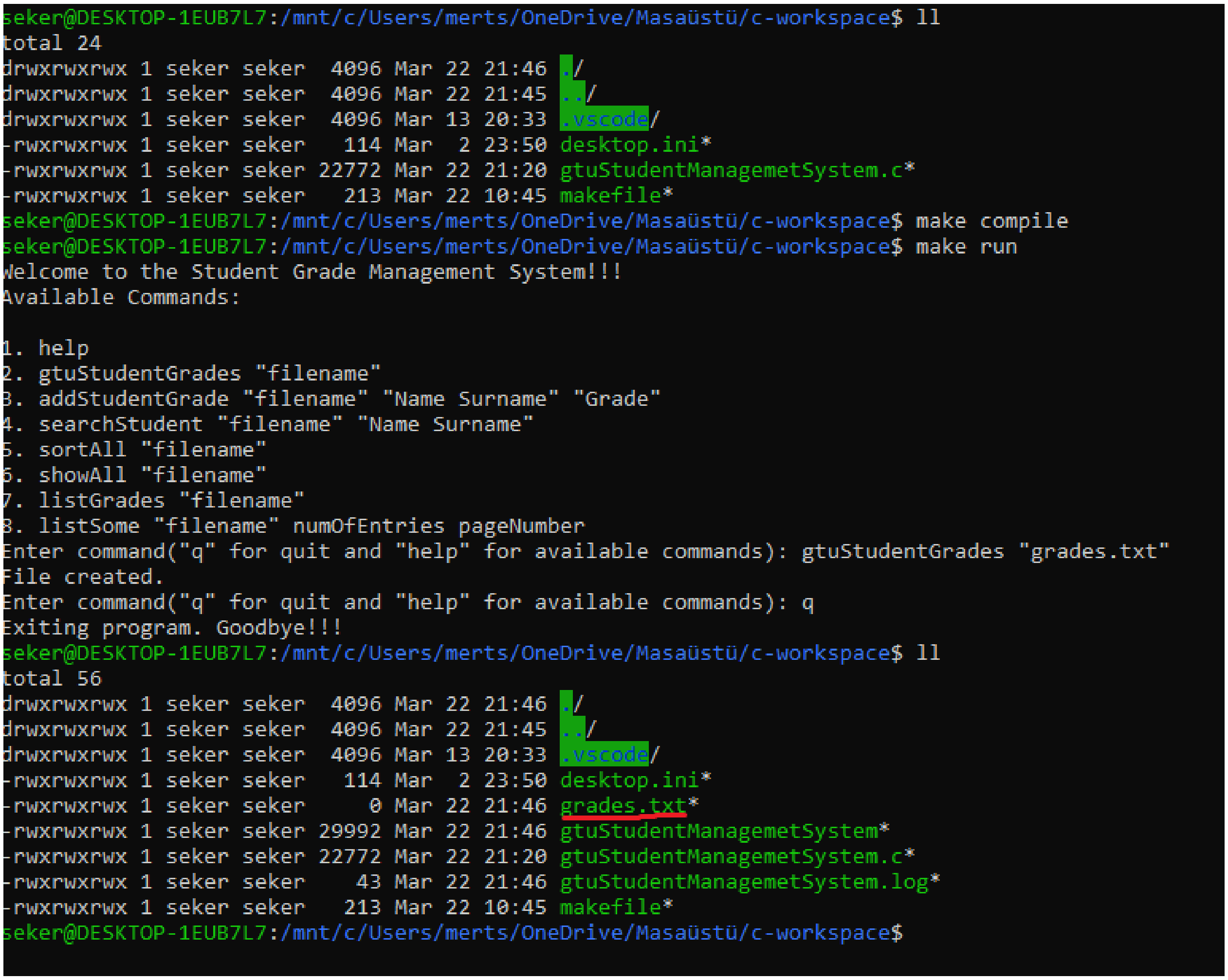
```
Welcome to the Student Grade Management System!!!
Available Commands:

1. help
2. gtuStudentGrades "filename"
3. addStudentGrade "filename" "Name Surname" "Grade"
4. searchStudent "filename" "Name Surname"
5. sortAll "filename"
6. showAll "filename"
7. listGrades "filename"
8. listSome "filename" numOfEntries pageNumber
Enter command("q" for quit and "help" for available commands): q
Exiting program. Goodbye!!!
seker@DESKTOP-1EUB7L7:/mnt/c/Users/merts/OneDrive/Masaüstü/c-workspace$
```

2) help



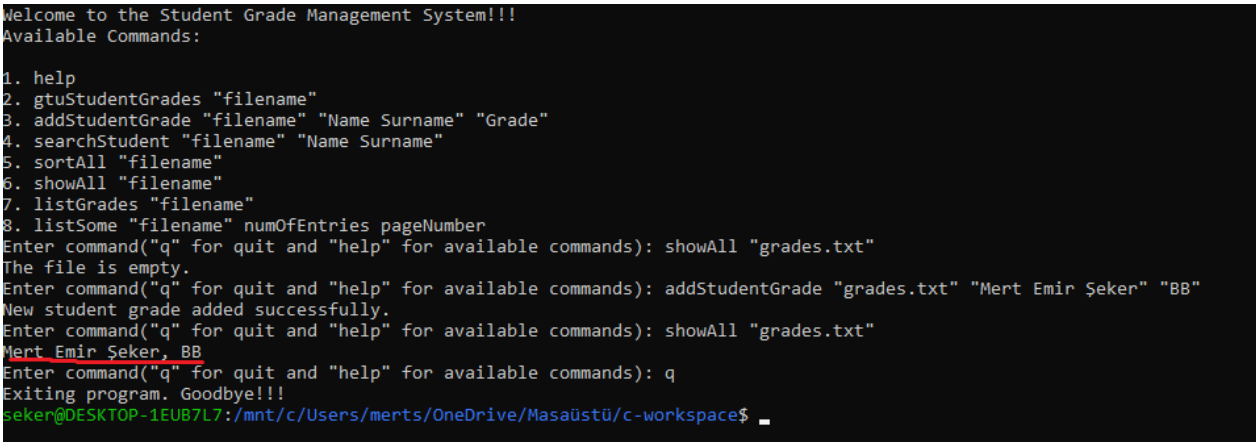
```
Welcome to the Student Grade Management System!!!
Available Commands:

1. help
2. gtuStudentGrades "filename"
3. addStudentGrade "filename" "Name Surname" "Grade"
4. searchStudent "filename" "Name Surname"
5. sortAll "filename"
6. showAll "filename"
7. listGrades "filename"
8. listSome "filename" numOfEntries pageNumber
Enter command("q" for quit and "help" for available commands): help
Available Commands:

1. help
2. gtuStudentGrades "filename"
3. addStudentGrade "filename" "Name Surname" "Grade"
4. searchStudent "filename" "Name Surname"
5. sortAll "filename"
6. showAll "filename"
7. listGrades "filename"
8. listSome "filename" numOfEntries pageNumber
Enter command("q" for quit and "help" for available commands):
```

3) gtuStudentGrades “filename”



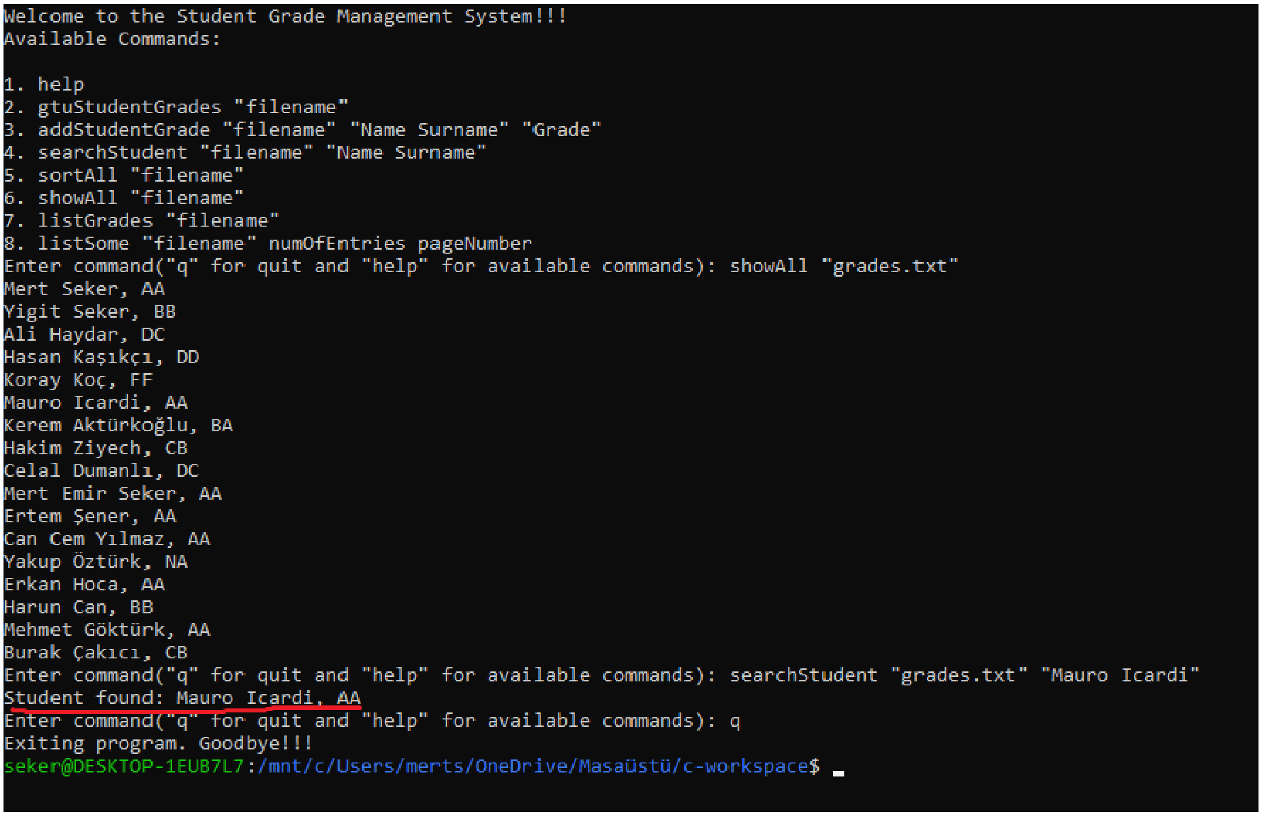
```
seker@DESKTOP-1EUB7L7:/mnt/c/Users/merts/OneDrive/Masaüstü/c-workspace$ ll
total 24
drwxrwxrwx 1 seker seker  4096 Mar 22 21:46 ./
drwxrwxrwx 1 seker seker  4096 Mar 22 21:45 ../
drwxrwxrwx 1 seker seker  4096 Mar 13 20:33 .vscode/
-rwxrwxrwx 1 seker seker   114 Mar  2 23:50 desktop.ini*
-rwxrwxrwx 1 seker seker 22772 Mar 22 21:20 gtuStudentManagemetSystem.c*
-rwxrwxrwx 1 seker seker   213 Mar 22 10:45 makefile*
seker@DESKTOP-1EUB7L7:/mnt/c/Users/merts/OneDrive/Masaüstü/c-workspace$ make compile
seker@DESKTOP-1EUB7L7:/mnt/c/Users/merts/OneDrive/Masaüstü/c-workspace$ make run
Welcome to the Student Grade Management System!!!
Available Commands:

1. help
2. gtuStudentGrades "filename"
3. addStudentGrade "filename" "Name Surname" "Grade"
4. searchStudent "filename" "Name Surname"
5. sortAll "filename"
6. showAll "filename"
7. listGrades "filename"
8. listSome "filename" numOfEntries pageNumber
Enter command("q" for quit and "help" for available commands): gtuStudentGrades "grades.txt"
File created.
Enter command("q" for quit and "help" for available commands): q
Exiting program. Goodbye!!!
seker@DESKTOP-1EUB7L7:/mnt/c/Users/merts/OneDrive/Masaüstü/c-workspace$ ll
total 56
drwxrwxrwx 1 seker seker  4096 Mar 22 21:46 ./
drwxrwxrwx 1 seker seker  4096 Mar 22 21:45 ../
drwxrwxrwx 1 seker seker  4096 Mar 13 20:33 .vscode/
-rwxrwxrwx 1 seker seker   114 Mar  2 23:50 desktop.ini*
-rwxrwxrwx 1 seker seker     0 Mar 22 21:46 grades.txt*
-rwxrwxrwx 1 seker seker 29992 Mar 22 21:46 gtuStudentManagemetSystem*
-rwxrwxrwx 1 seker seker 22772 Mar 22 21:20 gtuStudentManagemetSystem.c*
-rwxrwxrwx 1 seker seker    43 Mar 22 21:46 gtuStudentManagemetSystem.log*
-rwxrwxrwx 1 seker seker   213 Mar 22 10:45 makefile*
seker@DESKTOP-1EUB7L7:/mnt/c/Users/merts/OneDrive/Masaüstü/c-workspace$
```

4) addStudentGrade “filename” “Name Surname” “Grade”



```
Welcome to the Student Grade Management System!!!
Available Commands:

1. help
2. gtuStudentGrades "filename"
3. addStudentGrade "filename" "Name Surname" "Grade"
4. searchStudent "filename" "Name Surname"
5. sortAll "filename"
6. showAll "filename"
7. listGrades "filename"
8. listSome "filename" numOfEntries pageNumber
Enter command("q" for quit and "help" for available commands): showAll "grades.txt"
The file is empty.
Enter command("q" for quit and "help" for available commands): addStudentGrade "grades.txt" "Mert Emir Şeker" "BB"
New student grade added successfully.
Enter command("q" for quit and "help" for available commands): showAll "grades.txt"
Mert Emir Şeker, BB
Enter command("q" for quit and "help" for available commands): q
Exiting program. Goodbye!!!
seker@DESKTOP-1EUB7L7:/mnt/c/Users/merts/OneDrive/Masaüstü/c-workspace$ _
```

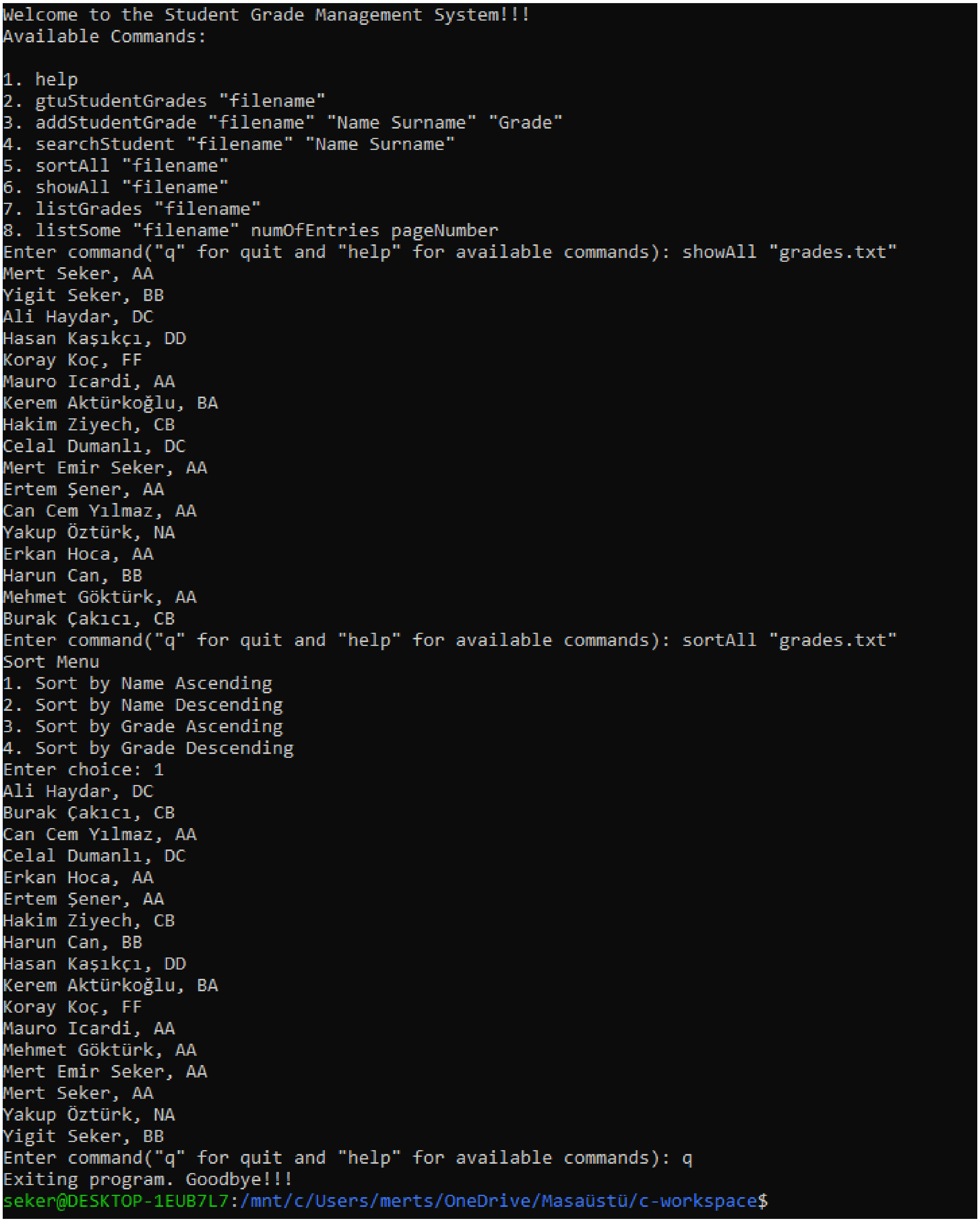
5) searchStudent “filename” “Name Surname”



```
Welcome to the Student Grade Management System!!!
Available Commands:

1. help
2. gtuStudentGrades "filename"
3. addStudentGrade "filename" "Name Surname" "Grade"
4. searchStudent "filename" "Name Surname"
5. sortAll "filename"
6. showAll "filename"
7. listGrades "filename"
8. listSome "filename" numOfEntries pageNumber
Enter command("q" for quit and "help" for available commands): showAll "grades.txt"
Mert Seker, AA
Yigit Seker, BB
Ali Haydar, DC
Hasan Kaşıkçı, DD
Koray Koç, FF
Mauro Icardi, AA
Kerem Aktürkoğlu, BA
Hakim Ziyech, CB
Celal Dumanlı, DC
Mert Emir Seker, AA
Ertem Şener, AA
Can Cem Yılmaz, AA
Yakup Öztürk, NA
Erkan Hoca, AA
Harun Can, BB
Mehmet Göktürk, AA
Burak Çakıcı, CB
Enter command("q" for quit and "help" for available commands): searchStudent "grades.txt" "Mauro Icardi"
Student found: Mauro Icardi, AA
Enter command("q" for quit and "help" for available commands): q
Exiting program. Goodbye!!!
seker@DESKTOP-1EUB7L7:/mnt/c/Users/merts/OneDrive/Masaüstü/c-workspace$ 
```

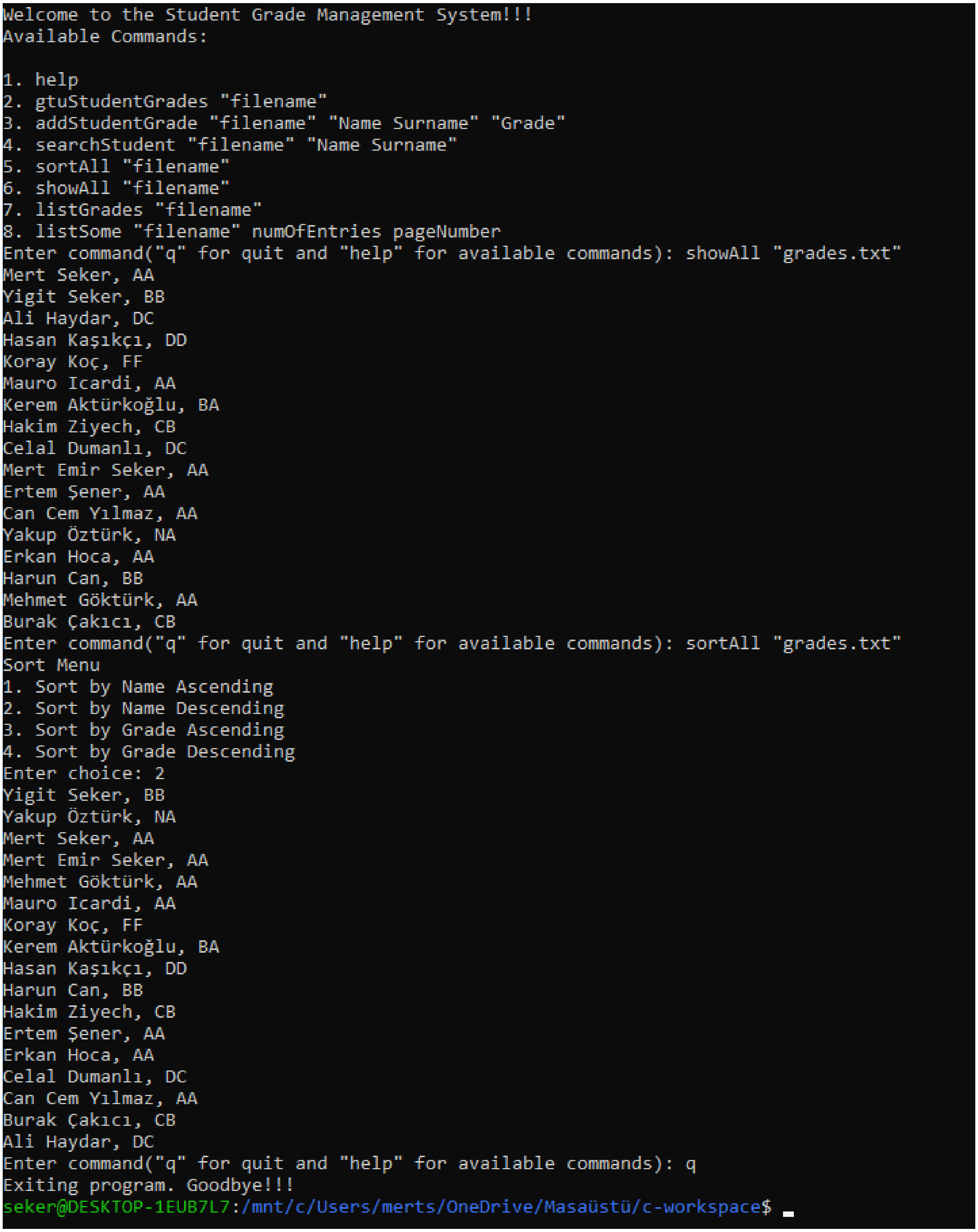
6) sortAll “filename”

a) Sort by Name Ascending



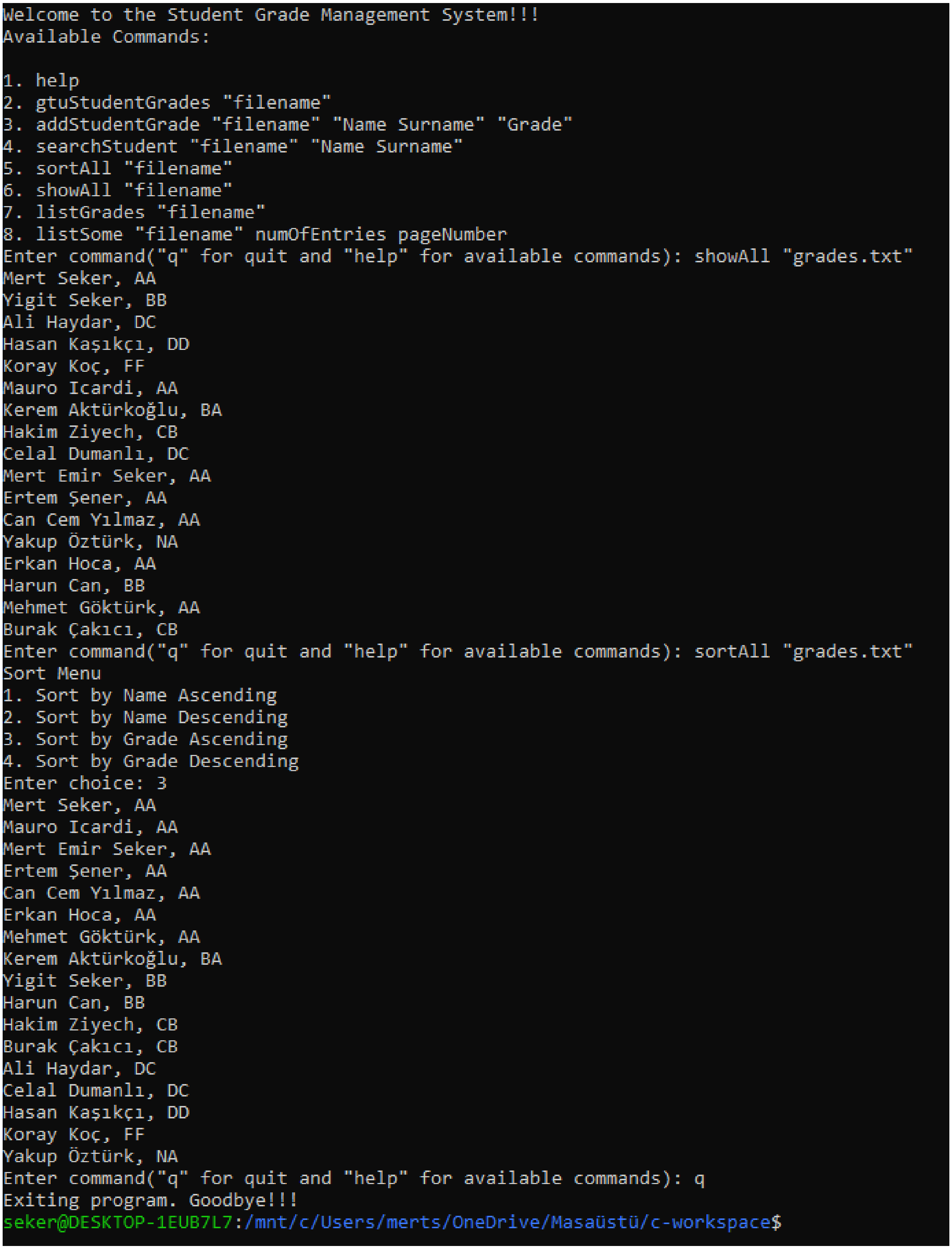
```
Welcome to the Student Grade Management System!!!
Available Commands:

1. help
2. gtuStudentGrades "filename"
3. addStudentGrade "filename" "Name Surname" "Grade"
4. searchStudent "filename" "Name Surname"
5. sortAll "filename"
6. showAll "filename"
7. listGrades "filename"
8. listSome "filename" numOfEntries pageNumber
Enter command("q" for quit and "help" for available commands): showAll "grades.txt"
Mert Seker, AA
Yigit Seker, BB
Ali Haydar, DC
Hasan Kaşıkçı, DD
Koray Koç, FF
Mauro Icardi, AA
Kerem Aktürkoğlu, BA
Hakim Ziyech, CB
Celal Dumanlı, DC
Mert Emir Seker, AA
Ertem Şener, AA
Can Cem Yılmaz, AA
Yakup Öztürk, NA
Erkan Hoca, AA
Harun Can, BB
Mehmet Göktürk, AA
Burak Çakıcı, CB
Enter command("q" for quit and "help" for available commands): sortAll "grades.txt"
Sort Menu
1. Sort by Name Ascending
2. Sort by Name Descending
3. Sort by Grade Ascending
4. Sort by Grade Descending
Enter choice: 1
Ali Haydar, DC
Burak Çakıcı, CB
Can Cem Yılmaz, AA
Celal Dumanlı, DC
Erkan Hoca, AA
Ertem Şener, AA
Hakim Ziyech, CB
Harun Can, BB
Hasan Kaşıkçı, DD
Kerem Aktürkoğlu, BA
Koray Koç, FF
Mauro Icardi, AA
Mehmet Göktürk, AA
Mert Emir Seker, AA
Mert Seker, AA
Yakup Öztürk, NA
Yigit Seker, BB
Enter command("q" for quit and "help" for available commands): q
Exiting program. Goodbye!!!
seker@DESKTOP-1EUB7L7:/mnt/c/Users/merts/OneDrive/Masaüstü/c-workspace$
```

b) Sort by Name Descending



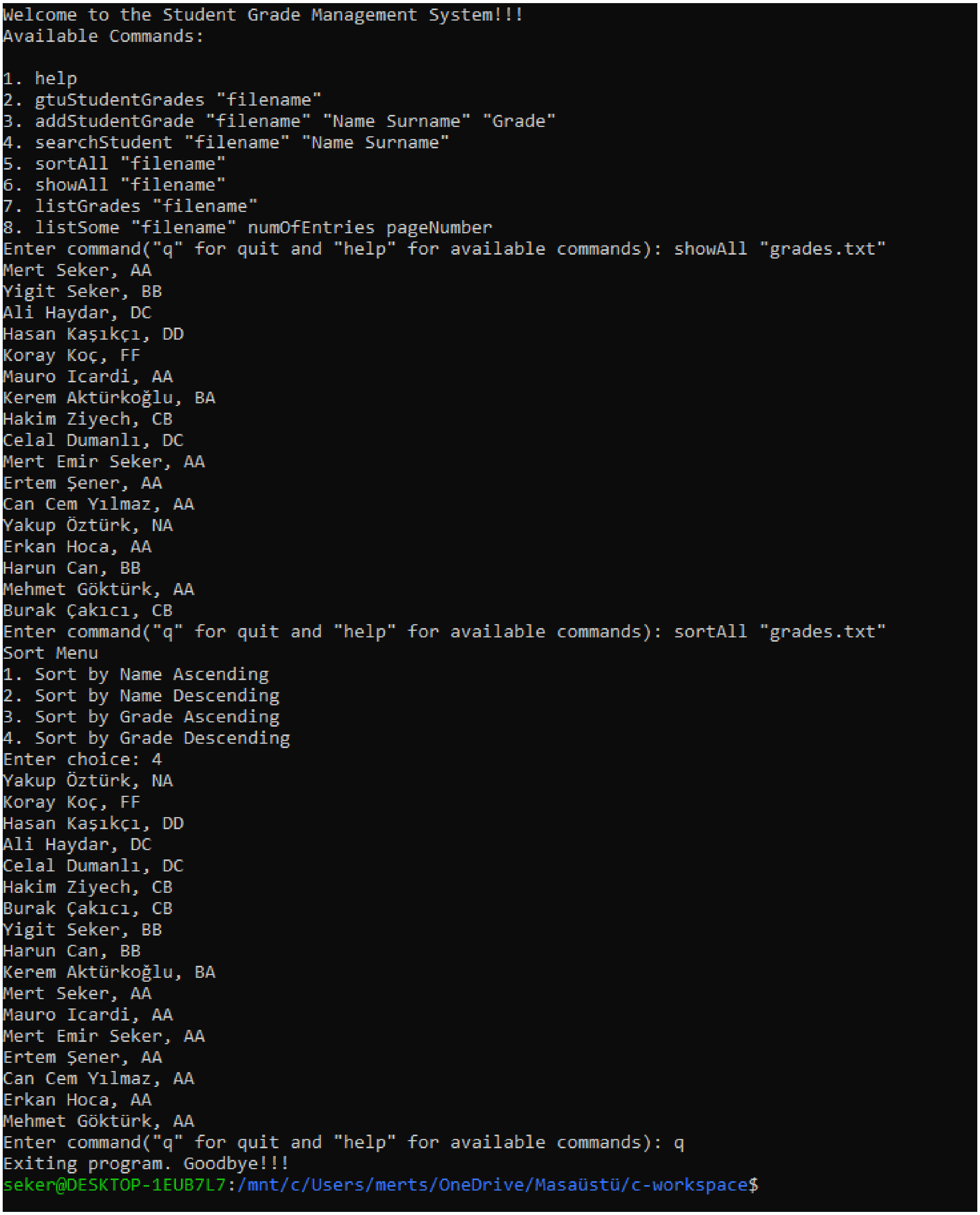
```
Welcome to the Student Grade Management System!!!
Available Commands:

1. help
2. gtuStudentGrades "filename"
3. addStudentGrade "filename" "Name Surname" "Grade"
4. searchStudent "filename" "Name Surname"
5. sortAll "filename"
6. showAll "filename"
7. listGrades "filename"
8. listSome "filename" numOfEntries pageNumber
Enter command("q" for quit and "help" for available commands): showAll "grades.txt"
Mert Seker, AA
Yigit Seker, BB
Ali Haydar, DC
Hasan Kaşıkçı, DD
Koray Koç, FF
Mauro Icardi, AA
Kerem Aktürkoğlu, BA
Hakim Ziyech, CB
Celal Dumanlı, DC
Mert Emir Seker, AA
Ertem Şener, AA
Can Cem Yılmaz, AA
Yakup Öztürk, NA
Erkan Hoca, AA
Harun Can, BB
Mehmet Göktürk, AA
Burak Çakıcı, CB
Enter command("q" for quit and "help" for available commands): sortAll "grades.txt"
Sort Menu
1. Sort by Name Ascending
2. Sort by Name Descending
3. Sort by Grade Ascending
4. Sort by Grade Descending
Enter choice: 2
Yigit Seker, BB
Yakup Öztürk, NA
Mert Seker, AA
Mert Emir Seker, AA
Mehmet Göktürk, AA
Mauro Icardi, AA
Koray Koç, FF
Kerem Aktürkoğlu, BA
Hasan Kaşıkçı, DD
Harun Can, BB
Hakim Ziyech, CB
Ertem Şener, AA
Erkan Hoca, AA
Celal Dumanlı, DC
Can Cem Yılmaz, AA
Burak Çakıcı, CB
Ali Haydar, DC
Enter command("q" for quit and "help" for available commands): q
Exiting program. Goodbye!!!
seker@DESKTOP-1EUB7L7:/mnt/c/Users/merts/OneDrive/Masaüstü/c-workspace$
```

c) Sort by Grade Ascending



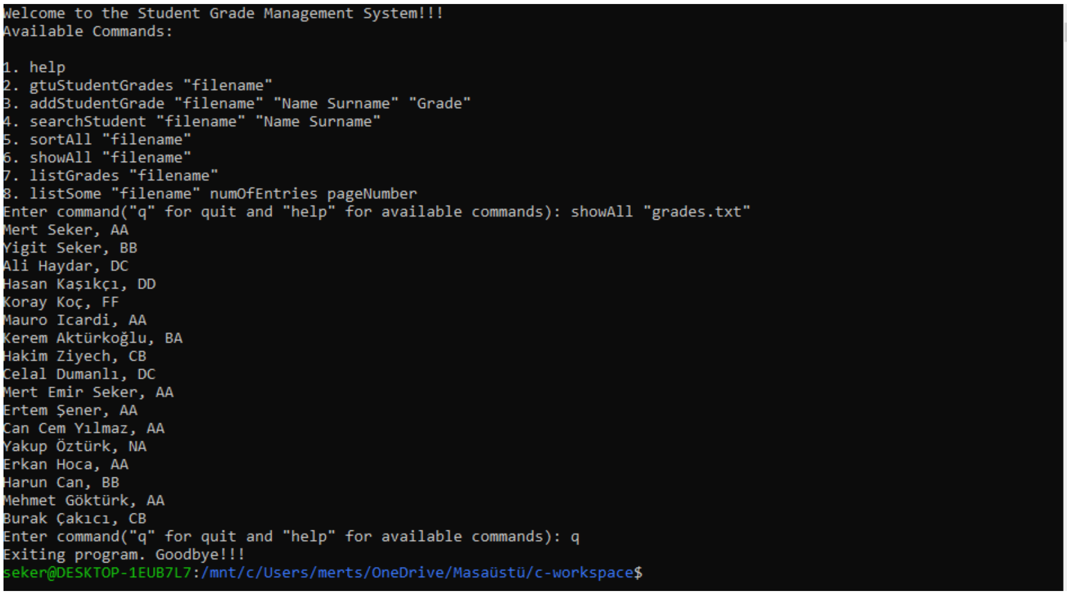
```
Welcome to the Student Grade Management System!!!
Available Commands:

1. help
2. gtuStudentGrades "filename"
3. addStudentGrade "filename" "Name Surname" "Grade"
4. searchStudent "filename" "Name Surname"
5. sortAll "filename"
6. showAll "filename"
7. listGrades "filename"
8. listSome "filename" numOfEntries pageNumber
Enter command("q" for quit and "help" for available commands): showAll "grades.txt"
Mert Seker, AA
Yigit Seker, BB
Ali Haydar, DC
Hasan Kaşıkçı, DD
Koray Koç, FF
Mauro Icardi, AA
Kerem Aktürkoğlu, BA
Hakim Ziyech, CB
Celal Dumanlı, DC
Mert Emir Seker, AA
Ertem Şener, AA
Can Cem Yılmaz, AA
Yakup Öztürk, NA
Erkan Hoca, AA
Harun Can, BB
Mehmet Göktürk, AA
Burak Çakıcı, CB
Enter command("q" for quit and "help" for available commands): sortAll "grades.txt"
Sort Menu
1. Sort by Name Ascending
2. Sort by Name Descending
3. Sort by Grade Ascending
4. Sort by Grade Descending
Enter choice: 3
Mert Seker, AA
Mauro Icardi, AA
Mert Emir Seker, AA
Ertem Şener, AA
Can Cem Yılmaz, AA
Erkan Hoca, AA
Mehmet Göktürk, AA
Kerem Aktürkoğlu, BA
Yigit Seker, BB
Harun Can, BB
Hakim Ziyech, CB
Burak Çakıcı, CB
Ali Haydar, DC
Celal Dumanlı, DC
Hasan Kaşıkçı, DD
Koray Koç, FF
Yakup Öztürk, NA
Enter command("q" for quit and "help" for available commands): q
Exiting program. Goodbye!!!
seker@DESKTOP-1EUB7L7:/mnt/c/Users/merts/OneDrive/Masaüstü/c-workspace$
```

d) Sort by Grade Descending

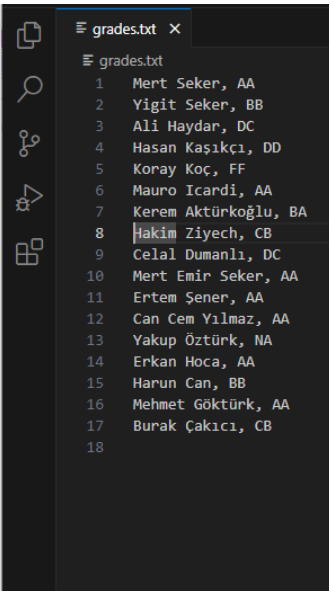


```
Welcome to the Student Grade Management System!!!
Available Commands:

1. help
2. gtuStudentGrades "filename"
3. addStudentGrade "filename" "Name Surname" "Grade"
4. searchStudent "filename" "Name Surname"
5. sortAll "filename"
6. showAll "filename"
7. listGrades "filename"
8. listSome "filename" numOfEntries pageNumber
Enter command("q" for quit and "help" for available commands): showAll "grades.txt"
Mert Seker, AA
Yigit Seker, BB
Ali Haydar, DC
Hasan Kaşıkçı, DD
Koray Koç, FF
Mauro Icardi, AA
Kerem Aktürkoğlu, BA
Hakim Ziyech, CB
Celal Dumanlı, DC
Mert Emir Seker, AA
Ertem Şener, AA
Can Cem Yılmaz, AA
Yakup Öztürk, NA
Erkan Hoca, AA
Harun Can, BB
Mehmet Göktürk, AA
Burak Çakıcı, CB
Enter command("q" for quit and "help" for available commands): sortAll "grades.txt"
Sort Menu
1. Sort by Name Ascending
2. Sort by Name Descending
3. Sort by Grade Ascending
4. Sort by Grade Descending
Enter choice: 4
Yakup Öztürk, NA
Koray Koç, FF
Hasan Kaşıkçı, DD
Ali Haydar, DC
Celal Dumanlı, DC
Hakim Ziyech, CB
Burak Çakıcı, CB
Yigit Seker, BB
Harun Can, BB
Kerem Aktürkoğlu, BA
Mert Seker, AA
Mauro Icardi, AA
Mert Emir Seker, AA
Ertem Şener, AA
Can Cem Yılmaz, AA
Erkan Hoca, AA
Mehmet Göktürk, AA
Enter command("q" for quit and "help" for available commands): q
Exiting program. Goodbye!!!
seker@DESKTOP-1EUB7L7:/mnt/c/Users/merts/OneDrive/Masaüstü/c-workspace$
```

7) showAll “filename”

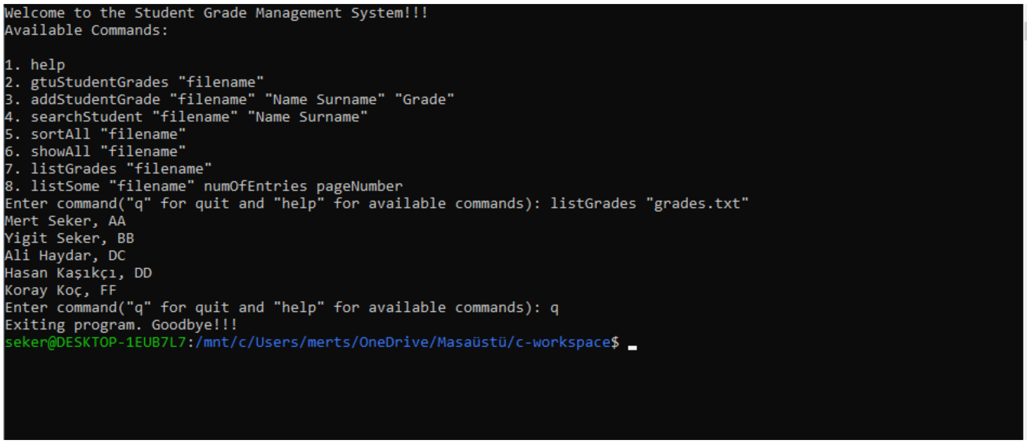


```
Welcome to the Student Grade Management System!!!
Available Commands:

1. help
2. gtuStudentGrades "filename"
3. addStudentGrade "filename" "Name Surname" "Grade"
4. searchStudent "filename" "Name Surname"
5. sortAll "filename"
6. showAll "filename"
7. listGrades "filename"
8. listSome "filename" numOfEntries pageNumber
Enter command("q" for quit and "help" for available commands): showAll "grades.txt"
Mert Seker, AA
Yigit Seker, BB
Ali Haydar, DC
Hasan Kaşıkçı, DD
Koray Koç, FF
Mauro Icardi, AA
Kerem Aktürkoğlu, BA
Hakim Ziyech, CB
Celal Dumanlı, DC
Mert Emir Seker, AA
Ertem Şener, AA
Can Cem Yılmaz, AA
Yakup Öztürk, NA
Erkan Hoca, AA
Harun Can, BB
Mehmet Göktürk, AA
Burak Çakıcı, CB
Enter command("q" for quit and "help" for available commands): q
Exiting program. Goodbye!!!
seker@DESKTOP-1EUB7L7:/mnt/c/Users/merts/OneDrive/Masaüstü/c-workspace$
```

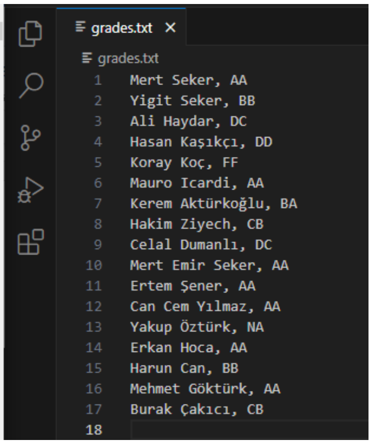


```
Mert Seker, AA
Yigit Seker, BB
Ali Haydar, DC
Hasan Kaşıkçı, DD
Koray Koç, FF
Mauro Icardi, AA
Kerem Aktürkoğlu, BA
Hakim Ziyech, CB
Celal Dumanlı, DC
Mert Emir Seker, AA
Ertem Şener, AA
Can Cem Yılmaz, AA
Yakup Öztürk, NA
Erkan Hoca, AA
Harun Can, BB
Mehmet Göktürk, AA
Burak Çakıcı, CB
```

8) listGrades “filename”

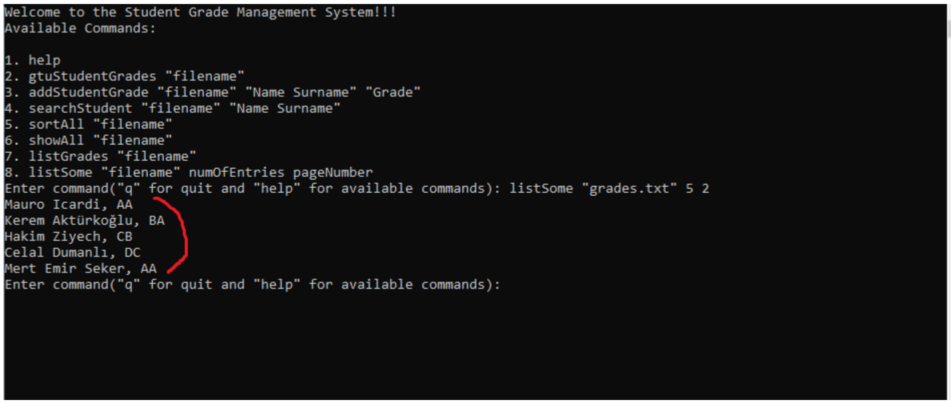


```
Welcome to the Student Grade Management System!!!
Available Commands:

1. help
2. gtuStudentGrades "filename"
3. addStudentGrade "filename" "Name Surname" "Grade"
4. searchStudent "filename" "Name Surname"
5. sortAll "filename"
6. showAll "filename"
7. listGrades "filename"
8. listSome "filename" numOfEntries pageNumber
Enter command("q" for quit and "help" for available commands): listGrades "grades.txt"
Mert Seker, AA
Yigit Seker, BB
Ali Haydar, DC
Hasan Kaşıkçı, DD
Koray Koç, FF
Enter command("q" for quit and "help" for available commands): q
Exiting program. Goodbye!!!
seker@DESKTOP-1EUB7L7:/mnt/c/Users/merts/OneDrive/Masaüstü/c-workspace$
```

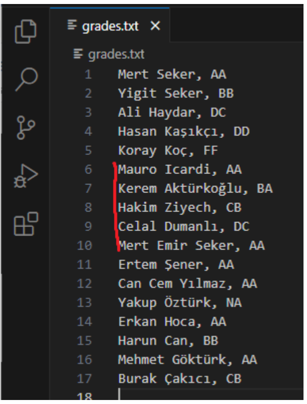


```
Mert Seker, AA
Yigit Seker, BB
Ali Haydar, DC
Hasan Kaşıkçı, DD
Koray Koç, FF
Mauro Icardi, AA
Kerem Aktürkoğlu, BA
Hakim Ziyech, CB
Celal Dumanlı, DC
Mert Emir Seker, AA
Ertem Şener, AA
Can Cem Yılmaz, AA
Yakup Öztürk, NA
Erkan Hoca, AA
Harun Can, BB
Mehmet Göktürk, AA
Burak Çakıcı, CB
```

9) listSome "filename" numOfEntries pageNumber



```
Welcome to the Student Grade Management System!!!
Available Commands:

1. help
2. gtuStudentGrades "filename"
3. addStudentGrade "filename" "Name Surname" "Grade"
4. searchStudent "filename" "Name Surname"
5. sortAll "filename"
6. showAll "filename"
7. listGrades "filename"
8. listSome "filename" numOfEntries pageNumber
Enter command("q" for quit and "help" for available commands): listSome "grades.txt" 5 2
Mauro Icardi, AA
Kerem Aktürkoğlu, BA
Hakim Ziyech, CB
Celal Dumanlı, DC
Mert Emir Seker, AA
Enter command("q" for quit and "help" for available commands):
```



grades.txt

```
Mert Seker, AA
Yigit Seker, BB
Ali Haydar, DC
Hasan Kaşıkçı, DD
Koray Koç, FF
Mauro Icardi, AA
Kerem Aktürkoğlu, BA
Hakim Ziyech, CB
Celal Dumanlı, DC
Mert Emir Seker, AA
Ertem Şener, AA
Can Cem Yılmaz, AA
Yakup Öztürk, NA
Erkan Hoca, AA
Harun Can, BB
Mehmet Göktürk, AA
Burak Çakıcı, CB
```