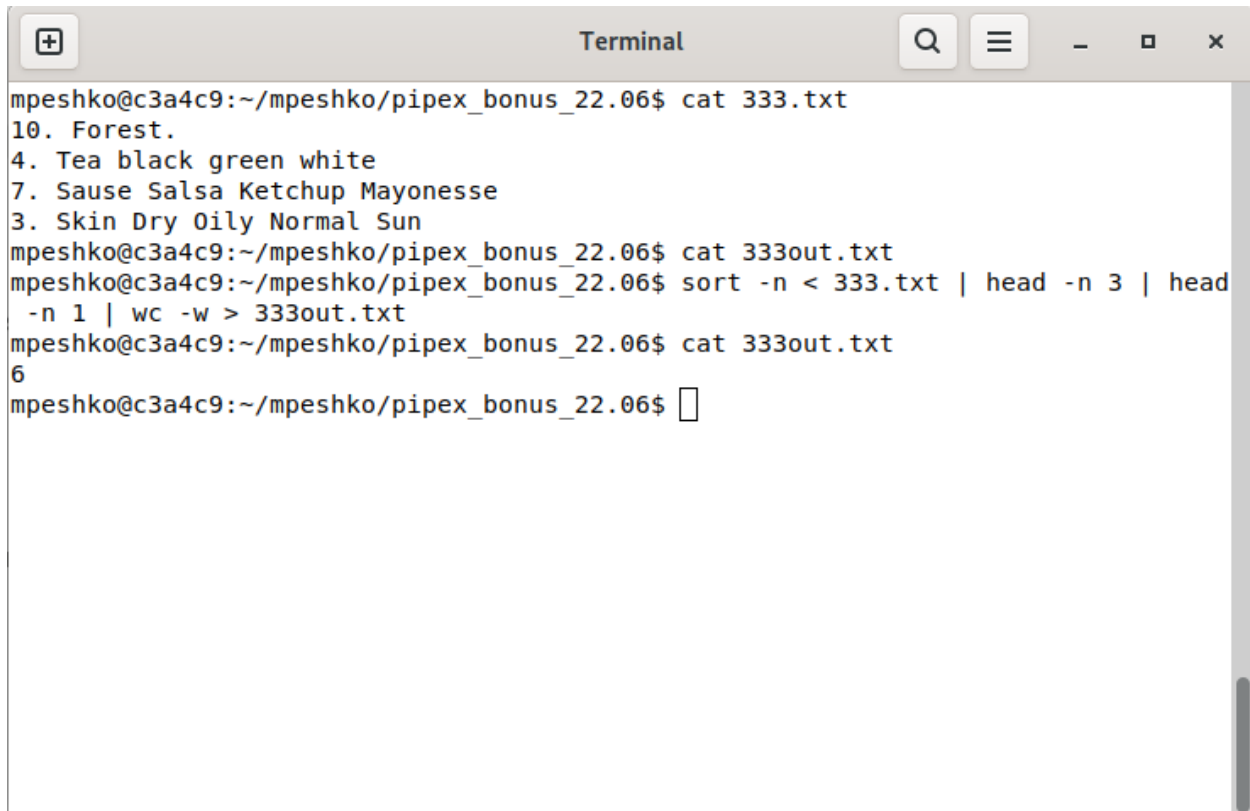


Pipex_bonus

Bash

```
sort -n < 333.txt | head -n 3 | head -n 1 | wc -w > 333out.txt
```

A terminal window titled "Terminal" with a search icon, a menu icon, and window control buttons. The terminal shows the following commands and output:

```
mpeshko@c3a4c9:~/mpeshko/pipex_bonus_22.06$ cat 333.txt
10. Forest.
4. Tea black green white
7. Sause Salsa Ketchup Mayonesse
3. Skin Dry Oily Normal Sun
mpeshko@c3a4c9:~/mpeshko/pipex_bonus_22.06$ cat 333out.txt
mpeshko@c3a4c9:~/mpeshko/pipex_bonus_22.06$ sort -n < 333.txt | head -n 3 | head
-n 1 | wc -w > 333out.txt
mpeshko@c3a4c9:~/mpeshko/pipex_bonus_22.06$ cat 333out.txt
6
mpeshko@c3a4c9:~/mpeshko/pipex_bonus_22.06$
```

Code

```
int main(int argc, char **argv, char **envp)
{
    int cmd;
    int write_to;
    int read_from;

    if (argc >= 5)
```

```

{
    cmd = 2;
    write_to = open_file(argv[argc - 1], 1);
    read_from = open_file(argv[1], 2);
    dup2(read_from, STDIN_FILENO);
    close(read_from);
}
while (cmd < argc - 2)
{
    read_from = child_process(argv[cmd++], envp, read_from);
}
close(read_from);
dup2(write_to, STDOUT_FILENO);
close(write_to);
execute(argv[argc - 2], envp);
}

```

```

int child_process(char *argv, char **envp, int read_from)
{
    pid_t    pid;
    int      fd[2];

    if (pipe(fd) == -1)
        error();
    pid = fork();
    if (pid == -1)
        error_fork();
    if (pid == 0)
    {
        close(fd[0]);
        dup2(fd[1], STDOUT_FILENO);
        execute(argv, envp);
    }
    else

```

```
{  
    close(fd[1]);  
    dup2(read_from, STDIN_FILENO);  
    close(read_from);  
    waitpid(pid, NULL, 0);  
}  
return(fd[0]);  
}
```

333.txt

- 1. Forest.
- 2. Tea black green white
- 3. Sause Salsa Ketchup Mayonesse
- 4. Skin Dry Oily Normal Sun

333out.txt

6

```
✓ TERMINAL zsh + ▢ ▢ ..
● c3a4c9% ./pipex_bonus 333.txt "sort -n" "head -n 3" "head -n 1" "wc -w" 333out.txt
write_to: 3
write_to: 4
close read_from(4)

read_from: 4
child_process. new pipe
fd[0]: 4, fd[1]: 5
close (fd[1]: 5)
I am reading data from 4
child_process returns read_from: 4

read_from: 4
child_process. new pipe
fd[0]: 4, fd[1]: 5
close (fd[1]: 5)
I am reading data from 4
child_process returns read_from: 4

read_from: 4
child_process. new pipe
fd[0]: 4, fd[1]: 5
close (fd[1]: 5)
I am reading data from 4
child_process returns read_from: 4

close(4)
close(3)
○ c3a4c9% █
```