```
/********************* cat.c file ****************/
#include "ucode.c"

char *_argv[100];
int main(int argc, char *argv[ ])
{
    prints("$$$$$$$$$$$$$$$$Yuzhu's grep $$$$$$$$$$$$$$&&&&&&&&&&&$\n");
        char line[1024], dummy = 0; // a null char at the end of mybuf[]
        int n;
        //simple grep without handling: grep from stdin
        if(argc == 2)
        {
                while(gets(line))
                {
                        if(strstr(line,argv[1]))
                        {
                                prints(line);
                        }
                }
        }
        else
        {
                int fd = open(argv[2],O_RDONLY);//open for read
                while( n = readline(fd, line))
                {
                         if(strstr(line,argv[1]))
                         {
                                printf("%s",line);
                         }

                }
            prints("$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n");
                close(fd);//close the file descriptor
        }

}

int readline(int fd, char *buf)
{
        int i = 0;
        char temp[5] = "";
        strcpy(buf,"");//clear buf
        while(strcmp(temp,"\n") != 0){
                if(read(fd,temp,1) > 0)
                {
                        strcat(buf,temp);
                        i++;
                }
                else //if read(fd,temp,1) == 0
                        break;
        }
        return i;
}

int parseLine(char *line)
{
  char *cp = line;
  argc = 0;
  while (*cp != 0){
    while (*cp == ' ')  *cp++ = 0;
    if (*cp != 0)// skip over blanks // token start
      _argv[argc++] = cp; // pointed by argv[ ]
    while (*cp != ' ' && *cp != 0) // scan token chars
      cp++;
    if (*cp != 0)
      *cp = 0;
    else // end of token
      break; // end of line
    cp++;// continue scan
  } //end outer while
  _argv[argc] = 0; // argv[argc]=0
  return argc;//return the number of argc's
}
```