Citirea argumentelor din linia de comanda

In C, citirea argumentelor din linia de comanda se face asemanator cu modul de lucru din Bash.

```c
/**
* main function gets an array of arguments, and the total number of arguments
*/
int main(int argc, char* argv[])
{
    return 0;
}
```

Cerinta:

Implementati functionalitatea programului **echo** in **C**.

Lucru cu fisiere in C

Pentru a deschide un fisier, folosim functia **fopen** din header-ul **stdio.h**.

Prototipul functiei este :

```c
FILE *fopen(const char *path, const char *mode);
```

Mode	Description
r	Open text file for reading. The stream is positioned at the beginning of the file.
r+	Open for reading and writing. The stream is positioned at the beginning of the file.
w	Truncate file to zero length or create text file for writing. The stream is positioned at the beginning of the file.
w+	Open for reading and writing. The file is created if it does not exist, otherwise it is truncated. The stream is positioned at the beginning of the file.
a	Open for reading and writing. The file is created if it does not exist, otherwise it is truncated. The stream is positioned at the beginning of the file.
a+	Open for reading and appending (writing at end of file). The file is created if it does not exist. The initial file position for reading is at the beginning of the file, but output is always appended to the end of the file.

Citirea din fisier se face cu functia fread, prototipul functiei este:

```
/**
* ptr  - pointer to a block of memopry
* size - size in bytes of each element to be read
* count - number of elements to be read
* stream - pointer to FILE object
* return - total number of bytes read
*/
size_t fread ( void * ptr, size_t size, size_t count, FILE * stream );
```

Scrierea in fisier se face cu functia fwrite, prototipul functiei este:

```
/**
* ptr - Pointer to the array of elements to be written, converted to a const void*.
* size - Size in bytes of each element to be written.
* count - Number of elements, each one with a size of size bytes.
* stream - Pointer to a FILE object that specifies an output stream.
*/
size_t fwrite ( const void * ptr, size_t size, size_t count, FILE * stream );
```

Exemplu:

```
/* Trivial file copy program using standard I/O library */

#include <stdio.h>
#define BSIZE 16384

int main()
{
    FILE *fin, *fout; /* Input and output handles */
    char buf[BSIZE];
    int count;

    fin = fopen("foo", "r");
    fout = fopen("bar", "w");

    // check if files were opened
    if(fin == NULL || fout == null)
    {
        // if not, exit with error code 1
        return 1;
    }

    // while there are still bytes to be read, copy contents
    while ((count = fread(buf, 1, BSIZE, fin)) > 0)
        fwrite(buf, 1, count, fout);

    fclose(fin);
    fclose(fout);
    return 0;
}
```

Cerinta:

Extindeti aplicatia de mai sus pentru a

1. implementa functionalitatea aplicatiei cp (copy) folosind fread, fwrite
2. In programul vostru, daca se seteaza flag-ul **-a** sub forma:

   ```
   <program_name> -a src_file_name dest_file_name
   ```

   Atunci se va copia in *continuarea* fisierului, daca acesta exista.

   . In programul vostru, daca se seteaza flag-ul **-lower** sub forma:

   ```
   <program_name> -lower src_file_name dest_file_name
   ```

   Atunci se va scrie in fisiere destinatie numai cu litere mici, fara majuscule (hint: folositi tabelul ASCII)

Directoare

Pentru a deschide un director se foloseste functia **opendir**:

```
DIR *opendir(const char *name);
```

Pentru a citi datele dintr-un director se foloseste functia **readdir**:

```
struct dirent *readdir(DIR *dirp);
```

Structura **dirent** este definita ca:

```
struct dirent
{
    ino_t          d_ino;       /* inode number */
    off_t          d_off;       /* offset to the next dirent */
    unsigned short d_reclen;    /* length of this record */
    unsigned char  d_type;      /* type of file; not supported
                                   by all file system types */
    char           d_name[256]; /* filename */
};
```

Pentru a inchide un director folosim functia **closedir**.

```
int closedir(DIR *dirp);
```

Exemplu, listarea elementelor dintr-un director:

```
#include <dirent.h>

#include <stdio.h>
```

```c
int main(void) {
  DIR *d;
  struct dirent *dir;
  d = opendir(".");
  if (d) {
    while ((dir = readdir(d)) != NULL) {
      printf("%s\n", dir->d_name);
    }
    closedir(d);
  }
  return(0);
}
```

Cerinta:

Modificati codul dat astfel incat sa listati **recursiv** continutul.

Hints:

1. La listare, ignorati "." si ".." (this directory and parent directory)
2. Tipul elementului, (director, fisier, etc, este dat de membrul d_type)