**IME DATOTEKE KOT ARGUMENT V UKAZNI VRSTICI**

Ime datoteke lahko:

1. podamo v metodi open(), ko inicializiramo datotečni kazalec – datoteko ne moremo spremeniti razen, če spremenimo kodo;
2. preberemo kot niz – datoteko lahko spremenimo ob zahtevanem vpisu imena;
3. podamo kot argument v ukazni vrstici.

Klic programa lahko izvedemo v ukazni vrstici cmd npr. ***program.exe*** poženemo z ukazom ***program*** (ni potrebno pisati .exe).

V ukazni vrstici cmd lahko zapišemo tudi več besed, pri čem je prva beseda vedno ime .exe datoteke oz. programa. Kje se potem to prenaša v sam program oz. kje lahko vidimo, da vpliva na izvajanje programske kode? Glede na to, da se vsaka koda začne izvajati v glavni funkciji main(), je za pričakovati, da se bo ukazna vrstica nekako morala poslati kot argument v main():

int main(int argc, char \*argv[])

**argc** - število besed, ki smo jih vpisali v ukazni vrstici (šteje **vse** besede vključno z imenom programa)

**\*argv[]** – tabela kazalcev na besede v ukazni vrstici

Za ime programa je vedno rezerviran prvi niz oz. argv[0].

Primer 1:

#include <iostream>

#include <fstream>

#include <cstdio>

#include <string>

using namespace std;

void vpis(char ime[]); //pošljemo drugo besedo iz cmd v funkcijo

int main(int argc, char \*argv[])

{

ifstream data;

string s;

data.open(argv[1]); //druga beseda v ukazni vrstici

if(data.is\_open())

{

while(getline(data,s))

cout<<s<<endl;

data.close();

}

return 0;

}

void vpis(char ime[]){

ofstream data (ime,ios::app);

data<<"to je prvi stavek\n";

data<<"v novi vrstici\n";

data.close();

}

**KOMENTAR:**

**Prva beseda v ukazni vrstici je rezervirana za ime programa!!** Npr. če program shranimo kot zapisi.exe, ga kličemo na način:

**zapisi stevila.txt**

in v tem primeru je ime datoteke (stevila.txt) druga beseda.

Primer 2:

Napiši program, ki briše vse zvezdice (\*) iz ene datoteke v drugo. Program poženemo :

BRISI ime\_stare\_datoteke ime\_nove\_datoteke

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

#include <fstream>

#include <cstdio>

#include <string>

using namespace std;

int main(int argc, char \*argv[])

{

ofstream zapis;

ifstream branje;

char c;

if (argc!=3)

{

cout<<"Pravilen klic programa je: BRISI datoteka\_stara datoteka\_nova!";

//preverjamo ali je podano ustrezno število besed

exit(1);

}

branje.open(argv[1]); //druga beseda v ukazni vrstici

if (!branje.is\_open())

{

cout<<"Ime stare datoteke ni pravilno.\n";

exit(1);

}

zapis.open(argv[2]); //tretja beseda v ukazni vrstici

while (branje>>noskipws>>c)

{

if (c!='\*')

zapis<<c;

}

branje.close();

zapis.close();

remove(argv[1]);

rename (argv[2],argv[1]);

return 0;

}

**KOMENTAR:**

Zakaj preverjamo število besed podanih v ukazni vrstici? Primer, ki nas zanima ima lahko podano premalo besed – v tem primeru je program nepredvidljiv. Če je podano preveč besed, to sicer ne deluje na delovanje programa, se pa lahko zgodi da je klic podan kot npr:

BRISI ZVEZDICE IZ ime\_stare\_datoteke

Kaj bi se zgodilo? Ime stare datoteke bi bilo ZVEZDICE, ime nove pa IZ, kar se verjetno ne ujema s tistim kar smo želeli. Zato je dobro preveriti, če je podano TOČNO število besed, ki jih pričakujemo.