

## [[IZP Home]]

### Navigace

- [Hlavní stránka](#)
- [Poslední změny](#)
- [Nápověda](#)

### Stránka

### Nástroje

- [Odkazuje sem](#)
- [Nahrát soubor](#)
- [Seznam souborů](#)
- [Seznam stránek](#)
- [Historie stránky](#)

**Stránka**

[Zdroj](#)

[Sledovat](#)

# Projekt1

## Legenda:

**Červeně  
označený  
text** značí  
změnu  
zadání. Na  
změnu  
zadání bude  
upozorněno  
na fóru.  
**Modře  
označený  
text**  
upřesňuje  
zadání. Takto  
označené  
fráze

nebudou mít na hodnocení téměř žádný vliv.

## Obsah

### 1 Projekt 1 - Práce s textem

#### 1.1 Popis projektu

#### 1.2 Detailní specifikace

##### 1.2.1 Překlad a odevzdání zdrojového souboru

##### 1.2.2 Syntax spuštění

##### 1.2.3 Implementační detaily

###### 1.2.3.1 Postup úprav souboru

###### 1.2.3.2 Příkazy pro editaci textu

###### 1.2.3.3 Poznámky k implementaci

##### 1.2.4 Neočekávané chování

##### 1.2.5 Příklady vstupů a výstupů

#### 1.3 Hodnocení

## Projekt 1 - Práce s textem

### Popis projektu

Cílem projektu je na praktickém problému naučit studenty vytvářet jednoduché programy, dekomponovat problém na podproblémy a správně zapisovat zdrojové soubory. Praktickým problémem je implementace textového editoru. Editor nebude interaktivní, tj. nebude komunikovat s uživatelem, ale příkazy pro úpravu textu bude načítat ze vstupního souboru. Editor bude řádkový (úpravy bude provádět po jednotlivých řádcích). Editor bude proudový - editace textu bude provedena čtením upravovaného vstupu po řádcích a zápisu výsledného obsahu souboru po řádcích na standardní výstup.

### Detailní specifikace

Program implementujte ve zdrojovém souboru *proj1.c*. Vstupní binární data budou čtena ze standardního vstupu (stdin), výstup bude tisknut na standardní výstup (stdout). Chování programu lze upřesnit jeho argumenty (viz spuštění programu).

### Překlad a odevzdání zdrojového souboru

Odevzdání: Odevzdejte zdrojový soubor *proj1.c* prostřednictvím informačního systému.

Překlad: Program překládejte s následujícími argumenty

```
$ gcc -std=c99 -Wall -Wextra -Werror proj1.c -o proj1
```

## Syntax spuštění

Program se spouští v následující podobě: (./proj1 značí umístění a název programu):

```
./proj1 JMENO_SOUBORU_S_PRIKAZY
```

Pokud je program spuštěn bez argumentů nebo s více než jedním argumentem, vypíše svůj krátký popis (nápořvědu) a úspěšně skončí. V opačném případě provádí čtení a zpracování dat vstupu.

## Implementační detaily

### Postup úprav souboru

Podle příkazů pro editor předepsané v příkazovém souboru (zde soubor zadaný na příkazové řádce) budou postupně prováděny úpravy textu na standardním vstupu a výsledný obsah bude tisknut na standardní výstup.

Vstupní soubor bude načítán po celých řádcích. Obsah výstupního souboru bude tisknut po celých řádcích. Každý řádek ve výstupním souboru bude zakončen znakem konce řádku.

### Příkazy pro editaci textu

Příkazový soubor obsahuje příkazy pro editaci textu. Každý příkaz je na jednom řádku. Příkazy se zpracovávají postupně od prvního po poslední. Postupné zpracování příkazů může narušit příkaz *g*. Editační příkazy mohou být následující:

- *iCONTENT* (insert) vloží před aktuální řádek řádek s obsahem CONTENT.
- *bCONTENT* (before) vloží na začátek aktuálního řádku obsah CONTENT.
- *aCONTENT* (after/append) vloží na konec aktuálního řádku obsah CONTENT.
- *d* (delete) smaže aktuální řádek.
- *dN* smaže N řádků ( $N > 0$ ).
- *r* (remove EOL) na aktuálním řádku odstraní znak konce řádku.
- *s PATTERN REPLACEMENT* (substitute) na aktuálním řádku provede substituci (nahrazení) výskytu řetězce PATTERN za řetězec REPLACEMENT. PATTERN musí být neprázdný. Pokud aktuální řádek neobsahuje podřetězec PATTERN, příkaz nic nedělá.
- *s:PATTERN:REPLACEMENT* substituce řetězce, přičemž PATTERN a REPLACEMENT mohou být od sebe oddělené libovolným tisknutelným znakem, zde vyjádřeno znakem `:`. PATTERN v takovém případě nesmí obsahovat daný oddělovač.
- *S:PATTERN:REPLACEMENT* stejné jako *s:PATTERN:REPLACEMENT*, ale nahrazuje všechny výskyty PATTERN.
- *n* (next) posun aktuálního řádku o jeden níž, tj. tiskne aktuální řádek.

- $nN$  posun aktuálního řádku o  $N$  níž. ( $N > 0$ ).
- $q$  (quit) ukončí editaci, další řádky nezpracovává a ani netiskne.
- $gX$  (goto) přeskoč s aktuálním příkazem pro editaci na  $X$ -tý příkaz.  $X$  vyjadřuje číslo řádku (příkazu) v příkazovém souboru ( $1 \leq X \leq$  počtu řádků příkazového souboru). Pozor, tento příkaz může způsobit, že se i správně implementovaný editor zacyklí.

Nepovinné příkazy:

- $fPATTERN$  (find) posun aktuálního řádku směrem dolů na řádek obsahující řetězec  $PATTERN$ .
- $cX PATTERN$  (conditioned goto) přeskoč s aktuálním příkazem pro editaci na  $X$ -tý příkaz v případě, že aktuální řádek obsahuje řetězec  $PATTERN$ . Pokud aktuální řádek řetězec neobsahuje, pokračuj s následujícím příkazem.
- $e$  (append EOL) přidej na konec aktuálního řádku znak konce řádku.

## Poznámky k implementaci

- Program končí editaci textu, pokud zpracoval poslední příkaz (pokud ten nebyl příkaz skoku), při příkazu  $q$  nebo pokud po příkazech  $n$  a  $d$  již nejsou data na vstupu.
- Pokud byly zpracovány všechny příkazy a vstupní soubor ještě obsahuje nějaký text, vytiskne se zbytek původního souboru bez změn.
- Program musí podporovat řádky o délce minimálně 1000 znaků.
- Program musí podporovat alespoň 100 příkazů v daném souboru příkazů.

Je zakázané použít následující funkce:

- volání z rodiny `malloc` a `free` - práce s dynamickou pamětí není v tomto projektu zapotřebí,
- volání `atoi` - neověřuje správnost vstupních dat.

## Neočekávané chování

Na chyby za běhu programu reagujte obvyklým způsobem: Na neočekávaná vstupní data, formát vstupních dat nebo chyby při volání funkcí reagujte přerušением programu se stručným a výstižným chybovým hlášením na příslušný výstup a odpovídajícím návratovým kódem. Hlášení budou v kódování ASCII česky nebo anglicky.

## Příklady vstupů a výstupů

```
$ cat head.txt
n5
q

$ cat vstup
Toto
je
ne
ja
ky
sou
bor
s
vice
radky
```

```
$ ./proj1 head.txt <vstup
Toto
je
ne
ja
ky
```

```
$ cat tabular.txt
b|
a |
n
g1

$ cat vstup
Toto je
nejaky soubor

$ ./proj1 tabular.txt <vstup
| Toto je |
| nejaky soubor |
```

```
$ cat mkcsv.txt
a,
r
n
n
g1

$ cat vstup
Jmeno pacienta
rok narozeni
Antonin Jedlicka
1963
Alzbeta Jedla
1965

$ ./proj1 mkcsv.txt <vstup
Jmeno pacienta,rok narozeni
Antonin Jedlicka,1963
Alzbeta Jedla,1965
```

## Hodnocení

Na výsledném hodnocení mají hlavní vliv následující faktory:

- přeložitelnost zdrojového souboru,
- formát zdrojového souboru (členění, zarovnání, komentáře, vhodně zvolené identifikátory),
- dekompozice problému na podproblémy (vhodné funkce, vhodná délka funkcí a parametry funkcí),
- správná volba datových typů, případně tvorba nových typů,
- správná funkcionálnost programu a
- ošetření chybových stavů.

Na tomto projektu lze získat prémiové hodnocení. Získání prémiového hodnocení je podmíněné správnou implementací řádného zadání.

Možnosti pro získání prémiového hodnocení jsou následující:

- detekce zacyklení v sekvenci příkazů,
- implementace nepovinných příkazů.