

Podstawy Programowania
Semestr letni 2022/23
Materiały z laboratorium i zadania domowe

Przemysław Olbratowski

3 marca 2023

Slajdy z wykładu są dostępne w serwisie UBI. Informacje organizacyjne oraz formularz do uploadu prac domowych znajdują się na stronie info.wsisiz.edu.pl/~olbratow. Przy zadaniach domowych w nawiasach są podane terminy sprawdzeń.

8.2 Zadania domowe z działu Łącuchy (17, 24, 31 maja)

8.2.1 Compact: Usuwanie pustych linii

Napisz program `compact`, który przyjmuje jako argumenty wywołania nazwy dwóch plików tekstowych i przepisuje zawartość pierwszego pliku do drugiego opuszczając linie puste oraz zawierające tylko znaki białe. Program załącza tylko pliki nagłówkowe `fstream` i `string`.

Przykładowy plik wejściowy `input.txt`

```
lorem ipsum

dolor sit

amet
consectetur adipiscing elit
```

Przykładowe wywołanie

```
Linux: ./compact input.txt output.txt
Windows: compact.exe input.txt output.txt
```

Przykładowy plik wyjściowy `output.txt`

```
lorem ipsum
dolor sit
amet
consectetur adipiscing elit
```

8.2.2 Ebook: Elektroniczna książka

Kolejne strony elektronicznej książki są zapisane w kolejnych liniach pliku tekstowego. Napisz program `ebook`, który przyjmuje nazwę tego pliku jako argument wywołania. Po uruchomieniu wypisuje na standardowe wyjście pierwszą stronę książki i wczytuje ze standardowego wejścia polecenie. Jeżeli wydano polecenie `next` lub `previous`, drukuje odpowiednio następną lub poprzednią stronę. Jeżeli podano numer strony, drukuje tę stronę. Po wyświetleniu nowej strony program ponownie wczytuje polecenie i tak dalej. Wykonanie kończy się po wydaniu polecenia `exit` lub napotkaniu końca pliku. Program załącza tylko pliki nagłówkowe `fstream`, `iostream`, `string` i `vector`.

Przykładowy plik wejściowy `input.txt`

```
text in the first page
text in the second page
text in the third page
```

Przykładowe wykonanie

```
Linux: ./ebook input.txt
Windows: ebook.exe input.txt
Out: text in the first page
In: next
Out: text in the second page
In: 3
Out: text in the third page
In: exit
```

8.2.3 Grep: Wyszukiwanie linii

Napisz program `grep`, który przyjmuje jako argumenty wywołania łańcuch tekstowy oraz nazwę pliku tekstowego i wypisuje na standardowe wyjście wszystkie linie tego pliku zawierające podany łańcuch. Program załącza tylko pliki nagłówkowe `fstream`, `iostream` i `string`.

Przykładowy plik wejściowy `input.txt`

```
ala ma kota
iza ma psa
hela ma chomika
basia ma rybki
```

Przykładowe wykonanie

```
Linux: ./grep "la ma" input.txt
Windows: grep.exe "la ma" input.txt
Out: ala ma kota
Out: hela ma chomika
```

8.2.4 History: Test z historii

Plik tekstowy `history.txt` zawiera listę wydarzeń historycznych zapisaną jak poniżej. Napisz program `history`, który wypisuje na standardowe wyjście nazwy kolejnych wydarzeń z tego pliku i wczytuje ze standardowego wejścia ich lata. Po każdej odpowiedzi program wypisuje `true` jeśli jest ona poprawna albo `false` w przeciwnym razie. Na końcu program wypisuje liczbę poprawnych odpowiedzi. Program załącza tylko pliki nagłówkowe `fstream`, `iostream` i `string`.

Przykładowy plik `history.txt`

```
-753 foundation of rome
1972 creation of the c language
```

Przykładowe wykonanie

```
Out: foundation of rome In: -753
Out: true
Out: creation of the c language In: 2018
Out: false
Out: 1
```

8.2.5 Initials: Inicjały - indywidualnie

Napisz funkcję `initials`, która przyjmuje stałą referencję łańcucha imion oraz nazwisk pewnej osoby i zwraca łańcuch jej inicjałów, czyli pierwszych liter kolejnych członów. Funkcja powinna być przystosowana do użycia w przykładowym programie poniżej. Funkcja korzysta tylko z plików nagłówkowych `cctype` i `string`.

Przykładowy program

```
int main() {
    std::cout << initials("John Fitzgerald Kennedy") << std::endl;
    std::cout << initials(std::string("andy warhol")) << std::endl; }
```

Wykonanie

```
Out: JFK
Out: aw
```

8.2.6 Labyrinth: Znajdowanie drogi w labiryncie

Pewien plik tekstowy zawiera planszę labiryntu zapisaną jak poniżej. Cały labirynt jest otoczony murem a chodzić po nim można tylko poziomo lub pionowo. Napisz program `labyrinth`, który przyjmuje jako argument wywołania nazwę pliku z planszą i wypisuje na standardowe wyjście tę samą planszę z zaznaczoną gwiazdkami drogą od pola 1 do 2. Program załącza tylko pliki nagłówkowe `fstream`, `iostream`, `string` i `vector`.

Przykładowy plik wejściowy `input.txt`

```
@@@@@@@@@
@         @
@ @ @@@@@@
@ @       @ @
@2@@@@@ @ @
@@@   @   @
@   @@@ @@@
@@     1@
@@@@@@@@
```

Przykładowe wykonanie

```
Linux: ./labyrinth input.txt
Windows: labyrinth.exe input.txt
Out: @@@@@@@@@
Out: @***   @
Out: @*@@@*@@@@
Out: @*@@@@* @ @
Out: @2@@@@* @ @
Out: @@@   @* @
Out: @   @@@*@@@
Out: @@     1@
Out:   @@@@@@
```

Wskazówka Planszę labiryntu przechowuj w strukturze danych typu `std::vector<std::string>`.

8.2.7 Orbilius: Test z języka obcego

Pewien plik tekstowy zawiera w pierwszej kolumnie słowa polskie, zaś w drugiej ich odpowiedniki w języku obcym, jak w przykładzie poniżej. Napisz program `orbilius`, który przyjmuje nazwę tego pliku jako argument wywołania i przeprowadza na tej podstawie test z języka obcego. Program wypisuje na standardowe wyjście losowo wybrane z pliku słowo polskie i wczytuje ze standardowego wejścia odpowiednik obcy. Pytanie to powtarza do uzyskania poprawnej odpowiedzi. Program zadaje dziesięć pytań, które mogą się powtarzać. Program załącza tylko pliki nagłówkowe `cstdlib`, `ctime`, `fstream`, `iostream`, `string` i `vector`.

Początek przykładowego pliku `english.txt`

```
woda water
powietrze air
olówek pen
```

Przykładowe wykonanie

```
Linux: ./orbilius english.txt
Windows: orbilius.exe english.txt
Out: dom           In: house
Out: ołówek        In: pencil
```

| | |
|----------------|---------------|
| Out: woda | In: air |
| Out: woda | In: water |
| Out: powietrze | In: air |
| Out: stol | In: table |
| Out: krzeslo | In: chair |
| Out: samochod | In: car |
| Out: motyl | In: butterfly |
| Out: kwiat | In: flower |
| Out: okno | In: window |

8.2.8 Palindrom: Wykrywanie palindromów

Palindrom to tekst, który czytany po literze od tyłu jest taki sam, jak czytany od przodu. Nie liczy się przy tym wielkość liter ani żadne znaki poza literami. Napisz program `palindrom`, który wczytuje ze standardowego wejścia jedną linię tekstu i wypisuje na standardowe wyjście `true`, jeśli jest on palindromem albo `false` w przeciwnym razie. Program załącza tylko pliki nagłówkowe `cctype`, `iostream` i `string`.

Przykładowe wykonanie

```
In: Ile Roman ladny dyndal na moreli?
Out: true
```

8.2.9 Sort: Sortowanie słów - grupowo

Napisz program `sort`, który czyta ze standardowego wejścia słowa do napotkania końca pliku i wypisuje je na standardowe wyjście w kolejności alfabetycznej. Program załącza tylko pliki nagłówkowe `iostream`, `string` i `vector`.

Przykładowe wykonanie

```
In: a long time ago in a galaxy far far away
Out: a a ago away far far galaxy in long time
```

8.2.10 Typeset: Podział tekstu na linie

Napisz program `typeset`, który przyjmuje jako argumenty wywołania dodatnią liczbę całkowitą n oraz nazwy dwóch plików tekstowych i przepisuje słowa z pierwszego pliku do drugiego oddzielając je pojedynczymi spacjami i dzieląc tekst na linie bez łamania słów. W każdej linii umieszcza największą możliwą liczbę słów tak, aby długość linii nie przekroczyła n znaków, nie licząc znaku końca linii. Program załącza tylko pliki nagłówkowe `fstream` i `string`.

Przykładowy plik wejściowy `input.txt`

```
lorem ipsum dolor sit amet consectetur adipiscing elit
sed eius mod tempor incididunt ut labore et dolore magna aliqua ut enim ad minim veniam
quis nostru exercitation ullam co laborios nisi ut aliquid ex ea commodi consequat
```

Przykładowe wywołanie

```
Linux: ./typeset 40 input.txt output.txt
Windows: typeset.exe 40 input.txt output.txt
```

Przykładowy plik wyjściowy output.txt

```
lorem ipsum dolor sit amet consectetur  
adipisci elit sed eius mod tempor  
incidunt ut labore et dolore magna  
aliqua ut enim ad minim veniam quis  
nostru exercitation ullam co laborios  
nisi ut aliquid ex ea commodi consequat
```