



## Program 1. (język C)

Napisać program, który dla każdej wczytanej linii danych (za linię danych przyjmujemy dowolny ciąg danych złożony ze znaków drukowanych tzn. liter, cyfr i znaków specjalnych) wypisze jej odpowiednik tworzony według niżej określonych zasad:

- każdy wyraz będzie napisany wielką literą
- wszystkie litery wewnątrz pojedynczego wyrazu (prócz pierwszej) będą zamienione na małe
- jeżeli wyrazu nie rozpoczyna litera, należy pominąć pierwszą zasadę i zastosować jedynie drugą

Przykład:

STDIN	STDOUT
ABCDEF AbCdEf A#13BC\$# 123AB4DE#\$\$% a12345 123456 123456C 0xFF 0xCZFD 0b0011	Abcdef Abcdef A#13bc\$# 123ab4de#\$\$% A12345 123456 123456C 0xff 0xczfd 0b011

**Ograniczenie!** Wczytywanie danych należy zrealizować wykorzystując tylko i wyłącznie funkcję `getchar()` z `stdio.h`.

## Program 2. (język C)

Napisać program, który dla każdego polskiego znaku diakrytycznego wypisze jego wielką lub małą postać – przeciwną do wczytanej (należy założyć, że podawane znaki będą kodowane za pomocą UTF-8).

**Ograniczenie!** Wczytywanie danych należy zrealizować wykorzystując tylko i wyłącznie funkcję `getchar()` z `stdio.h`. Można wykorzystać tylko i wyłącznie jeden plik nagłówkowy.

STDIN	STDOUT
ąĄęĘźĆż	ĄĄĘęŻĆŻ



### Program 3. (język C)

Napisać program, który dla każdego wczytanego bajtu danych wypisze jego reprezentację binarną. Plik wyjściowy musi zostać zapisany w 4 kolumnach oddzielonych spacjami (w każdej kolumnie reprezentacja pojedynczego bajtu).

STDIN	STDOUT
Abcdefab	01000001 01100010 01100011 01100100 01100101 01100110 01100001 01100010 01100011 01100100 01100101 01100110 01100001 01100010

**Ograniczenie!** Wczytywanie danych należy zrealizować wykorzystując tylko i wyłącznie funkcję `getchar()` z `stdio.h`. Można wykorzystać tylko i wyłącznie jeden plik nagłówkowy. Zabronione jest korzystanie z funkcji `printf()`, `sprintf()`, `fprintf()` i pochodnych.

### Program 4. (język C)

Napisać program, który dla każdego wczytanego bajtu danych wypisze jego reprezentację szesnastkową. Plik wyjściowy musi zostać zapisany w 12 kolumnach oddzielanych spacjami (w każdej kolumnie reprezentacja pojedynczego bajtu).

STDIN	STDOUT
Abcdefab	41 62 63 64 65 66 61 62 63 64 65 66 61 62 0A



## Program 5. (język C)

Napisać program, który dla wczytanej liczby/znaku wypisze jej/jego reprezentacje w każdym z trzech systemów liczbowych, kolejno: binarnym, szesnastkowym, dziesiętnym. Liczby na wejściu mogą być podawane w dowolnym z trzech wymienionych wcześniej systemów. Przyjmujemy, że liczby binarne rozpoczynają się od „0b”, szesnastkowe od „0x”, każde inne uznajemy za dziesiętne. Jeżeli na wejściu pojawi się znak ASCII (z wyjątkiem tych opisanych wcześniej - „0b” oraz „0x”), należy potraktować go jako liczbę (patrz przykład niżej). Plik wyjściowy musi zostać zapisany w trzech kolumnach, każda musi zawierać reprezentacje wczytanej liczby w jednym z podanych wcześniej systemów liczbowych. Należy zachować podaną wcześniej kolejność. Przy wypisywaniu należy użyć najkrótszej postaci (np. pomijając zbędne zera).

STDIN	STDOUT
1	0b1 0x1 1
0x01	0b1 0x1 1
0xFF	0b11111111 0xFF 255
0b1	0b1 0x1 1
0b00000001	0b1 0x1 1
0xff	0b11111111 0xFF 255
abc	0b01100001 0x61 97 0b01100010 0x62 98 0b01100011 0x63 99