

## Zadanie D: Parsowanie parametrów (20 punktów)

Napisz skrypt analizujący podane do niego argumenty i wypisujący na wyjściu:

- jeśli to wymagane (według parametrów opisanych poniżej) – całe standardowe wejście
- w przedostatniej linii – liczbę wystąpień opcji pomocy
- w ostatniej linii – ciąg, oddzielonych spacją, nazw plików podanych według parametrów opisanych poniżej

Obsługiwane parametry (wszystkie mogą wystąpić wielokrotnie) to:

- `-h` lub `--help` – opcja pomocy (zliczane dla przedostatniej linii)
- `-f` parametr plikowy. Po wystąpieniu `-f` kolejny argument interpretowany jest jako nazwa pliku
- `--file` argument plikowy. Bezpośrednio po `--file` musi wystąpić znak `=` i nazwa pliku (bez rozdzielających spacji)
- `-s` lub `--stdin` – flaga informująca, że standardowe wejście ma zostać wypisane. Brak takiej flagi oznacza, że standardowe wejście ma być zignorowane.
- Każdy inny parametr powinien zostać zignorowany

Jeśli parametr `-f` jest ostatni, albo po `--file` nie ma odpowiednio podanego parametru plikowego, to skrypt nic nie powinien wypisać, ale zakończyć się z kodem błędu 10.

Do obsługi bliżej nieokreślonej liczby parametrów przydatna może być komenda `shift`

W konstrukcji `case` przydatny może być fakt, że wzorce dopasowywane są jako globy. Ponadto można użyć `|` jako alternatywy dla całego wzorca, gdy kilka różnych wzorców ma mieć to samo działanie.

Uwaga:

- wysyłany plik powinien mieć rozszerzenie `.sh`
- na końcu skryptu w komentarzu ma się znaleźć imię i nazwisko autora rozwiązania
- plik powinien mieć uniksowe kodowanie końców linii (`\n`, a nie `\r\n`)

### Przykład 1:

```
./skrypt.sh -f plik1 --help -h --file=plik2 -h
./skrypt.sh --file
echo $?
echo "test" | ./skrypt.sh --stdin --help
```

Wynik ciągu poleceń:

```
3
plik1 plik2
10
test
1
```