

Wstęp

W pierwszym etapie tworzymy prosty program do obsługi paczkomatu. Program posiada możliwość stworzenia paczkomatu o dowolnym rozmiarze oraz obsługiwać procedurę dodawania paczek

Wymagania

- W aplikacji obsługujemy tylko jeden paczkomat
- Paczkomat jest prostokątem o rozmiarze $M \times N$
- *UWAGA!* Dotyczy wszystkich kolejnych etapów! Po stworzeniu paczkomatu, każda komenda wprowadzona przez użytkownika powinna wypisywać stan paczkomatu w takiej formacie w jakim przekazywany jest paczkomat na początku działania programu
- Paczkomat jest tworzony poprzez podanie jego wejściowej konfiguracji (rozmiaru oraz wstępnej zajętości) w formacie:

```
|S:X S:0|  
|S:0 S:0|
```

gdzie podany przykład jest paczkomatem o rozmiarze 2×2 z 1 skrytką zajętą i 3 pustymi (wolnymi)
 $S:X$ oznacza skrytkę zajętą
 $S:0$ oznacza skrytkę pustą

- Stworzenie paczkomatu odbywa się poprzez wielokrotne wczytanie linii paczkomatu, zakończenie tworzenia paczkomatu odbywa się poprzez wprowadzenie pustej linii
- Po stworzeniu paczkomatu można podać dowolną liczbę razy komendę dodającą paczki do paczkomatu w formie:

```
B;B
```

gdzie poszczególne paczki oddzielone są ;

- zakładamy że komendy dodające które przyjdą nigdy nie przepełnią paczkomatu i są możliwe do wykonania
- paczki powinny być wkładane do paczkomatu od lewej górnej skrytki a następnie w prawo, w przypadku zapełnienia całego rzędu rozważamy kolejny rząd od lewej strony
- jedna skrytka mieści tylko jedną paczkę

dla przykładu dla paczkomatu o rozmiarze 4×2 z następującą konfiguracją wstępną (3 skrytki zajęte, 5 wolnych)

```
|S:X S:0 S:X S:0|  
|S:0 S:0 S:0 S:X|
```

po wykonaniu operacji dodania 3 paczek

```
B;B;B
```

stan paczkomatu powinien prezentować się następująco

```
|S:X S:X S:X S:X|  
|S:X S:0 S:0 S:X|
```

po wykonaniu kolejnej komendy dodania

```
B
```

stan paczkomatu powinien prezentować się następująco

```
|S:X S:X S:X S:X|  
|S:X S:X S:0 S:X|
```