

BSP, czyli Bajtański System Plików

Bajtocja postanowiła skorzystać z dobrodziejstw Wolnego Oprogramowania. Ponieważ jednak wymagania tamtejszych informatyków są bardzo wysokie istniejący system Bajnux został zmodyfikowany. Jednym ze zmienionych elementów był system plików - stworzono całkiem nowy i funkcjonalny system plików BFS.

Niestety okazało się, że na bardzo obciążonych komputerach system plików ulega dość mocnej fragmentacji. Twoim zadaniem jest napisanie programu, który przetestuje jak wiele pracy będzie potrzebnej aby zdefragmentować system plików na każdym z dysków przekazanych ci do testowania.

Ponieważ operacją najbardziej obciążającą dysk (i komputer) jest zapis, Twój program ma zliczać wyłącznie liczbę operacji zapisu na dysk. Ponieważ musisz zapewnić prawidłowe działanie systemu, jeśli podczas defragmentacji wystąpi awaria nie możesz w procesie defragmentacji używać pamięci - wszystkie dane muszą cały czas znajdować się na dysku. Jedyną dostępną dla ciebie operacją jest przepisanie zawartości bloku o pozycji A do bloku o pozycji B. Nie musisz czyścić nieużywanych bloków - system plików będzie je widział jako nieużywane i sam nadpisze zawartość w razie potrzeby. **Dokonując defragmentacji, należy dążyć do tego, aby liczba operacji zapisu na dysk była minimalna. Po procesie defragmentacji pliki mają zajmować spójny, początkowy obszar dysku.**

Dane wejściowe:

W pierwszej linii podana zostanie liczba zestawów danych S ($1 \leq S \leq 100$). Każdy zestaw danych opisuje jeden dysk i składa się z następujących elementów:

- W pierwszej linii zapisany jest rozmiar dysku D (jednostka: bloki); $1 \leq D \leq 100000$.
Bloki na dysku ponumerowane są od 0 do $D-1$.
- W drugiej linii zapisana jest liczba aktualnie używanych bloków U ; $0 \leq U < D$
- Kolejne U linii zajmują opisy użytych bloków - każdy z nich to trzy liczby oddzielone spacjami: $P F N$
 - P - pozycja bloku na dysku; $0 \leq P < D$
 - F - numer pliku; $0 \leq F < 1000$
 - N - numer bloku w pliku; $0 \leq N < U$

Każdy plik na dysku posiada unikalny numer.

Bloki w pliku mają kolejne numery - tzn. jeśli w danych pojawi się blok o numerze bloku w pliku (plik o numerze K) równym 3 to na pewno są w tym zestawie danych również bloki (o numerach bloku w pliku) 0, 1, 2 dla tego pliku K .

Po defragmentacji dysku pliki muszą być ułożone w porządku rosnącym (w oparciu o numer pliku). Bloki w obrębie pliku także muszą być ułożone w porządku rosnącym (w oparciu o numer bloku w pliku).

Dane wyjściowe:

Dla każdego zestawu danych należy wypisać następujące wyniki:

- Linie zawierającą numer zestawu danych (zestawy są numerowane od 1).
- N linii opisujących pliki w systemie plików, każda o postaci "m a b" gdzie "m" to numer pliku, "a" to pozycja w systemie plików pierwszego bloku zawierającego dane pliku, "b" to pozycja ostatniego bloku zawierającego dane pliku.
- Linie zawierającą liczbę operacji zapisu koniecznych do przeprowadzenia defragmentacji.

Przykładowe dane wejściowe:

```
2
6
4
1 0 2
2 0 1
3 0 3
4 0 0
5
4
0 0 0
1 1 0
2 1 1
3 2 0
```

Przykładowe dane wyjściowe:

```
1
0 0 3
4
2
0 0 0
1 1 2
2 3 3
0
```