

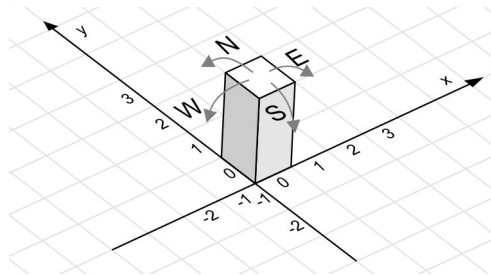
# Problem B

## Wędrujący klocek

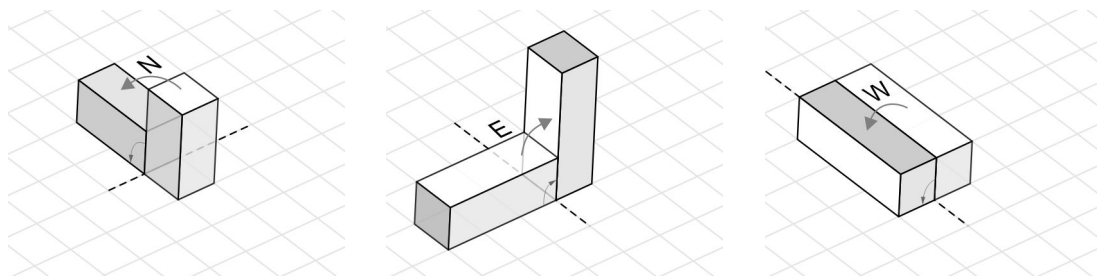
---

### IV OTWARTE MISTRZOSTWA OPOLA W PROGRAMOWANIU ZESPOŁOWYM

Płaszczyzna podzielona jest jak tablica dwuwymiarowa na pola określone parami liczb całkowitych ( $x$  – określa kolumnę,  $y$  – rząd). Na płaszczyźnie tej na polu o współrzędnych  $(0,0)$  znajduje się w chwili początkowej klocek o wysokości  $z$  zakresu od 1 do 10. Wymiary klocka to  $1 \times 1 \times k$ , gdzie  $k = 1 \dots 10$ . Następnie w serii ruchów klocek jest przetaczany na inne pozycje na planszy. Jeden ruch polega na obróceniu klocka o kąt  $90^\circ$  względem jednej z czterech krawędzi ściany, na której klocek w danej chwili stoi na planszy. W wyniku ruchu klocek zmienia swoje położenie na planszy tak, że leży obrócony inną ścianą do dołu. Cztery możliwe kierunki przetoczenia klocka oznaczane są literami N, S, W, E. Na poniższym rysunku przedstawiony jest klocek o wysokości 2, znajdujący się w położeniu początkowym. Strzałkami oznaczono cztery ruchy możliwe do wykonania przez klocek.



W kolejnych ruchach klocek przetacza się na kolejne pozycje planszy tak, że kolejne jego ściany stykają się z podłożem. Na poniższym rysunku przedstawiono przykładowe ruchy klocka.



### Zadanie

Zadanie polega na opracowaniu programu podającego na jakich polach planszy znajdzie się klocek po wykonaniu ostatniego ruchu.

Na niektórych polach klocek w czasie swej trasy może pojawić się wielokrotnie. Dlatego do każdego pola planszy, które odwiedził klocek można przypisać liczbę określającą ile razy znalazł się on na tym polu. Program powinien podać także największą z tych liczb, czyli wartość określającą ile najwięcej razy klocek znajdzie się na jakimś polu.

### Wejście

Program będzie sprawdzany przy pomocy zestawu  $N$  testów ( $N = 1 \dots 100$ ). Pierwszy wiersz standardowego wejścia zawiera liczbę testów. Od drugiego wiersza umieszczono dane wej-

ściowe dla kolejnych testów. Dane wejściowe dla jednego testu umieszczone są w dwóch wierszach. W pierwszym wierszu znajduje się liczba z zakresu od 1 do 10 określająca wysokość klocka w pozycji początkowej. W drugim wierszu znajduje się łańcuch tekstowy o długości od jednego do 100 znaków. Znakami tymi są litery N, S, W, E określające cztery możliwe kierunki przetaczania klocka. „N” oznacza przetoczenie klocka w kierunku dodatniego zwrotu osi  $y$ , „S” – w kierunku ujemnego zwrotu osi  $y$ . „E” oznacza przetoczenie klocka w kierunku dodatniego zwrotu osi  $x$ , a „W” w kierunku przeciwnym.

## Wyjście

Na standardowym wyjściu program dla każdego testu powinien wyprowadzić w kolejnych trzech wierszach informację o położeniu końcowym klocka oraz informację o tym ile najwięcej razy klocek znajdował się na jednym z pól trasy. W pierwszym wierszu wyniku dla danego testu powinny zostać umieszczone oddzielone spacjami współrzędne  $x$  kolejnych pól, na których klocek znajduje się w położeniu końcowym, wypisane w kolejności od najmniejszej do największej. W drugim wierszu – uszeregowane od najmniejszej do największej, także oddzielone pojedynczymi spacjami, współrzędne  $y$  klocka w położeniu końcowym. W trzecim wierszu powinna być umieszczona liczba naturalna będąca wartością maksymalną spośród wartości określających ile razy klocek był na kolejnych polach swojej trasy.

## Przykład

Dla 3 testów:

Test 1:

W 14 ruchach przetaczany jest klocek o wysokości 2. Na żadnym polu nie znajduje się więcej niż jeden raz. Po wykonaniu wszystkich ruchów klocek znajdzie się na polach (3,-3), (3,-2).

Test 2:

W 17 ruchach przetaczany jest klocek o wysokości 5. W czasie swojej trasy najwięcej, 8 razy znajdował się na tym samym polu. Po wykonaniu wszystkich ruchów klocek znajdzie się na polu (-5,0).

Test 3:

W 26 ruchach przetaczany jest klocek o wysokości 3. Po wykonaniu wszystkich ruchów klocek znajdzie się na polach (7,5), (8,5), (9,5). Na tym samym polu znajdował się maksymalnie 3 razy.

Dane wejściowe są podane następująco:

```
3
2
ENEESSESSESWWWN
5
WNSEWNSEWNSEWSEN
3
NNEESESWWNWWWWSSEEEENNE
```

Program powinien zwrócić w wyniku:

```
3 3
-3 -2
1
-5
0
8
7 8 9
5 5 5
3
```