

Rešitev druge domače naloge (Tipkovnica)

Ogrodje programa

Razdaljo, ki jo prepotuje Tanjin prst, izračunamo kot vsoto manhattanskih razdalj med zaporednimi tipkami. V ta namen se sprehodimo po zaporedju pritisnjenih tipk in v vsakem obhodu zanke (razen v prvem) izračunamo manhattansko razdaljo med prejšnjo in trenutno tipko. Razdalje med seboj seštejemo in dobimo iskani rezultat.

```
import java.util.Scanner;

public class Tipkovnica {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int oblika = sc.nextInt();           // oblika tipkovnice
        int d = sc.nextInt();                // število d v navodilih
        int dolzinaZaporedja = sc.nextInt(); // število pritisnjenih tipk

        int vrPrej = -1; // koordinati prejšnje tipke
        int stPrej = -1;
        int vrZdaj = -1; // koordinati trenutne tipke
        int stZdaj = -1;
        int pot = 0;      // celotna prepotovana razdalja

        for (int i = 1; i <= dolzinaZaporedja; i++) {
            int tipka = sc.nextInt();

            // izračunaj koordinati i-te pritisnjene tipke in ju shrani v vrZdaj in stZdaj
            switch (oblika) { // izračun je odvisen od oblike tipkovnice
                case 1:
                    ...

                case 2:
                    ...

                case 3:
                    ...

                case 4:
                    ...
            }

            if (i > 1) {
                pot += manhattanskaRazdalja(vrPrej, stPrej, vrZdaj, stZdaj);
            }

            // trenutna tipka bo v naslednjem obhodu zanke postala prejšnja
            vrPrej = vrZdaj;
            stPrej = stZdaj;
        }

        System.out.println(pot);
    }

    private static int manhattanskaRazdalja(int vr1, int st1, int vr2, int st2) {
```

```

        return (Math.abs(vr2 - vr1) + Math.abs(st2 - st1));
    }
}

```

Določitev koordinat posameznih tipk

Za določitev manhattanske razdalje med zaporednimi tipkami moramo izračunati koordinate posameznih tipk za podano obliko tipkovnice. Dogovorimo se, da ima zgornja vrstica tipkovnice koordinato 0, vrstica pod njo koordinato 1 itd., pri stolpcih pa ima skrajno levi koordinato 0, stolpec desno od njega koordinato 1 itd.

Ravnoprstnica

Najlažjo nalogo imamo pri ravnoprstnici: tipka t nahaja v vrstici s koordinato 0 in stolpcu s koordinato t .

```

public class Tipkovnica {

    public static void main(String[] args) {
        ...
        for (int i = 1; i <= dolzinaZaporedja; i++) {
            ...
            switch (oblika) {
                case 1:
                    vrZdaj = 0;
                    stZdaj = tipka;
                    break;
                ...
            }
            ...
        }
        ...
    }
}

```

Kvadratnica

Pri kvadratnici s stranico dolžine d se tipka t nahaja v vrstici s koordinato $\lfloor t/d \rfloor$ in stolpcu s koordinato $t \bmod d$:

```

public class Tipkovnica {

    public static void main(String[] args) {
        ...
        for (int i = 1; i <= dolzinaZaporedja; i++) {
            ...
            switch (oblika) {
                ...
                case 2:
                    vrZdaj = tipka / d;
                    stZdaj = tipka % d;
                    break;
            }
        }
    }
}

```

```

        ...
    }
    ...
}
...
}
}

```

Piramidnica

Pri piramidnici imamo nekoliko več dela. Vrstico, v kateri se nahaja tipka s številko `tipka`, lahko poiščemo s pomočjo zanke, ki se spreha po vrsticah piramidnice. V zanki vzdržujemo sledeče spremenljivke:

- `vr`: koordinata y tekoče vrstice (0 za prvo vrstico, 1 za drugo itd.);
- `stPrva`: koordinata x skrajno leve tipke v tekoči vrstici (za skrajno levo tipko v prvi vrstici velja $x = d - 1$, za skrajno levo tipko v zadnji vrstici pa $x = 0$);
- `prvaVVrstici`: številka skrajno leve tipke v tekoči vrstici;
- `stTipkVVrstici`: število tipk v tekoči vrstici.

Sledeča tabela prikazuje vrednosti spremenljivk v posameznih vrsticah piramidnice višine 6 (gl. navodila, str. 1 spodaj):

vr	stPrva	prvaVVrstici	stTipkVVrstici
0	5	0	1
1	4	1	3
2	3	4	5
3	2	9	7
4	1	16	9
5	0	25	11

Zanko, ki teče po vrsticah piramidnice, zaključimo, ko prispemo do vrstice, za katero velja `prvaVVrstici <= tipka` in `tipka < prvaVVrstici + stTipkVVrstici`. Koordinato x stolpca, ki vsebuje iskano tipko, določimo tako, da koordinati skrajno leve tipke v trenutni vrstici (`stPrva`) prištejemo razliko med številko iskane tipke in številko skrajno leve tipke v tekočem stolpcu.

```

public class Tipkovnica {

    public static void main(String[] args) {
        ...
        for (int i = 1; i <= dolzinaZaporedja; i++) {
            ...
            switch (oblika) {
                ...
                case 3: {
                    int prvaVVrstici = 0;
                    int stTipkVVrstici = 1;
                    int vr = 0;
                    int stPrva = d - 1;

```

```

        while (tipka >= prvaVVrstici + stTipkVVrstici) {
            prvaVVrstici += stTipkVVrstici;
            stTipkVVrstici += 2;
            vr++;
            stPrva--;
        }
        vrZdaj = vr;
        stZdaj = stPrva + tipka - prvaVVrstici;
        break;
    }
    ...
}
...
}
...
}
}
}

```

Spiralnica

Pri spiralnici se tipko 0 splača obravnavati posebej (njeni koordinati sta $x = y = \lfloor d/2 \rfloor$), za ostale tipke pa uberemo podoben pristop kot pri piramidnici, le da v zanki namesto po vrsticah potujemo po koncentričnih kvadratih od najbolj notranjega do najbolj zunanjega. V zanki vzdržujemo sledeče spremenljivke:

- **vrPrva**: koordinata y zgornje leve tipke v tekočem koncentričnem kvadratu;
- **stPrva**: koordinata x zgornje leve tipke v tekočem kvadratu;
- **prvaVKvadratu**: številka zgornje leve tipke v tekočem kvadratu;
- **stTipkVKvadratu**: število tipk v tekočem kvadratu;
- **stranicaKvadrata**: dolžina stranice tekočega kvadrata (minus 1).

Prve štiri spremenljivke so namenjene določitvi koncentričnega kvadrata, ki vsebuje iskano tipko, spremenljivka **stranicaKvadrata** pa nam bo pomagala določiti, na kateri stranici koncentričnega kvadrata se nahaja iskana tipka. Pri spiralnici z $d = 7$ so vrednosti spremenljivk v posameznih koncentričnih kvadratih od najbolj notranjega do najbolj zunanjega takšne (gl. navodila, str. 2 zgoraj):

vrPrva	stPrva	prvaVKvadratu	stTipkVKvadratu	stranicaKvadrata
2	2	1	8	2
1	1	9	16	4
0	0	25	24	6

V zanki, ki teče po koncentričnih kvadratih spiralnice, se v vsakem obhodu število tipk poveča za 8, dolžina stranice pa za 2. Zanko zaključimo, ko prispemo do koncentričnega kvadrata, za katerega velja $\text{prvaVKvadratu} \leq \text{tipka}$ in $\text{tipka} < \text{prvaVKvadratu} + \text{stTipkVKvadratu}$.

Ko poiščemo koncentrični kvadrat, moramo določiti še položaj iskane tipke znotraj kvadrata. Položaj je odvisen od razlike med številko iskane tipke in številko zgornje leve tipke v kvadratu. Če je razlika (spremenljivka **odmik** v programu) manjša od vrednosti **stranicaKvadrata**, se tipka nahaja na zgornji vrstici kvadrata. Če je razlika v intervalu

[stranicaKvadrata, 2 * stranicaKvadrata - 1], se tipka nahaja na desni stranici kvadrata itd. Ko najdemo stranico, z razmeroma preprostimi formulami določimo koordinati x in y iskane tipke.

```
public class Tipkovnica {

    public static void main(String[] args) {
        ...
        for (int i = 1; i <= dolzinaZaporedja; i++) {
            ...
            switch (oblika) {
                ...
                case 4: {
                    if (tipka == 0) {
                        vrZdaj = d / 2;
                        stZdaj = d / 2;
                        break;
                    }

                    int prvaVKvadratu = 1;
                    int stTipkVKvadratu = 8;
                    int vrPrva = d / 2 - 1;
                    int stPrva = d / 2 - 1;
                    int stranicaKvadrata = 2;

                    while (tipka >= prvaVKvadratu + stTipkVKvadratu) {
                        prvaVKvadratu += stTipkVKvadratu;
                        stTipkVKvadratu += 8;
                        vrPrva--;
                        stPrva--;
                        stranicaKvadrata += 2;
                    }

                    int odmik = tipka - prvaVKvadratu;
                    if (odmik < stranicaKvadrata) {
                        vrZdaj = vrPrva;
                        stZdaj = stPrva + odmik;
                    } else if (odmik < 2 * stranicaKvadrata) {
                        vrZdaj = vrPrva + odmik - stranicaKvadrata;
                        stZdaj = stPrva + stranicaKvadrata;
                    } else if (odmik < 3 * stranicaKvadrata) {
                        vrZdaj = vrPrva + stranicaKvadrata;
                        stZdaj = stPrva + stranicaKvadrata -
                            (odmik - 2 * stranicaKvadrata);
                    } else {
                        vrZdaj = vrPrva + stranicaKvadrata -
                            (odmik - 3 * stranicaKvadrata);
                        stZdaj = stPrva;
                    }
                    break;
                }
                ...
            }
            ...
        }
    }
}
```