

[Pregledna plošča](#) / [Moji predmeti](#) / [aps2uni](#) / 25. marec - 31. marec / [Izziv 4](#)

<b>Začeto dne</b>	torek, 26. marec 2019, 10:08
<b>Stanje</b>	Zaključeno
<b>Dokončano dne</b>	ponedeljek, 1. april 2019, 00:00
<b>Porabljeni čas</b>	5 dni 12 ure
<b>Točke</b>	1,00/1,00
<b>Ocena</b>	<b>10,00</b> od možne ocene 10,00 ( <b>100%</b> )

Napišite program, ki tabelo 32 bitnih celih števil uredi glede na število bitov, ki so enaki 1 v dvojiški predstavitvi tega števila.

Npr. tabela 6 5 4 1 3 bi bila urejena 4 1 6 5 3, ker 4 = 100, 1 = 1, 6 = 110, 5 = 101 in 3 = 11. Števila, ki imajo enako število enic ostanejo v istem vrstnem redu kot pred urejanjem.

Za urejanje uporabite stabilno urejanje s štejetjem (counting sort). Na standardnem vhodu boste najprej prejeli dolžino tabele  $n$ , za tem pa  $n$  celih števil.

Na standardni izhod izpisujete potek vpisovanja elementov v končno (urejeno tabelo). Za vsak zapis zapišete v eni vrstici par ( $e1$ ,  $pos$ ), kjer je  $e1$  število, ki ga zapisujete,  $pos$  pa indeks v tabeli kamor ta element zapisujete. V zadnji vrstici izpišite še urejeno tabelo ločeno s presledki.

For example:

Input	Result
5	(3,4)
6 5 4 1 3	(1,1)
	(4,0)
	(5,3)
	(6,2)
	4 1 6 5 3

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2
3 public class Izziv4 {
4
5     public static void main(String [] args) {
6         Scanner sc = new Scanner (System.in);
7         int dolzina = sc.nextInt();
8         int tabela [] = new int [dolzina];
9
10        for (int i = 0; i < tabela.length; i++)
11            tabela[i] = sc.nextInt();
12
13        int [] urejena = uredi(tabela, dolzina);
14
15        for (int i = 0; i < dolzina; i++)
16            System.out.print(urejena[i] + " ");
17    }
18
19    private static int [] uredi (int [] tabela, int dolzina) {
20        int [] urejena = new int [dolzina];
21        int [] biti = new int [32];
22        int [] urejena2 = new int [32];
    
```

	Input	Expected	Got	
✓	8 8 7 6 5 4 3 2 1	(1,3) (2,2) (3,6) (4,1) (5,5) (6,4) (7,7) (8,0) 8 4 2 1 6 5 3 7	(1,3) (2,2) (3,6) (4,1) (5,5) (6,4) (7,7) (8,0) 8 4 2 1 6 5 3 7	✓
✓	10 820 37 908 764 797 460 641 719 704 636	(636,7) (704,2) (719,9) (641,1) (460,5) (797,6) (764,8) (908,4) (37,0) (820,3) 37 641 704 820 908 460 797 636 764 719	(636,7) (704,2) (719,9) (641,1) (460,5) (797,6) (764,8) (908,4) (37,0) (820,3) 37 641 704 820 908 460 797 636 764 719	✓

	Input	Expected	Got	
✓	15 703 224 248 924 122 727 495 440 462 89 107 879 626 568 405	(405,8) (568,2) (626,7) (879,14) (107,6) (89,1) (462,10) (440,5) (495,13) (727,11) (122,4) (924,9) (248,3) (224,0) (703,12) 224 89 568 248 122 440 107 626 405 924 462 727 703 495 879	(405,8) (568,2) (626,7) (879,14) (107,6) (89,1) (462,10) (440,5) (495,13) (727,11) (122,4) (924,9) (248,3) (224,0) (703,12) 224 89 568 248 122 440 107 626 405 924 462 727 703 495 879	✓
✓	20 381 435 567 206 342 217 175 508 18 158 374 773 711 685 358 516 687 732 703 613	(613,8) (703,19) (732,15) (687,18) (516,1) (358,7) (685,14) (711,13) (773,2) (374,12) (158,6) (18,0) (508,17) (175,11) (217,5) (342,4) (206,3) (567,10) (435,9) (381,16) 18 516 773 206 342 217 158 358 613 435 567 175 374 711 685 732 381 508 687 703	(613,8) (703,19) (732,15) (687,18) (516,1) (358,7) (685,14) (711,13) (773,2) (374,12) (158,6) (18,0) (508,17) (175,11) (217,5) (342,4) (206,3) (567,10) (435,9) (381,16) 18 516 773 206 342 217 158 358 613 435 567 175 374 711 685 732 381 508 687 703	✓

Lemma. ALemma. ALemma. A

```
import java.util.Scanner;

public class Izziv4 {

    private static int[] countingSort(int[] a){
        int[] result = new int[a.length];
        int[] counts = new int[33];
        for (int i : a)
            counts[Integer.bitCount(i)]++;

        int cumSum = 0;
        for (int i=0; i<33 ; i++ ) {
            cumSum += counts[i];
            counts[i] = cumSum-1;
        }

        for (int i = a.length-1; i>= 0 ; i--) {
            int pos = Integer.bitCount(a[i]);
            result[counts[pos]] = a[i];
            System.out.println("(" + a[i] + ", " + counts[pos] + ")");
            counts[pos]--;
        }
        return result;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
        for (int i = 0; i < n ; i++ )
            a[i] = sc.nextInt();

        a = countingSort(a);

        for (int i : a )
            System.out.print(i + " ");
    }
}
```

Pravilno

Marks for this submission: 1,00/1,00.

[◀ Kviz 4](#)[5 - Deli in vladaj ▶](#)