

1. Sestrojte algoritmus, který načte 230 prvků (celá čísla), určí maximální a minimální hodnotu z načtených čísel, u maximální hodnoty, určí pořadí prvního výskytu a u minimální hodnoty určí pořadí posledního výskytu.
-

ŘEŠENÍ

```
Scanner scanner = new Scanner(System.in);
int max = Integer.MIN_VALUE, min = Integer.MAX_VALUE;
int maxPos = -1, minPos = -1;
int input;
for (int i = 0; i < 230; i++)
{
    input = scanner.nextInt();
    if (input > max)
    {
        max = input;
        maxPos = i;
    }
    if (input < min || input == min)
    {
        min = input;
        minPos = i;
    }
}
System.out.printf("Maximum: %d (první výskyt: %d)\n", max, maxPos);
System.out.printf("Minimum: %d (poslední výskyt: %d)\n", min, minPos);
```

Časová složitost: $\mathcal{O}(C)$

2. Sestrojte algoritmus, který bude načítat postupně hodnoty (přirozená čísla), načtení bude ukončeno 0. Z načtených hodnot určí součet všech, které jsou dělitelné 5 a z těch také určí maximum.
-

ŘEŠENÍ

```
Scanner scanner = new Scanner(System.in);
int input, sum = 0, max = Integer.MIN_VALUE;
do
{
    input = scanner.nextInt();
    if (input > 0 && input % 5 == 0)
    {
        sum += input;
        if (input > max)
        {
            max = input;
        }
    }
}
while (input != 0);
System.out.printf("Soucet: %d; maximum: %d\n", sum, max);
```

Časová složitost: $O(n)$

3. Odkrokuje následující algoritmus

```
{  
    int o = 4;  
    int p = 5;  
    int q = 5;  
    for (i = 1; i <= 5; i = i + 1)  
    {  
        o = o + 1;  
        p = p + 2;  
        q = q + o;  
    }  
}
```

o = p = q =

ŘEŠENÍ

i	o	p	q
1	5	7	10
2	6	9	16
3	7	11	23
4	8	13	31
5	9	15	40

Výsledné hodnoty:

$o = 9$
 $p = 15$
 $q = 40$

Časová složitost: $\mathcal{O}(C)$

4. Doplňte následující část kódu, který z konečného počtu p čísel, postupně načítaných do jednoduché proměnné číslo, vypíše každé páté liché číslo, které se mezi zadanými čísly vyskytne:

```
{
    int p = input.nextInt();
    int cislo;
    int pocet = 0;
    int i;
    for (i = 1; i <= p; i = i + 1)
    {
        .....;
        if (.....)
        {
            pocet = pocet + 1;
            if (.....)
            {
                System.out.println(cislo);
                pocet = .....;
            }
        }
    }
}
```

ŘEŠENÍ

```
{
    int p = input.nextInt();
    int cislo;
    int pocet = 0;
    int i;
    for (i = 1; i <= p; i = i + 1)
    {
        cislo = input.nextInt();
        if (cislo % 2 == 1)
        {
            pocet = pocet + 1;
            if (pocet == 5)
            {
                System.out.println(cislo);
                pocet = 0;
            }
        }
    }
}
```

Časová složitost: $O(n)$