**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**BEECROWD – QUESTÃO 2675**

**N** cartas numeradas de 1 a **N**. Você recebe estas cartas embaralhadas, e deve pegar uma a uma e colocá-las em um monte, mas só pode colocar uma carta no monte, se a carta que estiver no topo do monte for menor que esta, caso contrário você vai roubando do monte cartas até achar uma menor. No final sua pontuação é a soma das cartas que roubou. Como este jogo estava cansativo, foi solicitado a você um algoritmo que, dada a sequência de cartas, indique a soma que você terá roubado.

**ENTRADA(S):**

A entrada contém vários casos de testes, cada caso de teste contém duas linhas, na primeira o número N (0 < N ≤ 105) (90% dos casos de entrada são 0 < N ≤ 1.000) indicando a quantidade de cartas. Na segunda linha N valores, numerados de 1 a N em uma ordem qualquer. Os casos de teste terminam com o fim da entrada.

**Testes:**

5  
3 2 5 1 4

5  
1 2 3 4 5

3  
3 2 1

**SAÍDA(S):**

*Para cada caso de teste, um valor indicando a soma que receberá no final do jogo em uma linha.*

10

0

5

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**CÓDIGO FONTE**

import *java.util.Scanner*; *// Scanner class*

*public* *class* StealPack { *// StealPack class*

*public* *static* void main(String[] args) { *// main method*

        Scanner scanner = new Scanner(System.in); *// scanner object*

        while (scanner.hasNextInt()) { *// while there is another integer*

            int N = scanner.nextInt(); *// store number of items*

            long ans = 0; *// answer*

            int[] n = new int[N]; *// array of items*

            for (int i = 0; i < N; i++) { *// loop through items*

                n[i] = scanner.nextInt(); *// store item*

            } *// end for*

            int mesa = n[N - 1]; *// store last item*

            for (int i = N - 2; i >= 0; i--) { *// loop through items*

                if (n[i] < mesa) { *// if item is smaller than mesa*

                    mesa = n[i]; *// set mesa to item*

                } else { *// if item is bigger than mesa*

                    ans += n[i]; *// add item to answer*

                } *// end if*

            } *// end for*

            System.out.println(ans); *// print answer*

        } *// end while*

        scanner.close(); *// close scanner*

    } *// end main*

}

Após preencher o array de itens, definimos o último item do array como aquele que está sobre a mesa. Então, percorrendo todo o array de trás pra frente, caso encontremos algum item cujo valor seja menor que aquele na mesa, o item da mesa (do topo do monte) passa a ser o novo item encontrado. Do contrário, adicionamos aquele item à variável de resposta que, ao fim do loop, manterá apenas aqueles cujo valor é maior ou igual ao valor da mesa recorrente.