

Dado um vetor de inteiros e uma quantidade de elementos, indique a maior soma que pode ser calculada com essa quantidade, navegando-se pelas suas extremidades.

Você pode pegar:

- A quantidade a partir do início;
- A quantidade a partir do final;
- Parte da quantidade do início e parte do final;

Só não pode pegar diretamente do meio.

Exemplos:

- [8,7,8,9,1,2,1,4], 4: 32 (8+7+8+9)
- [5,0,-1,3,-2,5,7,9],4: 26 (5+9+7+5)
- [7,1,3,-1,3,0,7,1],5: 19 (7+1+3+1+7)
- [5,0,-1,3,8,5,7,9],4: 29 (8+5+7+9)

Python:

```
def calc(v,n):  
    #  
    return soma  
  
if __name__=="__main__":  
    print(calc([8,7,8,9,1,2,1,4],4)) # 32
```

Java:

```
public class Maxsum {  
    private int calc(int [] v,int n) {  
        return soma;  
    }  
    public static void main(String [] args) {  
        Maxsum a = new Maxsum();  
        System.out.println(a.calc(new int [] {8,7,8,9,1,2,1,4},4)); // 32  
    }  
}
```