[Esercizio 1] Bandiera

Scrivete un programma che legga da linea di comando un numero intero n>0 e che stampi una bandiera di n+1 righe costituite ciascuna da n caratteri contenente il motivo diagonale mostrato negli esempi.

Esempi di esecuzione

```
>go run Es1_Bandiera.go 7
+++++++
*++++++
**+++++
***++++
****+++
*****++
******+
*******

>go run Es1_Bandiera.go 1
+
*

>go run Es1_Bandiera.go 5
+++++
*++++
**+++
***++
****+
*****
```

[Esercizio 2] Numero Migliore

Scrivere un programma che riceva in input come argomento del main un numero intero positivo n e che calcoli e stampi il massimo numero ottenibile rimuovendo 3 cifre decimali consecutive da n.

Se n ha meno di 3 cifre, il programma deve stampare il valore float64 NaN (ottenibile con una funzione del package math).

Per esempio, si consideri n = 45678. Il massimo numero ottenibile rimuovendo 3 cifre è 78 poiché i numeri ottenuti rimuovendo 3 cifre consecutive sono: 45 (rimuovendo “678”), 48 (rimuovendo “567”) e 78 (rimuovendo “456”)

Notate che non è possibile considerare permutazioni delle cifre, quindi 84 non è una soluzione ammissibile.

Esempi di esecuzione

```
>go run Es2_Migliore.go 32
numero migliore: NaN

>go run Es2_Migliore.go 327
numero migliore: 0

>go run Es2_Migliore.go 45678
numero migliore: 78

>go run Es2_Migliore.go 32751960
numero migliore: 51960

>go run Es2_Migliore.go 3275
numero migliore: 5
```

[Esercizio 3] Albero di Natale

Scrivere un programma che legga da linea di comando un intero n e un carattere c a scelta e che stampi un albero di Natale di altezza n composto unicamente dal carattere c.

L'albero deve essere composto da una occorrenza del carattere nella prima riga, 3 occorrenze del carattere nella seconda, 5 nella terza, e così via. Nell'ultima riga ci saranno 2*n-1 occorrenze del carattere.

Se n<=0 il programma stampa il messaggio "Albero troppo piccolo".

Esempi di esecuzione

```
>go run Es3_Natale.go -1 *  (Linux: $go run Es3_Natale.go -1 '*')

Albero troppo piccolo

>go run Es3_Natale.go 0 *  (Linux: $go run Es3_Natale.go 0 '*')

Albero troppo piccolo

>go run Es3_Natale.go 1 *  (Linux: $go run Es3_Natale.go 1 '*')
*

>go run Es3_Natale.go 3 #  (Linux: $go run Es3_Natale.go 3 '#')
  #
 ###
#####

>go run Es3_Natale.go 6 +
     +
    +++
   +++++
  +++++++
 +++++++++
++++++++++
```

[Esercizio 4] Massimo comune divisore di due numeri

Scrivere un programma che legga da standard input due numeri interi a e b e che stampi il loro massimo comune divisore mcd(a,b).

Il massimo comune divisore di due numeri a e b è il massimo numero n che divide sia a che b.

Se a e/o b sono numeri negativi si deve considerare il loro valore assoluto.

Il valore assoluto di x, indicato con |x|, è uguale a x se x>=0, è uguale a -x se x<0. Ad esempio, |3| = 3 poiché 3>=0, |-4| = 4 poiché -4 <0.

Quindi:

mcd(63,-27) = mcd(63,|-27|) = mcd(63,27) = 9

Casi particolari:

- se n è un numero intero qualsiasi (positivo, negativo o nullo): mcd(1,n) = 1
- se n è un numero intero qualsiasi (positivo, negativo o nullo): mcd(0,n)=|n| (infatti |n| è un divisore di n, e n è divisore di zero); da ciò segue che mcd(0,0)=0

Esempi di esecuzione

```
>go run Es4_Mcd.go

27 63

mcd(27,63)=9

>go run Es4_Mcd.go

0 0

mcd(0,0)=0

>go run Es4_Mcd.go

4 0

mcd(4,0)=4

>go run Es4_Mcd.go

76324746 0

mcd(76324746,0)=76324746

>go run Es4_Mcd.go

6 1

mcd(6,1)=1
```

[Esercizio 5] Biscione di cifre

Scrivere un programma che legga da linea di comando due interi strettamente positivi `m` ed `n` e che stampi un biscione di cifre (da `0` a `9`) di larghezza `m` e altezza `n`, come indicato negli esempi sottostanti.

Si noti che, dopo aver stampato la cifra `9`, il biscione continua ripartendo dalla cifra `0`.

Esempi

```
>go run Es5_Biscione.go 1 1
0

>go run Es5_Biscione.go 1 4
0
1
2
3

>go run Es5_Biscione.go 5 4
0-1-2-3-4
        5
0-9-8-7-6
1

>go run Es5_Biscione.go 5 7
0-1-2-3-4
        5
0-9-8-7-6
1
2-3-4-5-6
        7
2-1-0-9-8
```