Laboratorio di Fondamenti di Informatica Lab04_2015-10-27

Anno accademico 2015/2016

1. Il cifrario di Cesare

Uno dei modi più semplici (e ormai meno efficaci) di cifrare un messaggio consiste nell'applicare una trasformazione "rotatoria" alle lettere che lo compongono, sostituendo ogni lettera con la lettera che si trova, nell'ordinamento alfabetico, k posizioni più avanti. Le ultime k-1 lettere dell'alfabeto sono trattate "rientrando" dall'inizio dell'alfabeto. Esempio:

```
offset? 2
mi piace questo corso
ok rkceg swguvq eqtuq
```

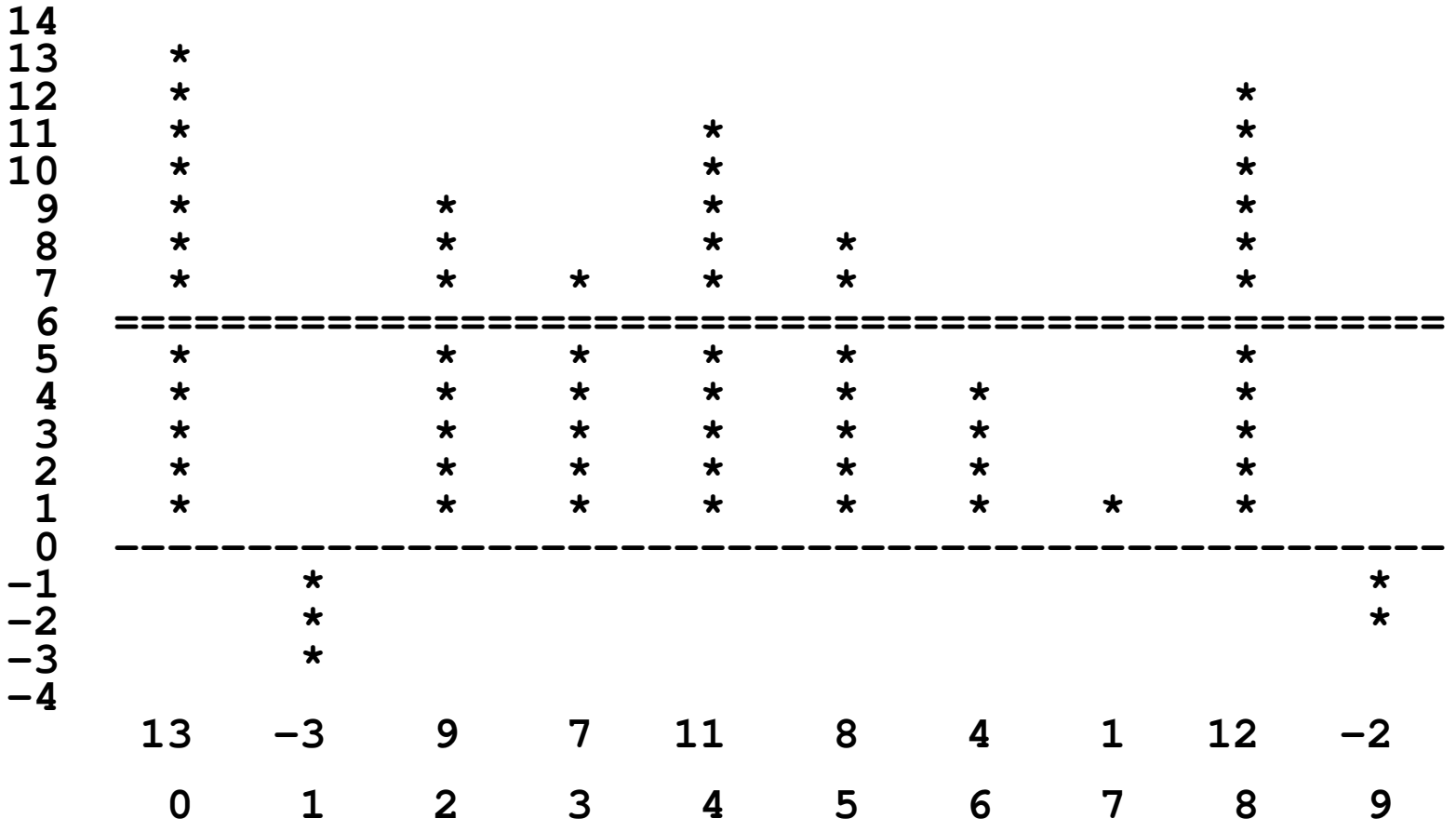
Per k=0 il messaggio resta inalterato, per k=1 ogni lettera è sostituita dalla successiva (e la 'z' dalla 'a'), per k=-1 dalla precedente (e la 'a' dalla 'z').

Si scriva un programma C che legga un intero k e una sequenza di caratteri e restituisca su stdout la sequenza cifrata. (per brevità, si considerino solo i caratteri minuscoli).

Si noti che, essendo 26 le lettere dell'alfabeto inglese, per k=13 l'algoritmo di cifratura è identico a quello di decifratura, e che per k = 26 il messaggio resta inalterato come per k=0. Si verifichi la "robustezza" della soluzione proposta utilizzando valori di k pari a 26 o superiori (per k=27 il comportamento dev'essere uguale a quello con k = 1)..

2. (ancora) Istogrammi

Scrivere un programma che visualizzi (come illustrato in calce) i valori di un array di 10 interi *qualsiasi* (cioè anche negativi o nulli), evidenziandone (con una "riga" di '=') anche il valor medio (approssimato per troncamento –nell'esempio è 6).



3. Scomposizione in fattori primi

Scrivere un programma che legge un numero intero positivo *n* da stdin e ne calcola e visualizza la scomposizione in fattori primi. Ad esempio:

```
Numero: 29750
29750 = 2 x 5^3 x 7 x 17
```

Progettare il programma in autonomia è difficile per chi non ha esperienza. Volendo un aiutino, però…

Suggerimenti (che trasformano il nero in rosso!):

In assenza di idee migliori, si può procedere come segue:

1. Il programma «genera» tutti i numeri primi *p* compresi tra 2 e n
2. Per ciascun valore di p, se p è un divisore di n, si conta quante volte occorre dividere n per p prima di ottenere un numero non più divisibile per p, di modo da capire con quale esponente sia da considerare il fattore p
3. La scomposizione si può dunque stampare via via che si procede a calcolarla

Il passo 1 può sembrare non banale, ma è sufficiente considerare tutti i numeri k da 2 a n e valutare se hanno divisori compresi tra 2 e k-1. *In alternativa, si può ragionare delle proprietà dei numeri…*

4. Numeri Fortunati

Scrivere un programma che stampi tutti i *numeri* ***fortunati*** minori di 200.

Un **numero fortunato** è un numero naturale in un insieme generato come segue:

Si inizia con la successione di tutti i numeri interi:

1, 2, 3, 4, 5, 6, 7, 8, 9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25…

Si eliminano poi **tutti i *secondi* numeri** (marcati in grigio: restano solo i dispari):

1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43…

Il ***secondo*** termine rimasto in questa sequenza **è 3**. Si eliminano dunque **tutti i *terzi* numeri** rimasti nella sequenza (marcati in rosso):

1, 3, 7, 9, 13, 15, 19, 21, 25, 27, 31, 33, 37, 39, 43…

Il ***terzo*** termine rimasto ora **è 7**. Si eliminano dunque **tutti i *settimi* numeri** rimasti (marcati in blu):

1, 3, 7, 9, 13, 15, 21, 25, 27, 31, 33, 37, 43…

il ***quarto*** termine rimasto ora è 9, si eliminano quindi tutti i ***noni*** numeri (marcati in giallo). Poi si eliminano i **13simi**, poi i **15simi**, poi i **21simi**, …

Ripetendo la procedura "indefinitamente", restano solo i numeri **fortunati**:

1, 3, 7, 9, 13, 15, 21, 25, 31, 33, 37, 43, 49, 51, 63, 67, 69, 73, 75, 79, 87, 93, 99…