Esercizio 1

Scrivere un programma che acquisisca da tastiera una sequenza di caratteri terminata da '!' e riporti sul monitor una sequenza derivata dalla precedente secondo le regole dell'alfabeto *farfallino* (ogni vocale viene raddoppiata inserendovi in mezzo una *f*).

Esempio:

quanto mi piace questo corso!

q**ufuafa**nt**ofo** m**ifi** p**ifiafa**c**efe** q**ufuefe**st**ofo** c**ofo**rs**ofo**!

Per semplicità, si considerino solo le lettere minuscole.
Ogni carattere diverso dalle vocali deve restare inalterato.
Si acquisisca da tastiera un carattere alla volta.

Estensioni e Varianti

- Regola del vamavvo: tutte le 'r' sino sostituite con delle 'v'
- Variante fiorentina: tutte le 'c' che siano iniziali di parola siano sostituite con 'h'
- Variante urlata: la frase sia trascritta interamente in lettere maiuscole
- Applicandole tutte insieme, ad esempio:
  che corso! → HHEFE HOFOVSOFO!

Esercizio 2

Dato un numero positivo Q,

- scrivere la sua rappresentazione in binario naturale, applicando il tradizionale algoritmo per divisioni successive
  (per convenzione, in questo esercizio l'output si intende corretto se letto da destra a sinistra);
- indicare anche il minimo numero di bit utilizzato.

Es:

Input: 19 in decimale, Output: 11001 in binario.

Stesura informale dell'algoritmo: (lasciata come esercizio)

ESERCIZIO 3

Si scriva un programma in linguaggio C che letto un numero intero positivo dallo standard input, visualizzi a terminale il quadrato del numero stesso facendo uso soltanto di operazioni di somma.
Si osservi che il quadrato di ogni numero intero positivo N può essere costruito sommando tra loro i primi N numeri dispari.
Esempio: N = 5; $N^2 = 1 + 3 + 5 + 7 + 9 = 25$.

Stesura informale dell'algoritmo:
Premessa:
l' idea di soluzione è quella di scandire i primi N numeri dispari esprimendoli nella forma (i+i+1) al variare dell'intero i tra 0 e N-1 e accumulare la loro somma man mano che si procede nella loro scansione in un'altra variabile.
$N^2 = (2\cdot0+1) + ... + (2\cdot i+1) + ... + (2\cdot(N-1)+1)$;
quindi si utilizzeranno almeno 3 variabili: N, i, S rispettivamente per il numero, il contatore e l' accumulatore.

1. Inizio dell' algoritmo
2. Leggi un numero intero positivo dallo standard input.
3. Inizializza un contatore i a 0
4. Inizializza un accumulatore S a 0
5. Finché il valore del contatore è minore del numero letto
    a. Somma all' accumulatore il doppio del valore del contatore incrementato di 1
    b. Incrementa il contatore
    c. Torna al punto 5.
6. Stampa a terminale il valore dell' accumulatore
7. Fine dell' algoritmo

ESERCIZIO 4

Si definisce *Triangolare* un numero costituito dalla somma dei primi N numeri interi positivi per un certo N.
Esempio: per Q = 10 si ha Q = 1 + 2 + 3 + 4, da cui N=4.
Scrivere un programma C che stabilisca se un numero intero positivo Q, letto dallo standard input, è un numero triangolare o meno, utilizzando soltanto operazioni tra numeri interi.
In caso affermativo stampare a video il numero inserito e il massimo degli addendi che lo compongono.

Stesura informale dell'algoritmo

Idea di soluzione: se Q – 1 – 2 – 3 – ... – i – ... – N == 0 per un certo N allora Q è Triangolare.

1. leggi il numero positivo Q dallo standard input
2. inizializza un contatore i a zero;
3. memorizza in una variabile S il valore della variabile in ingresso.
4. Finché il numero S è maggiore di zero
    4.1. incrementa di 1 il valore del contatore
    4.2. sottrai a S il valore del contatore i
    4.3. torna a 4.
5. Se (il valore residuo di S è zero) allora
    5.1. il numero è triangolare
    5.2. il valore del massimo degli addendi è uguale al contatore i
    5.3. la variabile Q contiene il valore della variabile in ingresso
6. altrimenti il numero NON è triangolare.

Nota: è buona norma, in generale, non modificare le variabili contenenti i dati in ingresso perché può accadere che sia necessario accedere a tali valori in diversi punti del programma.
(Nel caso appena mostrato senza l' ausilio di un' altra variabile S non sarebbe possibile, al termine della computazione, stampare a video il valore del numero inserito -- così come richiesto dalla traccia del problema.)

Esercizio 5

Scrivere un programma che legge da stdin una sequenza (di lunghezza a priori illimitata) di numeri interi positivi, terminata da 0, e indica, alla fine della sequenza, qual è la lunghezza della massima sottosequenza di numeri consecutivi in ordine crescente.

Esempi:

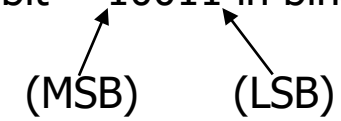
13 3 8 4 5 1 17 0
Lung. max = 2

21 19 18 14 9 6 4 3 0
Lung. max = 1

2 1 3 6 8 5 1 12 18 17 0
Lung. max = 4

Esercizio 6

Dato un numero positivo Q, scrivere la sua rappresentazione in binario naturale, indicando anche il minimo numero di bit utilizzato.
Esempio:
Input: 19 in decimale, Output: con 5 bit = 10011 in binario.



Esercizio 7

Scrivere un programma che legge un intero positivo n da stdin e verifica se n può essere scomposto nella somma di due quadrati (verifica cioè se a,b,n | $a^2+b^2=n$).
Se sì, stampare a video la scomposizione.

Esempi:

2 ==> 2 = 1 + 1 = $1^2 + 1^2$
28 ==> NON SCOMPONIBILE
146 ==> 146 = 25 + 121 = $5^2 + 11^2$

a) Mostrare, quando ve ne è più di una, tutte le diverse scomposizioni dello stesso numero (ad esempio 50 ha due scomposizioni, 1+49 e 25+25, mentre 5525 è il primo numero ad avere ben sei diverse scomposizioni e 8125 è il primo ad averne esattamente cinque).

b) Verificare anche la scomponibilità in somma di tre quadrati.