

Fai una copia del programma che hai realizzato per l'esercitazione 4 da modificare in questa occasione. Metti la copia in una *directory* di nome `alienRover` e assegna il nome `<cognome_1>.<cognome_2>.c` al *file* con il codice sorgente, dove, evidentemente `<cognome_>` rappresenta il cognome di uno degli autori.

Nella versione odierna il rover parte sempre da fermo da una quota fissata pari a $h = 200$ m. Il passo di simulazione ha una durata costante $dt = 5$ ms. Ricordiamo che la quota $y(t)$ raggiunta al tempo $t > 0$ e la corrispondente velocità $v(t)$ sono

$$y(t) = h - \frac{1}{2}gt^2, \quad v(t) = gt,$$

dove $g = 3.711 \text{ ms}^{-1}$ è l'accelerazione di gravità su Marte. Il programma deve fare i passi seguenti.

- (1) Partendo da $t = 0$, calcola la posizione e la velocità del rover a tutti gli istanti di tempo successivi, intervallati di dt costante, memorizzando i valori delle variabili che rappresentano queste grandezze in altrettanti *array* di taglia opportuna. Il calcolo degli elementi degli *array* si arresta quando il rover tocca il suolo.
- (2) Una volta che il rover abbia toccato il suolo il programma stampa il messaggio `# rover atterrato. Velocita' all'impatto:` , seguito dall'indicazione della velocità raggiunta al momento dell'impatto espressa in km/h.
- (3) La presenza di vento nell'atmosfera di Marte è simulata come segue: ogni volta che il programma esegue un passo di simulazione, una raffica di vento sposta il rover a destra o a sinistra, in modo casuale, di una quantità proporzionale alla differenza di quota $\delta = y(t) - y(t - dt)$ tra l'istante t della simulazione e quello precedente.
- (4) Il programma, per ciascuna delle posizioni $y(t)$ calcolate in precedenza, determina il valore della coordinata orizzontale $x(t)$ del rover durante la caduta, prendendo come origine la posizione di partenza.
- (5) La simulazione è ripetuta N volte, con N scelto dall'utente. Tutti i valori di $x(t)$ e $y(t)$ sono stampati sullo schermo. Mandi questi dati su un *file* e prova a fare un grafico di $y(t)$ in funzione di $x(t)$ usando `gnuplot`.
- (6) Un *array* di 80 componenti è usato per rappresentare un istogramma delle coordinate alle quali è caduto il rover. L'indice dell'*array* rappresenta la coordinata orizzontale raggiunta dal rover espressa in metri determinato in modo tale che l'indice 40 rappresenti la coordinata $x = 0$. In questa maniera il primo canale dell'istogramma contiene il numero di eventi in cui il rover si è allontanato di una distanza d compresa tra 39 e 40 m a sinistra, mentre l'ultimo elemento dell'*array* contiene il numero di eventi in cui il rover si è allontanato verso destra di una distanza compresa tra 38 e 39 m. Tenete conto del fatto che, in linea di principio, il rover può atterrare a distanza maggiori, in modulo, di 40 m.
- (7) Al termine della simulazione si calcola il massimo numero di eventi contenuto in un canale e si stampa l'istogramma in forma grafica. Detto x_i il numero di eventi nel canale i dell'istogramma, per farlo si stampano 80 colonne di caratteri `#`, ciascuna di altezza x_i . Verifica il risultato scrivendo su un *file* `istogramma.dat` il contenuto dell'istogramma, generando un grafico usando il comando `plot 'istogramma.dat' with boxes` di `gnuplot`.

Il team che ha la responsabilità scientifica della missione ha reso pubblici una serie di risultati, tra cui una serie di segnali radio captati durante la discesa. Un gruppo di ufologi sostiene che sul pianeta esiste vita intelligente in quanto, dopo aver trasformato secondo regole opportune (ma arbitrarie) i segnali in caratteri, è riuscito a individuare parole presenti nel nostro vocabolario. Segno evidente, secondo costoro, che i marziani stanno imparando la nostra lingua e cercano di comunicare.

I fisici che lavorano alla missione, tuttavia, vogliono verificare questa ipotesi nel modo seguente: generando a caso una sequenza di un milione di caratteri maiuscoli dell'alfabeto latino, includendo lo spazio e il segno di *a capo*, si produce un testo nel quale si cercano parole presenti nel vocabolario. Scrivi il programma che produce questo testo e analizzalo alla ricerca di parole non troppo lunghe: per esempio CIAO. Per una ricerca più agevole puoi usare il comando `grep` del sistema operativo: `grep CIAO a.dat` cerca la stringa CIAO nel *file* di testo `a.dat` e scrive la riga che la contiene sullo schermo.