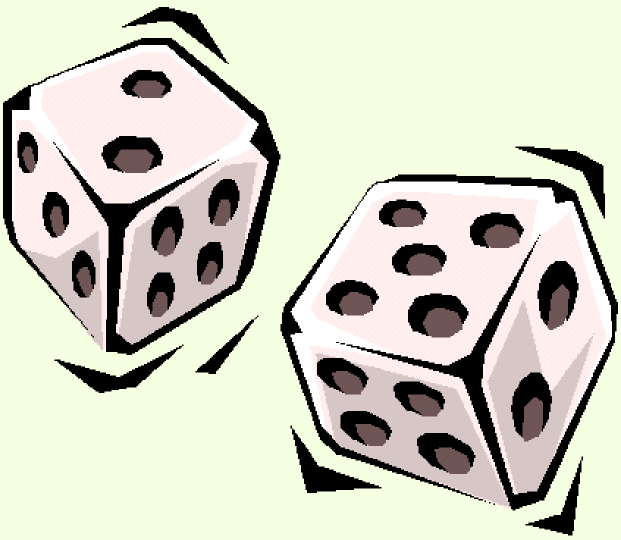
Studio sulla funzione "rand" del linguaggio C



di ***Cavenaghi Mattia***

matricola 640926

Indice:

❶ **Introduzione**

- ♣ **Le regole del Craps** pag. 2
- ♥ **Il codice del gioco** pag. 2

❷ **La risoluzione matematica**

- ♣ **I concetti di base** pag. 5
- ♥ **La probabilità condizionata** pag. 6
- ♠ **Il teorema di Bernoulli per le prove ripetute** pag. 7
- ♦ **Il test del χ^2** pag. 10

❸ **La risoluzione con il calcolatore**

- ♣ **Introduzione a MatLab** pag. 12
- ♥ **Il codice del programma d’analisi**
  - ✓ ***I concetti di base*** pag. 13
  - ✓ ***La probabilità condizionata*** pag. 14
  - ✓ ***Il teorema di Bernoulli per le prove ripetute*** pag. 14
  - ✓ ***Il test del χ^2*** pag. 14

❶ Introduzione

In questo progetto, ci occuperemo di studiare un modello, che descriva la distribuzione di probabilità, della funzione “*rand*” (nel linguaggio C, randomizza), utilizzata per generare i numeri casuali.

Per rendere più concreto il nostro ragionamento, useremo il gioco d’azzardo “*craps*”, il cui codice è stato riportato di seguito (`craps.c`).

Completate queste prime due panoramiche, ci addentreremo poi, nel progetto vero e proprio, scrivendo il modello della distribuzione di probabilità e un piccolo programma di analisi fatto con il linguaggio MatLab.

♣ Le regole del Craps

Il *craps* (noto anche come *crabs*), è un gioco inglese tra un banchiere e più giocatori, che prevede l'utilizzo di due dadi: un giocatore, prima di lanciare, punta una somma che possa essere coperta da uno o due avversari; se al primo lancio totalizza 7 o 11 punti, vince.

Se invece ottiene un punteggio pari a 2, 3, o 12, perde.

Indipendentemente dal risultato il giocatore ripete la scommessa ed il lancio. Se totalizza 4, 5, 6, 8, 9, o 10, il numero raggiunto diventa il punteggio del giocatore, il quale continua a lanciare i dadi finché non ottiene 7 (perdendo sia la scommessa sia il diritto a rilanciare) oppure lo stesso risultato raggiunto nel tiro precedente (nel qual caso vince).

♥ Il codice del gioco

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>

int craps(void);

int main()
{
      int saldo = 1000;
      int puntata;
      int stato_gioco;
      char ripeti = 'y';

      do
      {
            do
            {
            printf("Ti restano %d crediti\n", saldo);
            printf("Fai la tua puntata ");
            scanf("%d",&puntata);
            }

            while( puntata > saldo );

            stato_gioco= craps();

            if(stato_gioco==1)
            {
            printf("E\' andata bene!!!\n");
            saldo += puntata;
            }
```

```
            else
            {
            printf("E\' andata male...\n");
            saldo -= puntata;
            }
            printf("\n Ora hai %d crediti\n", saldo);
            printf(" Vuoi rischiare ancora? (y/n)");
            scanf(" %c",&ripeti);
      }
while(ripeti=='y' && saldo > 0);
printf("Alla prossima!\n");
}
int tira_dadi()
{
      int dado1, dado2, somma;
      dado1 = 1+rand()%6;
      dado2 = 1+rand()%6;
      somma = dado1+dado2;
      printf("\nIl giocatore ha ottenuto %d + %d =
      %d",dado1,dado2,somma);
      return somma;
}
void stampa_risultati (int finale)
{
      if (finale == 1)
      printf("\nHai vinto!\n");
      else printf("\nHai perso...\n");
}
int craps()
{
      int stato_gioco, somma_dadi, punteggio;
      somma_dadi = tira_dadi();

      switch(somma_dadi)
      {
      case 7:
      case 11: stato_gioco=1;break;
      case 2:
      case 12: stato_gioco=2; break;
      default: stato_gioco=0;
      punteggio=somma_dadi;
      }
      while(stato_gioco==0)
      {
      somma_dadi=tira_dadi();
      if (somma_dadi== punteggio)
      stato_gioco=1;
      else if (somma_dadi==7)
      stato_gioco=2;
      }
      stampa_risultati(stato_gioco);
      return stato_gioco;
}
```

Questo programma, che simula il gioco del *craps*, esegue una giocata e ritorna il risultato: 1 (vinto) o 2 (perso). All'avvio, si ha un budget di 1000 crediti, si digita la cifra da puntare ed il codice, genera una somma, ossia la combinazione dei dadi, che il generatore ha prodotto.

Dopo il "lancio", qualunque sia il risultato ottenuto, il programma chiede se si vuol continuare a giocare o se si vuol terminare la partita. Se il credito, è maggiore di 0 allora si prosegue con il gioco; se si è esaurito il credito, il programma termina.

Il cuore del nostro programma, è la parte di generazione dei numeri casuali, contemplato nella sezione `tira_dadi`:

```
...
int tira_dadi()
{
      int dado1, dado2, somma;
      dado1 = 1+rand()%6;
      dado2 = 1+rand()%6;
      somma = dado1+dado2;
      printf("\nIl giocatore ha ottenuto %d + %d =
      %d",dado1,dado2,somma);
      return somma;
}
...
```

La variabile `somma`, dà la somma dei risultati che la funzione *rand* genera.

`rand()` genera numeri pseudo-casuali, cioè numeri deterministici che "sembrano" casuali; affinché, ad ogni esecuzione del programma, si generi un diverso valore del lancio del dado, dobbiamo variare il seme usato per inizializzare il generatore dei numeri pseudo-casuali (operazione che in gergo viene chiamata randomizzazione).

Per generare il seme, usiamo la funzione `time()` della libreria `time.h` che rappresenta l'ora corrente del giorno espressa in secondi, la quale, variando istante per istante, genera, ad ogni esecuzione un numero differente (nel nostro caso, una coppia).

Il risultato, viene poi "formattato", ossia lo forziamo ad avere un valore tra 1 e 6, tramite l'operazione `1+rand()%6;` poiché il risultato 0, non è accettabile.

❷ La risoluzione matematica

♣ I concetti di base

Con il termine *distribuzione di probabilità*, si intende la probabilità con cui una certa classe di eventi si può presentare in un esperimento. Per capire questo concetto, è necessario tracciare una tabella, contenente tutti gli *eventi* possibili del nostro *sistema*:

D_1+D_2	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

Possiamo notare, che per un sistema composto da due dadi, non truccati, ed a sei facce, vi sono 6^2, *36 eventi possibili*, aventi ognuno, una probabilità d'uscita, pari ad $\frac{1}{36}$; tali eventi vengono quindi definiti *"equiprobabili"*.

Suddividiamo ora i nostri eventi, in tre classi:

a) *Eventi favorevoli (F):* 7 ed 11, evidenziati in verde, ed aventi probabilità d'uscita pari a $\frac{8}{36} = \frac{2}{9}$;
b) *Eventi sfavorevoli (S):* 2, 3 e 12 evidenziati in rosso, ed aventi probabilità d'uscita di $\frac{4}{36} = \frac{1}{9}$;
c) *Eventi incerti (I_i):* tutti gli altri possibili casi, aventi probabilità d'uscita di $\frac{24}{36} = \frac{2}{3}$. Gli eventi incerti, possono, a loro volta, essere suddivisi in:
   - *Favorevoli* (F_{I_i}): nel caso in cui al *(n+1)-esimo* lancio, ottengo lo stesso risultato dell'*n–esimo* lancio *i*;
   - *Sfavorevoli* (S_{I_i}): nel caso in cui il *(n+1)-esimo* lancio, ottiene 7.

Queste ultime due considerazioni portano ad un'ulteriore evoluzione del nostro problema: difatti, si potrebbe verificare, dopo un evento incerto, un evento favorevole, od uno sfavorevole. Queste condizioni, non possono essere ignorate, ed è per questo, che consideriamo la seguente nuova tabella:

$D_{1_i}+D_{2_i}$	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

Calcoliamo, ora, la probabilità condizionata degli eventi incerti.

♥ La probabilità condizionata

Calcoliamo ora la *probabilità condizionata* $P(F|I_i)$, ossia, che si verifichi un evento favorevole F, dopo l'evento incerto I_i (dove i = 4, 5, 6, 8, 9, 10):

$$P(F|I_i) = \frac{P(F) \cap P(I_i)}{P(I_i)} = \frac{\frac{5}{6} \cap \frac{2}{3}}{\frac{2}{3}} = \frac{\frac{5}{6} \cdot \frac{2}{3}}{\frac{2}{3}} = \frac{5}{6}$$

Ora, dobbiamo estrarre i soli possibili eventi favorevoli degli incerti:

$$P(F_{I_i}) = P(F|I_i) \cap P(I_i) = \frac{5}{6} \cdot \frac{2}{3} = \frac{5}{9}$$

Come prima, ora calcoliamo anche la *probabilità condizionata* che si verifichi un evento sfavorevole S, dopo l'evento incerto I_i:

$$P(S|I_i) = \frac{P(S) \cap P(I_i)}{P(I_i)} = \frac{\frac{1}{6} \cap \frac{2}{3}}{\frac{2}{3}} = \frac{\frac{1}{6} \cdot \frac{2}{3}}{\frac{2}{3}} = \frac{1}{6}$$

Ed anche gli eventi sfavorevoli degli incerti:

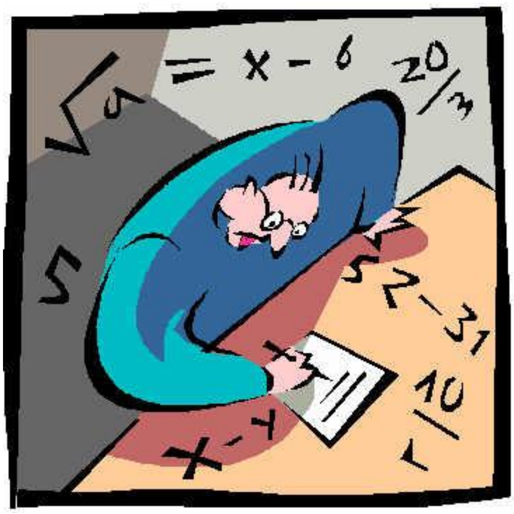
$$P(S_{I_i}) = P(S|I_i) \cap P(I_i) = \frac{1}{6} \cdot \frac{2}{3} = \frac{1}{9}$$

Dobbiamo inoltre ricordare, che la sommatoria, della probabilità di tutti gli eventi, che si possono verificare nel nostro sistema, è:

$$P_i = P(F) + P(S) + P(F_{I_i}) + P(S_{I_i}) = \frac{2}{9} + \frac{1}{9} + \frac{5}{9} + \frac{1}{9} = 1$$

Il nostro sistema, è quindi conforme alla definizione della *funzione di probabilità*:

$$\sum_{i=1}^{N} P_i = 1$$



♠ Il teorema di Bernoulli per le prove ripetute

Giunti a questo punto, possiamo calcolare la distribuzione di probabilità, grazie al *Teorema di Bernoulli;* se *p* è la probabilità (costante) di un evento in un esperimento, e se, su *n* prove eseguite, *m* risultano favorevoli all'evento, la probabilità è: $\left| \frac{m}{n} - p \right| < \varepsilon$, con ε tendente ad 1 quando *n* tende all'infinito, per qualsiasi $\varepsilon > 0$.

Generalizzando, ossia applicando l'equazione di Bernoulli alla classe degli eventi favorevoli, ricaviamo:

$$P(F) = 1 - P(S)^n = 1 - \left(\frac{7}{9}\right)^n$$

n	*P(F)*
0	0
2	0,395061
4	0,634049
6	0,778622
8	0,866080
10	0,918986
15	0,976941
20	0,993436
25	0,998131
30	0,999468
36	0,999882

Quest'ultima tabella, descrive come se ad ogni evento, fosse associata una prova, e, dopo 35 prove otteniamo sempre un risultato diverso dall'evento favorevole, saremmo portati a credere che, anche al 36° lancio, l'evento favorevole, non si verifichi.

Ma non è così; infatti il 36° lancio, da solo, è indipendente dai precedenti: e la probabilità di ogni singolo evento (lancio) continua ad essere $\frac{1}{36}$.

Difatti, il teorema di Bernoulli, concerne *serie di prove*, non *prove singole*!

Se invece di eseguire prove singole, eseguiamo delle serie di prove, da 36 o più lanci ciascuna, si avrà, nella maggior parte delle serie eseguite, in percentuale tanto maggiore quanto più è grande il numero delle serie, un numero all'incirca pari alla probabilità di ogni singolo evento, che sia esso: favorevole, sfavorevole od incerto.

"Ma allora, come facciamo a giustificare il nostro modello di distribuzione di probabilità, relativo alla funzione rand?".

Se, con l'ausilio della sola matematica, non riusciamo a fornire delle conclusioni ragionevoli, proviamo a ragionare per assurdo!

"Se, su 35 lanci, non otteniamo mai un evento favorevole, e neanche al 36° lancio, la distribuzione da noi calcolata, non è valida!".

"Inoltre, guardando le occorrenze dei vari eventi, le frequenze osservate, dovrebbero essere eguali a quelle da noi calcolate (frequenze attese)".

Eseguiamo allora *n* serie di 36 lanci di dadi ciascuna riportandone in tabella i risultati ottenuti; per semplicità, eseguiamo tre serie di 36 lanci ciascuna:

1° serie:

Lancio	Risultato	Lancio	Risultato	Lancio	Risultato
1	9	**13**	10	**25**	8
2	4	**14**	5	**26**	4
3	4	**15**	6	**27**	3
4	9	**16**	10	**28**	5
5	6	**17**	6	**29**	5
6	3	**18**	6	**30**	8
7	8	**19**	3	**31**	9
8	5	**20**	6	**32**	7
9	3	**21**	7	**33**	7
10	9	**22**	11	**34**	9
11	12	**23**	6	**35**	8
12	3	**24**	12	**36**	6

Eventi favorevoli osservati: $F_o = \frac{4}{36} = \frac{1}{9}$

Eventi sfavorevoli osservati: $S_o = \frac{7}{36}$

Eventi incerti osservati: $I_o = \frac{25}{36}$

2° serie:

Lancio	Risultato	Lancio	Risultato	Lancio	Risultato
1	4	**13**	5	**25**	7
2	4	**14**	7	**26**	7
3	9	**15**	7	**27**	11
4	8	**16**	4	**28**	12
5	4	**17**	9	**29**	7
6	4	**18**	4	**30**	7
7	12	**19**	7	**31**	9
8	8	**20**	9	**32**	6
9	11	**21**	8	**33**	7
10	7	**22**	4	**34**	6
11	7	**23**	6	**35**	4
12	10	**24**	6	**36**	3

Eventi favorevoli osservati: $F_o = \frac{12}{36} = \frac{1}{3}$

Eventi sfavorevoli osservati: $S_o = \frac{3}{36} = \frac{1}{12}$

Eventi incerti osservati: $I_o = \frac{21}{36}$

3° serie:

Lancio	Risultato	Lancio	Risultato	Lancio	Risultato
1	6	**13**	7	**25**	10
2	6	**14**	3	**26**	6
3	7	**15**	3	**27**	10
4	7	**16**	6	**28**	5
5	3	**17**	11	**29**	11
6	11	**18**	11	**30**	11
7	9	**19**	4	**31**	6
8	8	**20**	6	**32**	3
9	6	**21**	4	**33**	7
10	4	**22**	7	**34**	9
11	5	**23**	8	**35**	5
12	8	**24**	12	**36**	7

Eventi favorevoli osservati: $F_o = \frac{11}{36}$

Eventi sfavorevoli osservati: $S_o = \frac{5}{36}$

Eventi incerti osservati: $I_o = \frac{20}{36} = \frac{5}{9}$

♦ Il test del χ^2

Notiamo che le frequenze osservate sono simili a quelle da noi calcolate in precedenza. Ma non possiamo dare una definizione alla nostra *rand*, usando un metodo così "spannometrico", introduciamo quindi ,un nuovo strumento di calcolo: il *test del chi quadro (χ^2).*

Il test del χ^2 è un indice di scostamento tra una distribuzione di frequenze rilevate sperimentalmente (*frequenze osservate*), ed una distribuzione teorica (*frequenze attese*). Su tale indice, è necessario costruire, prima, delle *ipotesi*, ossia delle supposizioni non ancora realizzate, sulla quale derivare delle conseguenze.

Come ipotesi, consideriamo i nostri ragionamenti, fatti per assurdo:

1. *Se, su 35 lanci, non otteniamo mai un evento favorevole e, neanche al 36° lancio, la distribuzione da noi calcolata, non è valida.*
2. *Le frequenze osservate, dovrebbero essere simili a quelle da noi calcolate (frequenze attese).*

Per calcolare il χ^2 abbiamo bisogno di conoscere le frequenze osservate ed attese di ogni classe di eventi, già calcolate in precedenza:

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

Detto ciò possiamo scrivere:

1° serie:

	Frequenza osservate (O)	*Frequenze attese (E)*	*Totali*
Eventi favorevoli	4	8	12
Eventi sfavorevoli	7	4	11
Eventi incerti	25	24	49
	36	36	72

$$\chi^2_1 = \frac{(4 - 8)^2}{8} + \frac{(7 - 4)^2}{4} + \frac{(25 - 24)^2}{24} = 2 + 2,25 + 0,041 = 4,291$$

2° serie:

	Frequenza osservate (O)	*Frequenze attese (E)*	*Totali*
Eventi favorevoli	12	8	20
Eventi sfavorevoli	3	4	7
Eventi incerti	21	24	45
	36	36	72

$$\chi^2_2 = \frac{(12 - 8)^2}{8} + \frac{(3 - 4)^2}{4} + \frac{(21 - 24)^2}{24} = 2 + 0,25 + 0,375 = 2,625$$

3° serie:

	Frequenza osservate (O)	*Frequenze attese (E)*	*Totali*
Eventi favorevoli	11	8	19
Eventi sfavorevoli	5	4	9
Eventi incerti	20	24	44
	36	36	72

$$\chi^2_3 = \frac{(11-8)^2}{8} + \frac{(5-4)^2}{4} + \frac{(20-24)^2}{24} = 1{,}125 + 0{,}25 + 0{,}666 = 2{,}041$$

Confrontiamo ora i valori calcolati, con i valori messi a disposizione dai matematici, in apposite tabelle di distribuzione del χ^2, ricordando che abbiamo:

Gradi di Libertà (GDL) = (n° righe – 1) · (n° colonne – 1) = (3 - 1) · (2 - 1) = 2

Il valore di $\chi^2_{tabellato}$ per 2 GdL è:

df	**0.995**	**0.99**	**0.975**	**0.95**	**0.90**	**0.10**	**0.05**	**0.025**	**0.01**	**0.005**
1	/	/	0.001	0.004	0.016	2.706	3.841	5.024	6.635	7.879
2	0.010	0.020	0.051	0.103	0.211	4.605	5.991	7.378	9.210	10.597
3	0.072	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.345	12.838
...	...	...	...	...	...	...	...	...	...	...

Allora:

$\chi^2_1 < \chi^2_{tabellato}$

$\chi^2_2 < \chi^2_{tabellato}$

$\chi^2_3 < \chi^2_{tabellato}$

Possiamo confermare che le nostre ipotesi sono esatte, con un livello di probabilità del 99% e del 99,5%.

Affermiamo quindi:

1. *Il nostro modello di distribuzione di probabilità, è valido.*
2. *Le frequenze osservate, non si discostano molto dalle frequenze attese.*

Concludiamo, affermando che la funzione *"rand"* del linguaggio C, genera dei numeri pseudo-casuali, con una probabilità di tipo bernoulliano.

❸ La risoluzione con il calcolatore

♣ Introduzione a MatLab

Per completare il nostro progetto, abbiamo deciso di automatizzare nostro modello matematico, tramite un piccolo programma d'analisi, fatto con il linguaggio MatLab (file `analisi.m`).

MatLab è un linguaggio interattivo, ossia si "programma" inserendo dei comandi che vengono elaborati immediatamente, e restituendo, dopo un'esecuzione sequenziale, il risultato. Inoltre, tutti i dati vengono trattati come delle matrici (non a caso MatLab sta per *matrix laboratory*).

MatLab fornisce inoltre una serie di moduli aggiuntivi, che permettono di risolvere specifici problemi, inerenti diverse aree tematiche, quali: sistemi di controllo, reti neurali, logica fuzzy, wavelets, simulazione e molto altro ancora.

Il sistema MatLab consiste in cinque parti fondamentali:

1. *L'ambiente di sviluppo:* è un insieme di strumenti, molti dei quali forniti di un interfaccia grafica e che permettono, tra le varie utilità, di visualizzare il manuale, fare debug del codice, scrivere il codice in un apposito editor, ect…
2. *La libreria delle funzioni matematiche:* è una vasta collezione di funzioni, che vanno da quelle più semplici: seno, coseno, somma, ect… Fino ad arrivare alle operazioni su matrici, le trasformate di Fourier, le funzioni di Bessel, ect…
3. *Il linguaggio MatLab:* è un linguaggio ad alto livello, che elabora principalmente matrici e vettori. Con esso, si posso creare dai semplici programmi throw-away, alle vere e proprie applicazioni, più complesse.
4. *La grafica:* permette la visualizzazione delle matrici attraverso dei grafici, inoltre, include funzioni per l'elaborazione delle immagini e la creazione delle interfacce utente.
5. *L'API (Application Program Interface):* è una libreria di funzioni, che permette di creare programmi in C o Fortran e che interagiscono con gli script .M (estensione per gli script di MatLab).

Prima di utilizzare MatLab, si era pensato di adoperare il PROLOG o l'SMV, ma questi due linguaggi, si sono rivelati poco efficaci per l'elaborazione matematica, sebbene fossero molto potenti dal punto di vista della programmazione in linguaggio formale.

♥ Il codice del programma d'analisi

Per rendere più facile il commento e la lettura del programma d'analisi, abbiamo scelto di suddividerlo in argomenti diversi, ognuno trattante una sezione del capitolo sulla risoluzione matematica.

I concetti di base

```
% I concetti di base
totale = input('Inserire il numero totale degli eventi nel sistema:
            ');
favorevole = input('Inserire il numero di eventi favorevoli: ');
sfavorevole = input('Inserire il numero di eventi sfavorevoli: ');
incertoT = input('Inserire il numero di eventi incerti: ');

if favorevole + sfavorevole + incertoT == totale
else
    while (favorevole + sfavorevole + incertoT ~= totale)
        disp(' ')
        disp('Errore!!!')
        disp(' ')
        favorevole = input('Reinserire il numero di eventi
            favorevoli: ');
        sfavorevole = input('Reinserire il numero di eventi
            sfavorevoli: ');
        incertoT = input('Reinserire il numero di eventi incerti: ');
    end
end

disp('Gli eventi coincidono')
disp(' ')
disp('La probabilità degli eventi favorevoli nel sistema è: ')
favorevole/totale
disp('La probabilità degli eventi sfavorevoli nel sistema è: ')
sfavorevole/totale
disp('La probabilità degli eventi incerti totali è: ')
incertoT/totale
disp(' ')
disp('Specificare i casi degli eventi incerti');
favorevoleI = input('Inserire il numero di eventi favorevoli di un
                  dato evento incerto: ');
sfavorevoleI = input('Inserire il numero di eventi sfavorevoli di un
                  dato evento incerto: ');

if favorevoleI + sfavorevoleI == incertoT
else
    while favorevoleI + sfavorevoleI ~= incertoT
        disp(' ')
        disp('Errore!!!')
        disp(' ')
        favorevoleI = input('Reinserire il numero di eventi
            favorevoli di un dato evento incerto: ');
        sfavorevoleI = input('Reinserire il numero di eventi
            sfavorevoli di un dato evento incerto: ');
    end
end

disp('Gli eventi coincidono')
disp(' ')
```

In questa prima parte di codice, si possono notare i comandi che permettono l'inserimento dei dati da parte dell'utente (`input`); in particolare, l'utente inserisce il numero totale di eventi che si possono verificare nel sistema, ed il numero di eventi favorevoli, sfavorevoli ed incerti (totali).

Inoltre, si è realizzato un ciclo per la gestione dell’errore, nel caso che gli eventi totali non coincidessero con la somma dei singoli eventi (secondo la definizione della *funzione di probabilità*).

Infine, il programma chiede di specificare, tra gli eventi incerti, il numero di eventi favorevoli e sfavorevoli che si possono verificare.

La probabilità condizionata

```
% La probabilità condizionata
disp('La probabilità condizionata, affinchè l''evento incerto
      favorevole si verifichi è: ')
favorevoleI/incertoT
disp('Mentre la probabilità condizionata, affinchè l''evento incerto
      sfavorevole si verifichi è: ')
sfavorevoleI/incertoT
disp(' ')
```

Successivamente, vengono semplicemente calcolate le probabilità (condizionate), dei singoli eventi favorevoli e sfavorevoli, su tutti gli eventi incerti.

Il teorema di Bernoulli per le prove ripetute

```
% Il teorema di Bernoulli per le prove ripetute
disp('Calcolo della probabilità secondo il teorema di Bernoulli')
disp(' ')
lanci = input('Inserire il numero di lanci che si vogliono eseguire:
            ');
disp('Secondo il teorema di Bernoulli, la probabilità che su n lanci,
      si verifichi un evento favorevole è: ')
bernoulli = (1-(((sfavorevole+incertoT)/totale)^lanci))
disp(' ')
quest = input('Vuoi riprovare con un altro numero di lanci? [s/n]
                  ','s');

while quest == 's'
    if quest == 's'
        lanci = input('Reinserire il numero di lanci che si vogliono
                        eseguire: ');
        bernoulli = (1-(((sfavorevole+incertoT)/totale)^lanci))
        disp(' ')
        quest = input('Vuoi riprovare con un altro numero di lanci?
                        [s/n] ','s');
    else
        break
    end
end
```

Grazie ai dati finora inseriti, il programma, dopo aver ricevuto in input il numero di lanci che si vogliono eseguire, calcola la probabilità d’uscita dell’evento favorevole, secondo il teorema di Bernoulli.

Tale operazione, è ripetibile quante volte si desidera, semplicemente rispondendo in modo affermativo alla domanda.

Il test del χ^2

```
% Il test del chi quadro
disp(' ')
disp('Calcolo del chi quadro')
disp(' ')
f_osservati = input('Inserire il numero di eventi favorevoli
                        osservati: ');
```

```
s_osservati = input('Inserire il numero di eventi sfavorevoli
                        osservati: ');
i_osservati = input('Inserire il numero di eventi incerti osservati:
                        ');

if f_osservati + s_osservati + i_osservati == totale
else
    while (f_osservati + s_osservati + i_osservati ~= totale)
        disp(' ')
        disp('Errore!!!')
        disp(' ')
        f_osservati = input('Reinserire il numero di eventi
                              favorevoli osservati: ');
        s_osservati = input('Reinserire il numero di eventi
                              sfavorevoli osservati: ');
        i_osservati = input('Reinserire il numero di eventi incerti
                              osservati: ');
    end
end

disp(' ')
disp('Il valore del chi quadro calcolato è: ')
disp(' ')
chi_quadro = (((f_osservati - favorevole)^2)/favorevole) +
                  (((s_osservati - sfavorevole)^2)/sfavorevole) +
                  (((i_osservati - incertoT)^2)/incertoT)
quest = input('Vuoi provare con un''altra serie di eventi osservati?
                  [s/n] ','s');
disp(' ')

while quest == 's'
    if quest == 's'
        f_osservati = input('Reinserire il numero di eventi
                  favorevoli osservati: ');
        s_osservati = input('Reinserire il numero di eventi
                              sfavorevoli osservati: ');
        i_osservati = input('Reinserire il numero di eventi incerti
                              osservati: ');
        if f_osservati + s_osservati + i_osservati == totale
        else
            while (f_osservati + s_osservati + i_osservati ~= totale)
                disp(' ')
                disp('Errore!!!')
                disp(' ')
                f_osservati = input('Reinserire il numero di eventi
                                        favorevoli osservati: ');
                s_osservati = input('Reinserire il numero di eventi
                                        sfavorevoli osservati: ');
                i_osservati = input('Reinserire il numero di eventi
                                        incerti osservati: ');
            end
        end
        disp(' ')
        disp('Il valore del chi quadro calcolato è: ')
        chi_quadro = (((f_osservati - favorevole)^2)/favorevole) +
                        (((s_osservati - sfavorevole)^2)/sfavorevole)
                        + (((i_osservati - incertoT)^2)/incertoT)
        disp(' ')
        quest = input('Vuoi provare con un''altra serie di eventi
                        osservati? [s/n] ','s');
    else
        break
```

```
    end
end
```

L'ultima parte di codice, serve per il calcolo del parametro χ^2; il programma chiede all'utente di inserire il numero di eventi che si sono osservati durante il lancio dei dadi e, dopo un'opportuna verifica di coerenza con gli eventi totali, calcola il valore del χ^2.

Anche in questo caso, come per il calcolo del teorema di Bernoulli, l'intera operazione è ripetibile, sempre rispondendo alla domanda che viene fatta alla fine.