

Relazione - Programmazione Procedurale  
Progetto per la sessione autunnale 2024

Biagi Alessio  
a.biagi1@campus.uniurb.it  
Matricola: 322735

Mangiola Cristiano  
c.mangiola@campus.uniurb.it  
Matricola: 306880

Università di Urbino "Carlo Bo"  
2023 - 2024

# Chapter 1

## Specifica del problema

Scrivere un programma ANSI C che acquisisce dalla tastiera due anni compresi tra 1900 e 2099, calcola il giorno e il mese in cui cadono il Martedì Grasso per il primo anno e il Giovedì Grasso per il secondo anno in base al calendario gregoriano (non è consentito prelevare le date da tabelle precompilate) e poi per ciascuna delle due date stampa sullo schermo le cifre del giorno e le prime tre lettere maiuscole del mese (ignorare tutte le lettere successive) con caratteri giganti ognuno formato da asterischi e occupante 5 posizioni sia in altezza che in larghezza.

```
*****  *          *  *  *  *****
      *  **          ** **  *  *  *
***** *  *          * * *  *****
      *  *          * * *  *
***** *****      *  *  *  *  *  *
```

## Chapter 2

# Analisi del problema

### 2.1 Dati in ingresso

I dati in ingresso del problema sono rappresentati dai due anni di riferimento indicati dall'utente, che devono essere dei numeri naturali, di quattro cifre e compresi tra il 1900 e il 2099 (estremi compresi).

Siano  $a, b$  gli anni di riferimento:

$$a, b \in \mathbb{N} \wedge 1900 \leq a, b \leq 2099$$

### 2.2 Dati in uscita

I dati in uscita sono i seguenti:

- Per il primo anno indicato, verranno restituite le cifre del giorno e le prime tre lettere del mese corrispondente al Martedì Grasso, secondo il calendario gregoriano, espresse da asterischi.
- Per il secondo anno indicato, verranno restituite le cifre del giorno e le prime tre lettere del mese corrispondente al Giovedì Grasso, secondo il calendario gregoriano, espresse da asterischi.

Entrambe le cifre dei dati in uscita verranno rappresentate da asterischi occupanti 5 posizioni sia in altezza che in larghezza.

### 2.3 Relazioni intercorrenti

#### 2.3.1 Calendario gregoriano

Il calendario gregoriano è un sistema di datazione introdotto da Papa Gregorio XIII, è strutturato in 12 mesi (da 28 a 31 giorni) per un totale di 365 giorni, 5 ore, 49 minuti e 16 secondi. L'anno bisestile cade ogni 4 anni (con un totale di 366 giorni) ma non negli anni secolari (eccezione fatta per quelli divisibili per 400), e vengono eliminati 3 anni bisestili ogni 400. Se un anno è bisestile, febbraio è composto da 29 giorni anziché 28.

#### 2.3.2 Martedì grasso

Il Martedì Grasso è una data mobile che varia di anno in anno e rappresenta l'ultimo giorno del carnevale. Le sue origini risalgono al Medioevo, quando era consuetudine consumare tutti i cibi "grassi" e carnevaleschi prima del periodo di Quaresima, caratterizzato da digiuno e penitenza. Da qui il nome "Martedì Grasso", che riflette l'usanza di consumare cibi ricchi e calorici in previsione dell'austero periodo quaresimale.

### 2.3.3 Giovedì grasso

Il Giovedì Grasso è una data mobile che varia di anno in anno e fa riferimento all'inizio delle festività di carnevale. Tradizionalmente, questo giorno era dedicato alle maschere ed ai divertimenti, con festeggiamenti che si protraevano fino a notte fonda. L'origine del nome "Giovedì Grasso" è simile a quella del Martedì Grasso: si riferisce all'usanza di consumare cibi grassi e carnevaleschi in abbondanza prima del periodo di Quaresima.

### 2.3.4 Pasqua

La Pasqua è una festa mobile, il cui giorno e mese di celebrazione variano a seconda dell'anno. Il principio per determinare la data di Pasqua fu stabilito dal Concilio di Nicea nel 325 d.c., che decretò che essa dovesse cadere la domenica successiva al primo plenilunio di primavera. Per convenzione, la Chiesa cattolica fissa l'equinozio di primavera al 21 marzo, anche se l'equinozio astronomico può variare tra il 19 e 21 marzo. Per questo, la Pasqua cade nel periodo compreso tra il 22 marzo e il 25 aprile. Questa affermazione è supportata dalle seguenti deduzioni:

- Se il primo plenilunio di primavera cade il 21 marzo e questo coincide con un sabato, la Pasqua si celebrerà il giorno successivo, ossia il 22 marzo.
- Se il plenilunio si verifica il 20 marzo, si dovrà attendere un intero ciclo lunare di circa 29 giorni per il plenilunio successivo, che avverrà il 18 aprile. Se questa data coincide con una domenica, la Pasqua sarà celebrata la domenica successiva, cioè il 25 aprile.

### 2.3.5 Metodo Aritmetico di Gauss

Il metodo aritmetico di Gauss è una tecnica matematica sviluppata dal matematico Carl Friedrich Gauss nel XIX secolo per calcolare la data della Pasqua in un dato anno. Poiché la Pasqua è una data fondamentale nel calendario cristiano, questo metodo permette anche di determinare altre date religiose.

- Preciso, in quanto fornisce la data esatta della Pasqua per un ampio intervallo di anni.
- Universale, poiché funziona per qualsiasi anno del calendario gregoriano.

## Chapter 3

# Progettazione dell'algoritmo

### 3.1 Scelte di progetto

Per determinare la data della Pasqua, è stato scelto di adottare il metodo aritmetico sviluppato da Gauss, descritto in dettaglio nella Sezione 3.1.1. Questa scelta è motivata dalla comprovata efficacia del metodo, ampiamente validato nel corso del tempo, che ne conferma l'accuratezza e l'affidabilità. Inoltre, il metodo è perfettamente compatibile con il calendario gregoriano, rendendolo una soluzione appropriata per questo contesto.”

#### 3.1.1 Calcolo della Pasqua con il metodo di Gauss

Il metodo di Gauss per determinare la data di Pasqua si basa su due parametri  $M$  e  $N$ , che per il calendario gregoriano, indicato successivamente con  $T$ , dal 1900 al 2099 sono precalcolati e seguono le seguenti regole:

$$T = \text{gregoriano} \cup A \in \{x \in \mathbb{N} \mid 1900 \leq x \leq 2099\} \Rightarrow (M = 24, N = 5)$$

L'algoritmo utilizza inoltre alcune variabili intermedie  $(a, b, c, d, e)$  per semplificare il calcolo. Di seguito viene presentata una descrizione dettagliata dell'algoritmo:

1. Si calcolano i valori di  $a, b$  e  $c$  in base all'anno di riferimento  $A$ :

$$A \equiv a \pmod{19}$$

$$A \equiv b \pmod{4}$$

$$A \equiv c \pmod{7}$$

2. Utilizzando  $a, b, c, M$  e  $N$ , si calcolano i valori di  $d$  ed  $e$ :

$$(19a + M) \equiv d \pmod{30}$$

$$(2b + 4c + 6d + N) \equiv e \pmod{7}$$

- Se  $(d + e) < 10$ , allora la Pasqua cade il giorno  $(d + e + 22)$  di marzo, altrimenti la Pasqua cade il giorno  $(d + e - 9)$  di aprile.

Inoltre, vengono gestite alcune eccezioni specifiche:

- Se la data risultante è il 26 aprile, allora la Pasqua cade il 19 aprile.
- Se la data risultante è il 25 aprile e contemporaneamente  $d = 28, e = 6$  e  $a > 10$ , allora la Pasqua cade il 18 aprile.

### 3.1.2 Gestione delle variabili e dei parametri del metodo di Gauss

Nell'implementazione dell'algoritmo di Gauss, si osserva che:

- Le variabili  $a, b$  e  $c$  del metodo di Gauss dipendono esclusivamente dall'anno di riferimento fornito in input. Queste variabili non cambiano in base al calcolo della data di Pasqua.
- L'espressione  $2b + 4c$  è anch'essa dipendente solo dall'anno di riferimento. Quindi, è possibile precalcolarla e memorizzarla in una variabile separata, senza doverla ricalcolare per ogni data di Pasqua.
- La variabile  $a$  viene utilizzata non solo nel calcolo della data di Pasqua, ma anche nella gestione di una specifica eccezione. Pertanto, è necessario mantenere accessibile questo valore.
- Per ottimizzare le performance del programma, si sceglie di calcolare  $a$  e  $2b + 4c$  (memorizzato nella variabile  $f$ , all'esterno della logica principale del calcolo della Pasqua. In questo modo, si evita di ricalcolare questi valori ogni volta.
- I parametri  $M = 24$  e  $N = 5$  vengono definiti come costanti simboliche, per rendere il codice più leggibile e manutenibile.

### 3.1.3 Stampa del risultato

Si evidenziano codeste scelte di progetto:

- Per la stampa delle cifre, dato che ci troviamo in una situazione dove la numerazione è a base 10, saranno utilizzati solo i seguenti caratteri numerici: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- Per la stampa dei caratteri dei mesi, dato che sono limitati a Febbraio, Marzo e Aprile, i quali condividono alcune lettere, saranno utilizzati solo i caratteri: A, B, E, F, M, P, R.
- Per la stampa di numeri con un'unica cifra, essi saranno preceduti da uno zero, in modo da mantenere una coerenza nella lunghezza dell'output.
- Lo spazio tra il gruppo di cifre e il gruppo di lettere del mese avrà una larghezza di 7 posizioni e un'altezza di 5 posizioni.
- Lo spazio tra le cifre e tra le lettere avrà una larghezza di 1 posizione e un'altezza di 5 posizioni.

La stampa dei caratteri giganti sarà realizzata utilizzando una tecnica di stampa sequenziale riga per riga, che permette di non memorizzare l'intera rappresentazione del carattere prima di stamparlo, riducendo l'uso di memoria.

## 3.2 Passi dell'algoritmo

I passi dell'algoritmo per la risoluzione del problema sono i seguenti:

1. Acquisire il primo anno.
2. Controllare la validità del valore inserito. Il programma continua a richiedere il dato in input fino a che non riceve un valore valido.
3. Acquisire il secondo anno.
4. Controllare la validità del valore. Il programma continua a richiedere il dato in input continua a richiedere il dato in input fino a che non riceve un valore valido.
5. Calcolare i valori di  $a$  e dell'espressione  $2b + 4c$  del metodo di Gauss per entrambi gli anni.
  - (a) Definizione delle costanti  $M$  e  $N$  per il calendario gregoriano.
  - (b) Calcolo delle variabili  $a, b$  e  $c$  del metodo di Gauss per il primo anno.
  - (c) Calcolo dell'espressione  $2b + 4c$  per il primo anno.
  - (d) Calcolo del giorno e del mese in base al metodo di Gauss per il primo anno .
  - (e) Gestione delle eventuali eccezioni per il primo anno.

- (f) Calcolo delle variabili  $a, b$  e  $c$  del metodo di Gauss per il secondo anno.
  - (g) Calcolo dell'espressione  $2b+4c$  per il secondo anno.
  - (h) Calcolo del giorno e del mese in base al metodo di Gauss per il secondo anno.
  - (i) Gestione delle eventuali eccezioni per il secondo anno.
6. Calcolare la data del martedì grasso per il primo anno, sottraendo 47 giorni dalla data di Pasqua del primo anno.
  7. Calcolare la data del giovedì grasso per il secondo anno, sottraendo 52 giorni dalla data di Pasqua del secondo anno.
  8. Gestire le date quindi la sottrazione dei giorni, tenendo conto dei mesi e degli anni bisestili.
  9. Stampare il risultato per entrambe le date ricavate.
    - (a) Ricavare la prima e la seconda cifra del giorno del martedì grasso.
    - (b) Caricare il contenitore dei dati per la stampa del martedì grasso.
    - (c) Emettere i dati del martedì grasso utilizzando la stampa riga per riga.
    - (d) Ricavare la prima e la seconda cifra del giorno del giovedì grasso.
    - (e) Caricare il contenitore dei dati per la stampa del giovedì grasso.
    - (f) Emettere i dati del giovedì grasso utilizzando la stampa riga per riga.

## Chapter 4

# Implementazione dell'algoritmo

File sorgente martedì\_giovedì\_grasso:

```

/*****
/*   Progetto per la sessione d'esame autunnale 2023 / 2024   */
/*
/*   Autori: Biagi Alessio           Matricola: 322735       */
/*           Mangiola Cristiano       306880                */
*****/

/*****/
/* Inclusionione delle librerie */
/*****/

#include <stdio.h>

/*****/
/* Definizione delle costanti simboliche */
/*****/

#define M_GREGORIANO    24 /* Costante M del metodo di Gauss per il calendario gregoriano */
#define N_GREGORIANO    5  /* Costante N del metodo di Gauss per il calendario gregoriano */

/*****/
/* Dichiarazione delle funzioni */
/*****/
int acquisizione_anno(void);

void calcolo_gauss(int,
                  int *,
                  int *);

void calcolo_pasqua(int,
                  int,
                  int *,
                  int *);

void calcolo_martedì_giovedì_grasso(int,
                                    int,
                                    int,
```



```

        int *,
        int *);

void calcolo_giovedi_grasso(int,
        int,
        int,
        int *,
        int *);

void sottrazione_giorni(int *,
        int *,
        int *,
        int);

int giorni_mese(int,
        int);

int anno_bisestile(int);

void stampa_risultato(int,
        int);

void stampa_riga_carattere(char,
        int);

/*****
/* Definizione delle funzioni */
*****/

/* Definizione della funzione main */
int main(void)
{
    /* Dichiarazione delle variabili locali alla funzione */
    int primo_anno,          /* Input:  anno di riferimento per il calcolo del martedì grasso */
        secondo_anno,       /* Input:  anno di riferimento per il calcolo del giovedì grasso */
        a_1,                /* Lavoro: variabile a del metodo di Gauss per il primo anno */
        a_2,                /* Lavoro: variabile a del metodo di Gauss per il secondo anno */
        f_1,                /* Lavoro: 2b + 4c del metodo di Gauss, per il primo anno */
        f_2,                /* Lavoro: 2b + 4c del metodo di Gauss, per il secondo anno */
        giorno_pasqua_1,    /* Lavoro: contenitore del giorno della Pasqua, per il primo anno */
        giorno_pasqua_2,    /* Lavoro: contenitore del giorno della Pasqua, per il secondo anno */
        mese_pasqua_1,      /* Lavoro: contenitore del mese della Pasqua, per il primo anno */
        mese_pasqua_2,      /* Lavoro: contenitore del mese della Pasqua, per il secondo anno */
        giorno_martedi_grasso, /* Output: giorno in cui cade il martedì grasso per il primo anno */
        giorno_giovedi_grasso, /* Output: giorno in cui cade il giovedì grasso per il secondo anno */
        mese_martedi_grasso,  /* Output: mese in cui cade il martedì grasso per il primo anno */
        mese_giovedi_grasso; /* Output: mese in cui cade il giovedì grasso per il secondo anno */

    /* Stampare la specifica all'utente */
    printf("Il programma, previo inserimento di due anni tra il 1900 e il 2099,\n"
        "calcola il martedì grasso per il primo anno e il giovedì grasso per il secondo.\n");

    /* Acquisire il primo anno */
    primo_anno = acquisizione_anno();

```

```

/* Acquisire il secondo anno */
secondo_anno = acquisizione_anno();

/* Calcolare i valori di a e dell'espressione  $2b + 4c$  del metodo di Gauss */
/* Per il primo anno inserito: */
calcolo_gauss(primo_anno,
               &a_1,
               &f_1);

/* Per il secondo anno inserito: */
calcolo_gauss(secondo_anno,
               &a_2,
               &f_2);

/* Calcolare la data della pasqua */
/* Per il primo anno inserito: */
calcolo_pasqua(a_1,
               f_1,
               &giorno_pasqua_1,
               &mese_pasqua_1);

/* Per il secondo anno inserito: */
calcolo_pasqua(a_2,
               f_2,
               &giorno_pasqua_2,
               &mese_pasqua_2);

/* Calcolare la data del martedì grasso */
/* Per il primo anno inserito: */
calcolo_martedi_grasso(giorno_pasqua_1,
                       mese_pasqua_1,
                       primo_anno,
                       &giorno_martedi_grasso,
                       &mese_martedi_grasso);

/* Per il secondo anno inserito: */
calcolo_giovedi_grasso(giorno_pasqua_2,
                       mese_pasqua_2,
                       secondo_anno,
                       &giorno_giovedi_grasso,
                       &mese_giovedi_grasso);

/* Stampare il risultato del martedì grasso */
printf("\n\nIl martedì grasso nell'anno %d, cade il: \n\n",
       primo_anno);
stampa_risultato(giorno_martedi_grasso,
                 mese_martedi_grasso);

/* Stampare il risultato del giovedì grasso */
printf("\n\nIl giovedì grasso nell'anno %d, cade il: \n\n",
       secondo_anno);
stampa_risultato(giorno_giovedi_grasso,
                 mese_giovedi_grasso);

```

```

    printf("\n\n");
    return (0);
}

/* Definizione della funzione per l'acquisizione dell'anno */
int acquisizione_anno(void)
{
    /* Dichiarazione della variabili locali della funzione */
    int anno_scelto,          /* Output: anno digitato dall'utente */
        esito_lettura,        /* Lavoro: esito della scanf */
        acquisizione_errata;  /* Lavoro: esito complessivo della scanf */

    /* Acquisizione dell'anno */
    do
    {
        printf("\nInserisci un anno compreso tra il 1900 e il 2099: ");

        /* Acquisizione e validazione stretta */
        esito_lettura = scanf("%d",
                               &anno_scelto);

        acquisizione_errata = esito_lettura != 1 || anno_scelto < 1900 || anno_scelto > 2099;

        if(acquisizione_errata)
            printf("\nIl valore inserito non è corretto, inseriscine uno valido!\n");
        while (getchar() != '\n');
    }
    while (acquisizione_errata);
    return(anno_scelto);
}

/* Definizione della funzione per il calcolo di a e di 2b + 4c del metodo di Gauss */
void calcolo_gauss(int anno, /* Input: anno inserito dall'utente */
                   int *a,    /* Output: variabile a del metodo di Gauss */
                   int *f)    /* Output: identificatore per 2b + 4c */
{
    /* Dichiarazione delle variabili locali alla funzione */
    int b, /* Lavoro: variabile b del metodo di Gauss */
        c; /* Lavoro: variabile c del metodo di Gauss */

    /* Calcolare le variabili a, b e c del metodo di Gauss */
    *a = anno % 19;
    b = anno % 4,
    c = anno % 7;

    /* Calcolare l'espressione 2b + 4c */
    *f = (2 * b) + (4 * c);
}

/* Definizione della funzione per il calcolo del giorno e del mese di Pasqua */
void calcolo_pasqua(int a, /* Input: variabile a del metodo di Gauss */
                    int f, /* Input: identificatore di 2b + 4c */
                    int *giorno, /* Output: giorno di Pasqua */

```

```

        int *mese)    /* Output: giorno di Pasqua */
{
    /* Dichiarazione delle variabili locali alla funzione */
    int d,            /* Lavoro: variabile d del metodo di Gauss */
        e,            /* Lavoro: variabile e del metodo di Gauss */
        m = M_GREGORIANO, /* Lavoro: variabile M del metodo di Gauss */
        n = N_GREGORIANO; /* Lavoro: variabile N del metodo di Gauss */

    /* Calcolare giorno e mese, usando il metodo di Gauss */
    d = (19 * a + m) % 30;
    e = (f + 6 * d + n) % 7;
    *giorno = (d + e);

    if(*giorno < 10)
    {
        *giorno += 22;
        *mese = 3;
    }
    else
    {
        *giorno -= 9;
        *mese = 4;
    }

    /* Gestione delle eccezioni */
    if (*mese == 4)
    {
        if (*giorno == 26)
            *giorno = 19;

        if (*giorno == 25 && d == 28 && e == 6 && a > 10)
            *giorno = 18;
    }
}

/* Definizione della funzione per il calcolo del martedì grasso per il primo anno inserito. */
void calcolo_martedi_grasso(int giorno_pasqua, /* Input: giorno in cui cade la Pasqua */
                           int mese_pasqua,    /* Input: mese in cui cade la Pasqua */
                           int primo_anno,     /* Input: primo anno inserito dall'utente */
                           int *giorno_martedi_grasso, /* Output: giorno in cui cade il martedì grasso */
                           int *mese_martedi_grasso) /* Output: mese in cui cade il martedì grasso */
{
    *giorno_martedi_grasso = giorno_pasqua;
    *mese_martedi_grasso = mese_pasqua;

    sottrazione_giorni(giorno_martedi_grasso,
                       mese_martedi_grasso,
                       &primo_anno,
                       47);
}

/* Definizione della funzione per il calcolo del giovedì grasso per il secondo anno inserito. */
void calcolo_giovedi_grasso(int giorno_pasqua, /* Input: giorno in cui cade la Pasqua */
                            int mese_pasqua,    /* Input: mese in cui cade la Pasqua */

```

```

        int secondo_anno,          /* Input: secondo anno inserito dall'utente */
        int *giorno_giovedi_grasso, /* Output: giorno in cui cade il giovedì grasso */
        int *mese_giovedi_grasso) /* Output: mese in cui cade il giovedì grasso */
{
    *giorno_giovedi_grasso = giorno_pasqua;
    *mese_giovedi_grasso = mese_pasqua;

    sottrazione_giorni(giorno_giovedi_grasso,
                       mese_giovedi_grasso,
                       &secondo_anno,
                       52);
}

/* Definizione della funzione per sottrarre giorni da una data specificata */
void sottrazione_giorni(int *giorno,          /* Input / output: giorno della data */
                       int *mese,           /* Input / output: mese della data */
                       int *anno,           /* Input / output: anno della data */
                       int giorni_da_sottrarre) /* Input: numero di giorni da sottrarre */
{
    while (giorni_da_sottrarre > 0)
    {
        if (giorni_da_sottrarre >= *giorno)
        {
            giorni_da_sottrarre -= *giorno;
            (*mese)--;

            if (*mese < 1)
            {
                *mese = 12;
                (*anno)--;
            }
            *giorno = giorni_mese(*mese, *anno);
        }
        else
        {
            *giorno -= giorni_da_sottrarre;
            giorni_da_sottrarre = 0;
        }
    }
}

/* Definizione della funzione per ottenere il numero di giorni nel mese specificato */
int giorni_mese(int mese, /* Input: mese di cui calcolare i giorni */
                int anno) /* Input: anno di riferimento (per la gestione dell'anno bisestile) */
{
    switch (mese)
    {
        case 2:
            return anno_bisestile(anno) ? 29 : 28;
        case 4:
            return 30;
        case 3:
            return 31;
    }
}

```

```

        default:
            return 0;
    }
}

/* Definizione della funzione per verificare se un anno è bisestile */
int anno_bisestile(int anno) /* Input: anno da verificare */
{
    return (anno % 4 == 0 && anno % 100 != 0) || (anno % 400 == 0);
}

/* Definizione della funzione per la stampa del risultato */
void stampa_risultato(int giorno, /* Input: giorno da stampare */
                     int mese) /* Input: mese da stampare */
{
    /* Dichiarazione delle variabili locali alla funzione */
    char contenitore[6]; /* Lavoro: contenitore dei dati per la stampa */
    int  prima_cifra,    /* Lavoro: prima cifra da sinistra del giorno */
        seconda_cifra, /* Lavoro: seconda cifra da sinistra del giorno */
        v,              /* Lavoro: contatore per asse verticale */
        o;              /* Lavoro: contatore per avanzamento orizzontale */

    /* Ricavare la prima e la seconda cifra del giorno */
    prima_cifra = giorno / 10;
    seconda_cifra = giorno % 10;

    /* Caricare il contenitore dei dati per la stampa */
    contenitore[0] = '0' + prima_cifra;
    contenitore[1] = '0' + seconda_cifra;
    contenitore[2] = ' ';

    if (mese == 2)
    {
        contenitore[3] = 'F';
        contenitore[4] = 'E';
        contenitore[5] = 'B';
    }
    else if (mese == 3)
    {
        contenitore[3] = 'M';
        contenitore[4] = 'A';
        contenitore[5] = 'R';
    }
    else
    {
        contenitore[3] = 'A';
        contenitore[4] = 'P';
        contenitore[5] = 'R';
    }

    /* Emissione dei dati utilizzando la stampa riga per riga */
    for (v = 0; v < 5; v++)
    {

```

```

    for (o = 0; o < 6; o++)
        stampa_riga_carattere(contenitore[o], v);

    printf("\n");
}
}

/* Definizione della funzione per la stampa del carattere */
void stampa_riga_carattere(char carattere, /* Input: carattere da stampare */
                           int v)        /* Input: indicazione posizione asse verticale */
{
    /* Stampare la v-esima riga del carattere */
    switch(carattere)
    {
        case ' ':
            printf("%5s", "");
            break;
        case '0':
            printf(v == 0 || v == 4 ? " ***** " : " *   * ");
            break;
        case '1':
            if (v == 1)
                printf(" ** ");
            else if (v == 2)
                printf(" * * ");
            else if (v == 4)
                printf(" ***** ");
            else
                printf("  * ");
            break;
        case '2':
            if (v == 1)
                printf("   * ");
            else if (v == 3)
                printf(" *   ");
            else
                printf(" ***** ");
            break;
        case '3':
            printf(v == 1 || v == 3 ? "   * " : " ***** ");
            break;
        case '4':
            if (v == 0 ||
                v == 1)
                printf(" *   * ");
            else if (v == 2)
                printf(" ***** ");
            else
                printf("   * ");
            break;
        case '5':
        case '6':
            if (v == 1)
                printf(" *   ");
    }
}

```

```

    else if (v == 3)
        printf(carattere == '5' ? "      * " : " *   * ");
    else
        printf(" ***** ");
    break;
case '7':
    printf(v == 0 ? " ***** " : "      * ");
    break;
case '8':
    printf(v == 0 || v == 2 || v == 4 ? " ***** " : " *   * ");
    break;
case '9':
    if (v == 1)
        printf(" *   * ");
    else if (v == 3)
        printf("      * ");
    else
        printf(" ***** ");
    break;
case 'A':
    if (v == 0)
        printf("      * ");
    else if (v == 1)
        printf(" * * ");
    else if (v == 2)
        printf(" ***** ");
    else
        printf(" *   * ");
    break;
case 'B':
    if (v == 0 || v == 2 || v == 4)
        printf(" ***** ");
    else
        printf(" *   * ");
break;
case 'E':
    if (v == 0 || v == 2 || v == 4)
        printf(" ***** ");
    else
        printf(" *       ");
break;
case 'F':
    if (v == 0 || v == 2)
        printf(" ***** ");
    else
        printf(" *       ");
break;
case 'M':
    if (v == 1)
        printf(" ** ** ");
    else if (v == 2)
        printf(" * * * ");
    else
        printf(" *   * ");

```



```

        break;
case 'P':
    if (v == 0 ||
        v == 2)
        printf(" ***** ");
    else if (v == 1)
        printf(" *   * ");
    else
        printf(" *       ");
    break;
case 'R':
    if (v == 1)
        printf(" *   * ");
    else if (v == 3)
        printf(" * *   ");
    else if (v == 4)
        printf(" *   * ");
    else
        printf(" ***** ");
    break;
}
}

```

## Makefile:

```
# Makefile martedì_giovedì_grasso #

martedì_giovedì_grasso: martedì_giovedì_grasso.c Makefile
    gcc -ansi -Wall -O martedì_giovedì_grasso.c -o martedì_giovedì_grasso

pulisci:
    rm -f martedì_giovedì_grasso.o

pulisci_tutto:
    rm -f martedì_giovedì_grasso martedì_giovedì_grasso.o
```

## Chapter 5

# Testing del programma

I test condotti osservano che il programma accetta solo gli anni compresi nell'intervallo che va dal 1900 al 2099.

Il programma è in grado di gestire gli inserimenti errati o non compresi nell'intervallo, infatti in caso di acquisizione errata, il programma avvisa l'utente e richiede di inserire un valore che sia corretto rispetto ai criteri del programma.”

Test 1:

```
Il programma, previo inserimento di due anni tra il 1900 e il 2099,
calcola il martedì grasso per il primo anno e il giovedì grasso per il secondo.
```

```
Inserisci un anno compreso tra il 1900 e il 2099: 1932
```

```
Inserisci un anno compreso tra il 1900 e il 2099: .
```

```
Il valore inserito non è corretto, inseriscine uno valido!
```

```
Inserisci un anno compreso tra il 1900 e il 2099: ciao
```

```
Il valore inserito non è corretto, inseriscine uno valido!
```

```
Inserisci un anno compreso tra il 1900 e il 2099: 1950
```

```
Il martedì grasso nell'anno 1932, cade il:
```

```
*****  *****  *****  *****  *****
*   *   *   *   *   *   *   *   *   *   *
*   *   *****  *****  *****  *****
*   *   *   *   *   *   *   *   *   *   *
*****  *****  *   *****  *****
```

```
Il giovedì grasso nell'anno 1950, cade il:
```

```
   *   *****  *****  *****  *****
  **   *   *   *   *   *   *   *   *
* *   *****  *****  *****  *****
   *   *   *   *   *   *   *   *
*****  *****  *   *****  *****
```

## Test 2:

Il programma, previo inserimento di due anni tra il 1900 e il 2099, calcola il martedì grasso per il primo anno e il giovedì grasso per il secondo.

Inserisci un anno compreso tra il 1900 e il 2099: ciao

Il valore inserito non è corretto, inseriscine uno valido!

Inserisci un anno compreso tra il 1900 e il 2099: +

Il valore inserito non è corretto, inseriscine uno valido!

Inserisci un anno compreso tra il 1900 e il 2099: 1948

Inserisci un anno compreso tra il 1900 e il 2099: 1976

Il martedì grasso nell'anno 1948, cade il:

```
*      *****      *****      *****      *****
**    *  *          *      *      *  *
*  *  *  *          *****      *****      *****
*      *  *          *      *      *  *
*****      *****      *      *****      *****
```

Il giovedì grasso nell'anno 1976, cade il:

```
*****      *****      *****      *****      *****
*  *          *      *      *  *
*****      *****      *****      *****      *****
*      *  *          *      *      *  *
*****      *****      *      *****      *****
```

## Test 3:

Il programma, previo inserimento di due anni tra il 1900 e il 2099, calcola il martedì grasso per il primo anno e il giovedì grasso per il secondo.

Inserisci un anno compreso tra il 1900 e il 2099: 1905

Inserisci un anno compreso tra il 1900 e il 2099: 1900

Il martedì grasso nell'anno 1905, cade il:

```
*****      *****      *  *  *      *****
*  *          *      **  **      *  *  *  *
*  *          *      *  *  *      *****      *****
*  *          *      *  *  *  *  *  *  *  *
*****      *      *  *  *  *  *  *  *
```

Il giovedì grasso nell'anno 1900, cade il:

```
*****      *****      *****      *****      *****
*      *          *      *      *  *
*****      *****      *****      *****      *****
*      *          *      *      *  *
*****      *****      *      *****      *****
```

#### Test 4:

Il programma, previo inserimento di due anni tra il 1900 e il 2099, calcola il martedì grasso per il primo anno e il giovedì grasso per il secondo.

Inserisci un anno compreso tra il 1900 e il 2099: 1600

Il valore inserito non è corretto, inseriscine uno valido!

Inserisci un anno compreso tra il 1900 e il 2099: 1917

Inserisci un anno compreso tra il 1900 e il 2099: 1924

Il martedì grasso nell'anno 1917, cade il:

```
*****
* * *
*****
* * *
*****
```

Il giovedì grasso nell'anno 1924, cade il:

```
*****
* * *
*****
* * *
*****
```

#### Test 5:

Il programma, previo inserimento di due anni tra il 1900 e il 2099, calcola il martedì grasso per il primo anno e il giovedì grasso per il secondo.

Inserisci un anno compreso tra il 1900 e il 2099: 1957

Inserisci un anno compreso tra il 1900 e il 2099: è

Il valore inserito non è corretto, inseriscine uno valido!

Inserisci un anno compreso tra il 1900 e il 2099: 2000

Il martedì grasso nell'anno 1957, cade il:

```
*****
* * *
* * *****
* * *
*****
```

Il giovedì grasso nell'anno 2000, cade il:

```
*****
* * *
* * *****
* * *
*****
```

### Test 6:

Il programma, previo inserimento di due anni tra il 1900 e il 2099, calcola il martedì grasso per il primo anno e il giovedì grasso per il secondo.

Inserisci un anno compreso tra il 1900 e il 2099: 1969

Inserisci un anno compreso tra il 1900 e il 2099: 2020

Il martedì grasso nell'anno 1969, cade il:

```
*      *****      *****      *****      *****
**     *      *      *      *      *      *      *
* *    *****      *****      *****      *****
*      *      *      *      *      *      *      *
*****      *****      *      *****      *****
```

Il giovedì grasso nell'anno 2020, cade il:

```
*****      *****      *****      *****      *****
*      *      *      *      *      *      *      *
*****      *      *      *****      *****      *****
*      *      *      *      *      *      *      *
*****      *****      *      *****      *****
```

### Test 7:

Il programma, previo inserimento di due anni tra il 1900 e il 2099, calcola il martedì grasso per il primo anno e il giovedì grasso per il secondo.

Inserisci un anno compreso tra il 1900 e il 2099: 1980

Inserisci un anno compreso tra il 1900 e il 2099: 2024

Il martedì grasso nell'anno 1980, cade il:

```
*      *****      *****      *****      *****
**     *      *      *      *      *      *      *
* *    *****      *****      *****      *****
*      *      *      *      *      *      *      *
*****      *****      *      *****      *****
```

Il giovedì grasso nell'anno 2024, cade il:

```
*****      *****      *****      *****      *****
*      *      *      *      *      *      *      *
*      *      *****      *****      *****      *****
*      *      *      *      *      *      *      *
*****      *****      *      *****      *****
```

### Test 8:

Il programma, previo inserimento di due anni tra il 1900 e il 2099, calcola il martedì grasso per il primo anno e il giovedì grasso per il secondo.

Inserisci un anno compreso tra il 1900 e il 2099: 1996

Inserisci un anno compreso tra il 1900 e il 2099: 2052

Il martedì grasso nell'anno 1996, cade il:

```
*****
* * *
***** * *
* * *
*****
```

Il giovedì grasso nell'anno 2052, cade il:

```
*****
* * *
*****
* *
*****
```

### Test 9:

Il programma, previo inserimento di due anni tra il 1900 e il 2099, calcola il martedì grasso per il primo anno e il giovedì grasso per il secondo.

Inserisci un anno compreso tra il 1900 e il 2099: 2003

Inserisci un anno compreso tra il 1900 e il 2099: 2096

Il martedì grasso nell'anno 2003, cade il:

```
***** * *
* * *
* * *****
* * *
*****
```

Il giovedì grasso nell'anno 2096, cade il:

```
*****
* *
*****
* *
*****
```

### Test 10:

Il programma, previo inserimento di due anni tra il 1900 e il 2099, calcola il martedì grasso per il primo anno e il giovedì grasso per il secondo.

Inserisci un anno compreso tra il 1900 e il 2099: 2011

Inserisci un anno compreso tra il 1900 e il 2099: 1916

Il martedì grasso nell'anno 2011, cade il:

```
*****
* * * * *
* * * * *
* * * * *
* * * * *
```

Il giovedì grasso nell'anno 1916, cade il:

```
*****
* * * * *
* * * * *
* * * * *
* * * * *
```

### Test 11:

Il programma, previo inserimento di due anni tra il 1900 e il 2099, calcola il martedì grasso per il primo anno e il giovedì grasso per il secondo.

Inserisci un anno compreso tra il 1900 e il 2099: 2035

Inserisci un anno compreso tra il 1900 e il 2099: 1944

Il martedì grasso nell'anno 2035, cade il:

```
*****
* * * * *
* * * * *
* * * * *
* * * * *
```

Il giovedì grasso nell'anno 1944, cade il:

```
*****
* * * * *
* * * * *
* * * * *
* * * * *
```



## Test 12:

Il programma, previo inserimento di due anni tra il 1900 e il 2099, calcola il martedì grasso per il primo anno e il giovedì grasso per il secondo.

Inserisci un anno compreso tra il 1900 e il 2099: 2072

Inserisci un anno compreso tra il 1900 e il 2099: 1960

Il martedì grasso nell'anno 2072, cade il:

```
*****
*      *
*****
*      *
*****
*      *
```

Il giovedì grasso nell'anno 1960, cade il:

```
*****
*      *
*****
*      *
*****
*      *
```

## Chapter 6

# Verifica del programma

### 6.1 Brano di codice scelto

```
void calcolo_gauss(int anno, int *a, int *f)
{
    int b, c;
    *a = anno % 19;
    b = anno % 4;
    c = anno % 7;
    *f = (2 * b) + (4 * c);
}
```

Il brano del codice scelto può essere riscritto utilizzando variabili brevi ed evitando scritture compatte:

```
*a = anno % 19;
b = anno % 4;
c = anno % 7;
*f = (2 * b) + (4 * c);
```

### 6.2 Proprietà da verificare

La proprietà che si vuole verificare è formalizzata nel seguente modo:

$$(0 \leq a < 19) \cup (0 \leq b < 4) \cup (0 \leq c < 7) \cup (f = 2b + 4c)$$

### 6.3 Svolgimento

Osserviamo che il brano di codice scelto è privo di iterazione e di ricorsione, quindi è possibile utilizzare le triple di Hoare e poi applicare meccanicamente Dijkstra.

Verificare la correttezza del programma  $S$  che vuole calcolare un risultato  $R$  (postcondizione), significa determinare il predicato  $Q$  (precondizione) che risolve l'espressione logica.

$$\{Q\} S \{R\} = \text{vero}$$

La postcondizione  $R$  risulta essere uguale a:

$$(0 \leq a < 19) \cup (0 \leq b < 4) \cup (0 \leq c < 7) \cup (f = 2b + 4c)$$

Il programma  $S$  rappresenta una sequenza di assegnazioni, per questo applicheremo la regola di Dijkstra per le assegnazioni in sequenza:

$$wp(S1; S2, R) = wp(S1, wp(S2, R))$$

Procediamo con le assegnazioni:

1. Per  $a = \text{anno} \% 19$ :

$$\begin{aligned} wp(a = \text{anno} \% 19, (0 \leq a < 19) \cup (0 \leq \text{anno} \% 4 < 4) \cup (0 \leq \text{anno} \% 7 < 7)) \\ = (0 \leq \text{anno} \% 19 < 19) \cup (0 \leq \text{anno} \% 4 < 4) \cup (0 \leq \text{anno} \% 7 < 7) \end{aligned}$$

2. Per  $b = \text{anno} \% 4$ :

$$\begin{aligned} wp(b = \text{anno} \% 4, (0 \leq a < 19) \cup (0 \leq b < 4) \cup (0 \leq \text{anno} \% 7 < 7)) \\ = (0 \leq a < 19) \cup (0 \leq \text{anno} \% 4 < 4) \cup (0 \leq \text{anno} \% 7 < 7) \end{aligned}$$

3. Per  $c = \text{anno} \% 7$ :

$$\begin{aligned} wp(c = \text{anno} \% 7, (0 \leq a < 19) \cup (0 \leq b < 4) \cup (0 \leq c < 7)) \\ = (0 \leq a < 19) \cup (0 \leq b < 4) \cup (0 \leq \text{anno} \% 7 < 7) \end{aligned}$$

4. Per  $f = (2 * b) + (4 * c)$ :

$$\begin{aligned} wp(f = (2 * b) + (4 * c), R) &= R[f / (2 * b) + (4 * c)] \\ &= (0 \leq a < 19) \cup (0 \leq b < 4) \cup (0 \leq c < 7) \cup ((2 * b) + (4 * c) = 2b + 4c) \\ &= (0 \leq a < 19) \cup (0 \leq b < 4) \cup (0 \leq c < 7) \end{aligned}$$

La preconditione più debole del programma è quindi:

$$(0 \leq \text{anno} \% 19 < 19) \cup (0 \leq \text{anno} \% 4 < 4) \cup (0 \leq \text{anno} \% 7 < 7)$$

La quale risulterà sempre vera, per qualsiasi valore "anno" che sia intero, dato che:

- $\text{anno} \% 19$  è sempre compreso tra 0 e 18;
- $\text{anno} \% 4$  è sempre compreso tra 0 e 3;
- $\text{anno} \% 7$  è sempre compreso tra 0 e 6.

Quindi, qualunque sia la preconditione  $Q$ , risulta che la tripla di Hoare  $\{Q\} S \{R\}$  è vera, permettendo al programma di risultare sempre corretto rispetto al problema, indipendentemente dal valore di input di "anno".