

Relazione

Progetto per la sessione invernale
2023/2024

AUTORI

Mattia Gasperoni
matricola: 329235

Giacomo Cercolani
matricola: 328305

Università degli Studi di Urbino Carlo Bo
Insegnamento di Programmazione Procedurale

1 Specifica del Problema

Scrivere un programma ANSI C che acquisisce dalla tastiera un anno compreso tra 1900 e 2099, calcola il giorno e il mese in cui cade il Lunedì dell'Angelo in quell'anno sia secondo il calendario gregoriano che secondo il calendario giuliano (non è consentito prelevare la data da tabelle precompilate) e poi 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.

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

2 Analisi del Problema

2.1 Dati di ingresso del problema

L'unico dato di ingresso del problema è l'anno scelto dall'utente, deve essere un numero naturale di 4 cifre compreso tra il 1900 e il 2099, sono inclusi gli estremi.

2.2 Dati di uscita del problema

I dati di uscita del problema sono:

- Le cifre del giorno e le prime tre lettere del mese in cui cade il Lunedì dell'Angelo secondo il calendario gregoriano nell'anno scelto dall'utente.
- Le cifre del giorno e le prime tre lettere del mese in cui cade il Lunedì dell'Angelo secondo il calendario giuliano nell'anno scelto dall'utente.

2.3 Relazioni intercorrenti tra i dati del problema

2.3.1 Calendario gregoriano e giuliano

La differenza sostanziale tra il calendario giuliano e quello gregoriano consiste nel modo in cui vengono trattati gli anni bisestili¹.

Criteri per la nozione di bisestile nei rispettivi calendari:

- Nel calendario gregoriano un anno è definito bisestile se soddisfa le seguenti condizioni: è divisibile per 4 ma non divisibile per 100 a meno che non sia divisibile anche per 400
- Nel calendario giuliano un anno è definito bisestile se e solo se l'anno è divisibile per 4

2.3.2 Algoritmo di Gauss

L'algoritmo di Gauss è un metodo efficace per determinare la data della Pasqua, sia per il calendario gregoriano che per quello giuliano. A seconda dell'anno specifico, ci saranno vari parametri che differiscono tra i due calendari e che saranno utilizzati nei calcoli per stabilire la data della Pasqua.

2.3.3 Pasqua e Lunedì dell'Angelo

La Pasqua è una festività mobile, il che significa che non cade a una data fissa, infatti la sua data viene calcolata sulla base di un calendario lunisolare² e varia di anno in anno secondo i cicli lunari. Più precisamente, la Pasqua cade la domenica successiva al primo plenilunio della stagione primaverile.

La chiesa Cristiana ha stabilito convenzionalmente che l'equinozio di primavera cade sempre il 21 marzo. La data della Pasqua viene calcolata a partire dall'inizio dell'anno religioso ebraico, ovvero il giorno della prima luna nuova dopo l'equinozio di primavera. Un monaco cristiano ha calcolato che la data della Pasqua è sempre compresa tra il 22 marzo e il 25 aprile.

Il Lunedì dell'Angelo, noto anche come Lunedì di Pasqua cade sempre il giorno successivo alla Domenica di Pasqua.

Come spiegato nella sezione precedente, dato che la Pasqua cade in un periodo limitato di tempo, anche il Lunedì dell'Angelo può cadere solo in un intervallo di tempo specifico, ovvero tra il 23 marzo e il 26 aprile.

¹Un anno bisestile è un anno solare che ha un giorno aggiuntivo.

²Anno solare a più fasi lunari

3 Progettazione dell'Algoritmo

3.1 Scelte di Progetto

Per il calcolo della data della Pasqua, si utilizzerà l'algoritmo sviluppato dal matematico tedesco Carl Friedrich Gauss. La scelta di utilizzare tale algoritmo è motivata dalla sua attendibilità e precisione, caratteristiche che lo hanno reso molto popolare e ampiamente utilizzato da diversi siti di informazione.

È importante sottolineare che l'algoritmo di Gauss è compatibile sia con il calendario gregoriano, attualmente in uso nella maggior parte del mondo, sia con il calendario giuliano, utilizzato in alcune Chiese ortodosse. Questa compatibilità rende l'algoritmo di Gauss uno strumento estremamente versatile per il calcolo della data della Pasqua.

3.1.1 Suddivisione delle funzioni

Per aumentare la leggibilità il più possibile, abbiamo deciso di implementare quattro funzioni distinte. Ogni funzione ha un identificatore intuitivo che indica immediatamente il suo scopo.

Abbiamo riconosciuto la necessità di implementare una funzione per la stampa della data, dove per “data” intendiamo le cifre e le prime tre lettere del mese in cui cade il Lunedì dell'Angelo. Questa funzione è richiesta per la stampa di entrambi i calendari. Inoltre, all'interno di tale funzione è presente anche il calcolo del Lunedì dell'Angelo, comune per entrambi i calendari.

3.1.2 Algoritmo di Gauss per il Calcolo della Pasqua

In questa sezione, discuteremo del metodo aritmetico sviluppato dal matematico Carl Friedrich Gauss intorno al 1800 per il calcolo della data di Pasqua. Iniziamo definendo alcuni parametri, X , Y e l'anno per il quale vogliamo determinare la data di Pasqua N . Questi parametri saranno fondamentali nel processo di calcolo.

Nel calendario *gregoriano*:

$$1900 \leq N \leq 2099 \rightarrow (X = 24, Y = 5)$$

Nel calendario *giuliano*:

$$N \geq 1582 \rightarrow (X = 15, Y = 6)$$

Successivamente si eseguono i seguente calcoli:

$$a = N \% 19$$

$$b = N \% 4$$

$$c = N \% 7$$

$$d = (19a + X) \% 30$$

$$e = (2b + 4c + 6d + Y) \% 7$$

* % da come risultato il resto della divisione

Possiamo infine dire che se $(d + e) < 10$ la Pasqua cadrà il giorno $(d + e + 22)$ di marzo, altrimenti la Pasqua cadrà il giorno $(d + e - 9)$ del mese di aprile.

3.1.3 Stampa a Caratteri Giganti

In conformità con le specifiche del problema, abbiamo implementato la stampa a caratteri giganti. Abbiamo preso le seguenti decisioni:

- Per la stampa dei giorni, utilizziamo i numeri da 0 a 9, abbiamo creato un pattern³ per ciascuno di essi.
- Per la stampa del giorno con una singola cifra, verrà semplicemente stampato il numero stesso. Mentre per i giorni che hanno due cifre si calcola la prima e la seconda cifra e poi vengono stampate.
- Per la stampa dei primi tre caratteri del mese, abbiamo notato che i mesi di nostro interesse sono solo marzo e aprile. Pertanto, abbiamo creato i pattern delle rispettive prime tre lettere del mese.

³disposizione di asterischi che generano un motivo grafico.

- Le cifre del giorno e le prime tre lettere del mese, come da specifica, sono formate da asterischi e occupano 5 posizioni sia in altezza che in larghezza. Lo spazio tra la prima cifra e la seconda cifra del giorno equivale a 1, lo stesso vale per lo spazio tra le prime tre lettere del mese. Mentre lo spazio tra l'ultima cifra del giorno e la prima lettera del mese è di 5 posizioni.

3.2 Passi dell'Algoritmo

I passi dell'algoritmo per risolvere il problema sono i seguenti:

- Acquisizione e validazione dell'anno
- Calcolo del giorno e del mese di Pasqua secondo il calendario gregoriano
- Calcolo e stampa del lunedì dell'Angelo secondo il calendario gregoriano
 - Calcolo del giorno e del mese del lunedì dell'Angelo
 - * Verifica che il lunedì dell'Angelo cada in un giorno plausibile
 - Stampare a caratteri giganti il lunedì dell'Angelo
 - * verificare se il giorno ha 1 o 2 cifre
 - * stampa effettiva riga per riga degli asterischi
- Calcolo del giorno e del mese di Pasqua secondo il calendario giuliano
- Calcolo e stampa del lunedì dell'Angelo secondo il calendario giuliano
 - Calcolo del giorno e del mese del lunedì dell'Angelo
 - * Verifica che il lunedì dell'Angelo cada in un giorno plausibile
 - Stampare a caratteri giganti il lunedì dell'Angelo
 - * verificare se il giorno ha 1 o 2 cifre
 - * stampa effettiva riga per riga degli asterischi

Per evitare la ridondanza del codice nella stampa delle cifre del giorno e dei primi tre caratteri del mese, e nella verifica della correttezza del giorno del Lunedì dell'Angelo, abbiamo ritenuto necessario sviluppare una funzione dedicata. Questa funzione è stata progettata per essere riutilizzata due volte, ottimizzando così l'efficienza del nostro codice.

4 Implementazione dell'Algoritmo

File sorgente calcolo_lunedì_angelo.c:

```
/* **** */
/* Programma per il calcolo della data del Lunedì dell'Angelo */
/* **** */

/* **** */
/* inclusione delle librerie */
/* **** */

#include <stdio.h>
#include <string.h>

/* **** */
/* dichiarazione delle funzioni */
/* **** */

int acquisizione_anno(void);

void calcolo_pasqua_gregoriano(int,
                               char*);

void calcolo_pasqua_giuliano(int,
                              char*);

void calcolo_stampa_lunedì(int,
                           int);

/* **** */
/* definizione delle funzioni */
/* **** */

/*definizione della funzione main*/
int main()
{
    int anno; /*input: anno scelto dall'utente*/

    /*ottenimento e validazione stretta dell'anno*/
    anno = acquisizione_anno();

    /*calcolo della Pasqua nel calendario gregoriano*/
    calcolo_pasqua_gregoriano(anno,
                              "gregoriano");

    /*calcolo della Pasqua nel calendario giuliano*/
    calcolo_pasqua_giuliano(anno,
                            "giuliano");

    return (0);
}

/*definizione della funzione per acquisire e
validare un anno nel range specificato (1900-2099)*/
int acquisizione_anno(void)
{
    /*dichiarazione delle variabili locali alla funzione */
    int anno, /* output: anno scelto dall'utente nel range specificato*/
        esito_lettura, /* lavoro: esito della scanf */
        acquisizione_errata; /* lavoro: esito complessivo dell'acquisizione */

    /*acquisizione e validazione stretta di un anno */
    do
    {
        printf("\nInserisci un anno compreso tra il 1900 e il 2099: ");
        esito_lettura = scanf("%d",
                              &anno);
        acquisizione_errata = esito_lettura != 1 || anno < 1900 || anno > 2099;
        if (acquisizione_errata)
            printf("\nAnno non accettabile!\n");
        while (getchar() != '\n');
    }
    while (acquisizione_errata);
    return(anno);
}
```

```

/*definizione della funzione che calcola
la data di Pasqua nel calendario gregoriano*/
void calcolo_pasqua_gregoriano(int anno,          /* input: anno scelto dall'utente */
                                char *calendario) /* output: nome calendario */
{
    /*dichiarazione delle variabili locali alla funzione*/
    int a,          /* lavoro: variabile per il calcolo della data di Pasqua*/
        b,          /* lavoro: variabile per il calcolo della data di Pasqua*/
        c,          /* lavoro: variabile per il calcolo della data di Pasqua*/
        d,          /* lavoro: variabile per il calcolo della data di Pasqua*/
        e,          /* lavoro: variabile per il calcolo della data di Pasqua*/
        f,          /* lavoro: variabile per il calcolo della data di Pasqua*/
        g,          /* lavoro: variabile per il calcolo della data di Pasqua*/
        h,          /* lavoro: variabile per il calcolo della data di Pasqua*/
        i,          /* lavoro: variabile per il calcolo della data di Pasqua*/
        k,          /* lavoro: variabile per il calcolo della data di Pasqua*/
        l,          /* lavoro: variabile per il calcolo della data di Pasqua*/
        m,          /* lavoro: variabile per il calcolo della data di Pasqua*/
        mese,       /* output: mese di Pasqua*/
        giorno;     /* output: giorno di Pasqua*/

    /*calcolo del mese e del giorno di Pasqua nel calendario gregoriano*/
    a = anno % 19;
    b = anno / 100;
    c = anno % 100;
    d = b / 4;
    e = b % 4;
    f = (b + 8) / 25;
    g = (b - f + 1) / 3;
    h = (19 * a + b - d - g + 15) % 30;
    i = c / 4;
    k = c % 4;
    l = (32 + 2 * e + 2 * i - h - k) % 7;
    m = (a + 11 * h + 22 * l) / 451;
    mese = (h + l - 7 * m + 114) / 31;
    giorno = ((h + l - 7 * m + 114) % 31) + 1;

    /*stampa del menu per il calendario gregoriano*/
    printf("\nIl lunedì dell'Angelo nel calendario %s cade il:\n",
           calendario);

    /*invocazione della funzione per il calcolo del lunedì
    dell'Angelo e la stampa a caratteri giganti*/
    calcolo_stampa_lunedì(mese,
                           giorno);
}

/*definizione della funzione che calcola
la data di Pasqua nel calendario giuliano*/
void calcolo_pasqua_giuliano(int anno,          /* input: anno scelto dall'utente */
                              char *calendario) /* output: nome calendario */
{
    /*dichiarazione delle variabili locali alla funzione */
    int a,          /* lavoro: variabile per il calcolo della data di Pasqua*/
        b,          /* lavoro: variabile per il calcolo della data di Pasqua*/
        c,          /* lavoro: variabile per il calcolo della data di Pasqua*/
        d,          /* lavoro: variabile per il calcolo della data di Pasqua*/
        e,          /* lavoro: variabile per il calcolo della data di Pasqua*/
        mese,       /* output: mese di Pasqua*/
        giorno;     /* output: giorno di Pasqua*/

    /*calcolo del mese e del giorno di Pasqua nel calendario giuliano*/
    a = anno % 4;
    b = anno % 7;
    c = anno % 19;
    d = (19 * c + 15) % 30;
    e = (2 * a + 4 * b - d + 34) % 7;
    mese = (d + e + 114) / 31;
    giorno = ((d + e + 114) % 31) + 1;

    /*stampa del menu per il calendario giuliano*/
    printf("\nIl lunedì dell'Angelo nel calendario %s cade il: \n",
           calendario);

    /*invocazione della funzione per il calcolo del lunedì

```

```

dell'Angelo e la stampa a caratteri giganti*/
calcolo_stampa_lunedì(mese,
                      giorno);
}

/*definizione della funzione per il calcolo del lunedì
dell'Angelo e la stampa a caratteri giganti*/
void calcolo_stampa_lunedì(int mese, /* output: mese del lunedì dell'Angelo*/
                          int giorno) /* output: giono del lunedì dell'Angelo*/
{
    /*dichiarazione delle variabili locali alla funzione */
    int i, /* lavoro: variabile per lo scorrimento dell'array dei giorni*/
        j, /* lavoro: variabile per lo scorrimento dell'array dei mesi*/
        prima_cifra, /* output: prima cifra del giorno */
        seconda_cifra; /* output: seconda cifra del giorno */

    char *pattern_giorni[10][5] =
    {
        {"*****", " * ", " * ", " * ", "*****"},
        {" * ", " ** ", " * ", " * ", "*****"},
        {"*****", " * ", "*****", " * ", "*****"},
        {"*****", " * ", "*****", " * ", "*****"},
        {" * ", " * ", "*****", " * ", " * "},
        {"*****", " * ", "*****", " * ", "*****"},
        {"*****", " * ", "*****", " * ", "*****"},
        {"*****", " * ", "*****", " * ", "*****"},
        {"*****", " * ", " * ", " * ", " * "},
        {"*****", " * ", "*****", " * ", "*****"},
        {"*****", " * ", "*****", " * ", "*****"}
    }; /* lavoro: stampa gigante per i numeri da 0 a 9*/

    char *pattern_mesi[2][3][5] =
    {
        {
            {" * ", " ** ", " * ", " * ", " * "}, /*m*/
            {" * ", " * ", "*****", " * ", " * "}, /*a*/
            {"*****", " * ", "*****", " * ", " * "}, /*r*/
        },
        {
            {" * ", " * ", "*****", " * ", " * "}, /*a*/
            {"*****", " * ", "*****", " * ", " * "}, /*p*/
            {"*****", " * ", "*****", " * ", " * "}, /*r*/
        }
    }; /* lavoro: stampa gigante per le prime tre lettere dei mesi di marzo e aprile*/

    /*ottenimento mese e giorno del lunedì dell'Angelo,
    verifica se la Pasqua cade l'ultimo di marzo*/
    if (mese == 3)
    {
        if (giorno < 31)
        {
            /*ottenimento lunedì dell'Angelo*/
            mese = 3;
            giorno++;
        }
        else
        {
            /*passaggio al primo di aprile*/
            mese = 4;
            giorno = 1;
        }
    }
    else
    {
        giorno++;
    }

    /*ciclo for per la stampa a caratteri giganti usando il pattern corrispondente*/
    for (i = 0;
         (i < 5);
         ++i)
    {
        /*stampa del giorno se ha una sola cifra*/
        if (giorno <= 9)
        {

```



```

        printf("%s      ",
               pattern_giorni[giorno][i]);
    }
    /*stampa del giorno se ha due cifre*/
    else
    {
        /*calcolo della prima e della seconda cifra del giorno*/
        prima_cifra = giorno / 10;
        seconda_cifra = giorno % 10;

        printf("%s %s      ",
               pattern_giorni[prima_cifra][i],
               pattern_giorni[seconda_cifra][i]);
    }
    /*ciclo for per la stampa delle prime tre lettere del mese*/
    for (j = 0;
         (j < 3);
         ++j)
    {
        printf("%s ",
               pattern_mesi[mese - 3][j][i]);
    }
    /*manda a capo la riga*/
    printf("\n");
}
}
}

```

Makefile:

```
calcolo_lunedì_angelo: calcolo_lunedì_angelo.c Makefile
    gcc -ansi -Wall -O calcolo_lunedì_angelo.c -o calcolo_lunedì_angelo

pulisci:
    rm -f calcolo_lunedì_angelo.o

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

5 Testing del Programma

I test effettuati rivelano che il programma accetta solo date comprese tra il 1900 e il 2099 e se durante l'acquisizione di un'anno l'utente digitata un qualsiasi carattere non numerico o una data non conforme al range preimpostato e preme invio, il programma chiederà di rieffettuare la scelta dell'anno.

Test 1

Inserisci un anno compreso tra il 1900 e il 2099: 1902

Il lunedì dell'Angelo nel calendario gregoriano cade il:

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

Il lunedì dell'Angelo nel calendario giuliano cade il:

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

Test 2

Inserisci un anno compreso tra il 1900 e il 2099: 1925

Il lunedì dell'Angelo nel calendario gregoriano cade il:

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

Il lunedì dell'Angelo nel calendario giuliano cade il:

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

Test 3

Inserisci un anno compreso tra il 1900 e il 2099: 1936

Il lunedì dell'Angelo nel calendario gregoriano cade il:

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

Il lunedì dell'Angelo nel calendario giuliano cade il:

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

Test 4

Inserisci un anno compreso tra il 1900 e il 2099: 2100

Anno non accettabile!

Test 5

Inserisci un anno compreso tra il 1900 e il 2099: 1965

Il lunedì dell'Angelo nel calendario gregoriano cade il:

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

Il lunedì dell'Angelo nel calendario giuliano cade il:

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

Test 6

Inserisci un anno compreso tra il 1900 e il 2099: 1984

Il lunedì dell'Angelo nel calendario gregoriano cade il:

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

Il lunedì dell'Angelo nel calendario giuliano cade il:

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

Test 7

Inserisci un anno compreso tra il 1900 e il 2099: 2012

Il lunedì dell'Angelo nel calendario gregoriano cade il:

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

Il lunedì dell'Angelo nel calendario giuliano cade il:

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

Test 8

Inserisci un anno compreso tra il 1900 e il 2099: 2045

Il lunedì dell'Angelo nel calendario gregoriano cade il:

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

Il lunedì dell'Angelo nel calendario giuliano cade il:

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

Test 9

Inserisci un anno compreso tra il 1900 e il 2099: 2073

Il lunedì dell'Angelo nel calendario gregoriano cade il:

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

Il lunedì dell'Angelo nel calendario giuliano cade il:

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

Test 10

Inserisci un anno compreso tra il 1900 e il 2099: 2094

Il lunedì dell'Angelo nel calendario gregoriano cade il:

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

Il lunedì dell'Angelo nel calendario giuliano cade il:

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

6 Verifica del Programma

6.1 Brano di Codice Scelto

```
if (giorno < 31)
{
    /*ottenimento lunedì dell 'Angelo*/
    mese = 3;
    giorno++;
}
else
{
    /*passaggio al primo di aprile*/
    mese = 4;
    giorno = 1;
}
```

Il brano di codice scelto può essere riscritto utilizzando variabili brevi, ciò ci tornerà utile successivamente durante lo sviluppo della formula logica.

```
if (g < 31)
{
    m = 3;
    g = g + 1;
}
else
{
    m = 4;
    g = 1;
}
```

6.2 Proprietà da Verificare

Osserviamo innanzitutto che il brano di codice è privo di iterazione e di ricorsione, quindi possiamo utilizzare le triple di Hoare e poi applicare le regole di Dijkstra.

Verificare la correttezza del programma S che vuole calcolare un risultato R (postcondizione), vuol dire determinare il predicato Q (precondizione) che risolve l'equazione logica:

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

La postcondizione è formalizzata come segue:

$$\{R\} = (m = 3 \vee (m = 4 \wedge g \geq 1))$$

6.3 Svolgimento

Il programma S è un'istruzione di selezione "if (β) S_1 else S_2 ", quindi andiamo ad applicare la relativa regola di Dijkstra per il calcolo della precondizione più debole:

$$wp(S, R) = ((\beta \implies wp(S_1, R)) \vee (\neg\beta \implies wp(S_2, R)))$$

Il predicato β equivale a $g < 31$, mentre la sua negazione $\neg\beta$ equivale a $g \geq 31$

Il primo gruppo di istruzioni S_1 è composto da due istruzioni di assegnamento perciò lo tratteremo utilizzando questa formula:

$$wp(S_1, R) = wp(S_3, wp(S_4, R))$$

Anche il secondo gruppo di istruzioni S_2 è composto da due istruzioni di assegnamento perciò lo tratteremo utilizzando questa formula:

$$wp(S_2, R) = wp(S_5, wp(S_6, R))$$

Applichiamo ora le regole di Dijkstra:

$wp(S_1, R)$:

$$wp(S_3, wp(S_4, R)) = (R_{m,3})_{g,g+1}$$

$$wp(S_3, wp(S_4, R)) = ((m = 3 \vee (m = 4 \wedge g \geq 1))_{m,3})_{g,g+1}$$

$$wp(S_3, wp(S_4, R)) = (3 = 3 \vee (3 = 4 \wedge g \geq 1))_{g,g+1}$$

$$wp(S_3, wp(S_4, R)) = (3 = 3 \vee (3 = 4 \wedge g + 1 \geq 1))$$

$$wp(S_3, wp(S_4, R)) = (vero \vee (falso \wedge g + 1 \geq 1))$$

$$wp(S_3, wp(S_4, R)) = (vero \vee falso)$$

$$wp(S_3, wp(S_4, R)) = vero$$

$wp(S_2, R)$:

$$wp(S_5, wp(S_6, R)) = (R_{m,4})_{g,g=1}$$

$$wp(S_5, wp(S_6, R)) = ((m = 3 \vee (m = 4 \wedge g \geq 1))_{m,4})_{g,g=1}$$

$$wp(S_5, wp(S_6, R)) = (4 = 3 \vee (4 = 4 \wedge g \geq 1))_{g,g=1}$$

$$wp(S_5, wp(S_6, R)) = (4 = 3 \vee (4 = 4 \wedge 1 \geq 1))$$

$$wp(S_5, wp(S_6, R)) = (4 = 3 \vee (4 = 4 \wedge 1 \geq 1))$$

$$wp(S_5, wp(S_6, R)) = (falso \vee (4 = 4 \wedge 1 \geq 1))$$

$$wp(S_5, wp(S_6, R)) = (falso \vee vero)$$

$$wp(S_5, wp(S_6, R)) = vero$$

Di seguito calcoliamo $wp(S, R)$ sostituendo opportunamente β , $\neg\beta$, $wp(S_1, R)$ e $wp(S_2, R)$:

$$wp(S, R) = ((\beta \implies wp(S_1, R)) \wedge (\neg\beta \implies wp(S_2, R)))$$

$$wp(S, R) = ((g < 31 \implies vero) \wedge (g \geq 31 \implies vero))$$

$$wp(S, R) = (vero \wedge vero)$$

$$wp(S, R) = vero$$

La preconditione più debole del programma è quindi sempre vera. Perciò, qualunque sia la preconditione Q, risulta che la tripla di Hoare $\{Q\} S \{R\}$ è vera. Il programma è quindi sempre corretto rispetto al problema.