

## **RELAZIONE**

Progetto per la sessione d'esame invernale 2022 / 2023

### **AUTORI**

Papadopol Lucian Ioan

*matricola 320648*

Spaccamiglio Luca

*matricola 322270*

Università degli Studi di Urbino Carlo Bo  
Insegnamento di Programmazione Procedurale

## 1. Specifica del Problema

La congettura di Beal asserisce che se  $a^x + b^y = c^z$  dove  $a, b, c, x, y, z \in \mathbb{N}$  con  $a, b, c \geq 1$  e  $x, y, z \geq 3$ , allora  $a, b, c$  hanno un fattore primo in comune.

La congettura di Collatz asserisce che la funzione  $f: \mathbb{N}_{>0} \rightarrow \mathbb{N}_{>0}$  definita ponendo  $f(n) = n / 2$  se  $n$  è pari ed  $f(n) = 3 \cdot n + 1$  se  $n$  è dispari genera 1 dopo un numero finito di applicazioni ai numeri man mano ottenuti.

La congettura di Cramér asserisce che il valore assoluto della differenza tra due numeri primi consecutivi  $\geq 11$  è minore del quadrato del logaritmo naturale del più piccolo dei due numeri.

Scrivere un programma ANSI C che chiede all'utente quale congettura intende considerare e poi la verifica acquisendo dalla tastiera  $a, b, c, x, y, z$  nel primo caso (se non vale  $a^x + b^y = c^z$ , il programma lo stampa sullo schermo e poi verifica comunque se  $a, b, c$  hanno un fattore primo in comune e ne stampa l'esito sullo schermo),  $n > 0$  nel secondo caso (il programma stampa sullo schermo tutti i numeri generati), due numeri primi consecutivi  $\geq 11$  nel terzo caso (il programma stampa sullo schermo sia il valore assoluto della differenza tra i due numeri che il quadrato del logaritmo naturale del più piccolo dei due numeri).

## 2. Analisi del Problema

### 2.0 Considerazioni iniziali

I dati di ingresso e in uscita del problema si differenziano in base alla congettura scelta dall'utente fra le tre previste dalla specifica del problema.

### 2.1 Dati di Ingresso del Problema

- Nella congettura di Beal i dati di ingresso sono rappresentati dai parametri dell'equazione  $a^x + b^y = c^z$  dove  $a, b, c, x, y, z \in \mathbb{N}$  con  $a, b, c \geq 1$  e  $x, y, z \geq 3$ .
- Nella congettura di Collatz l'unico dato in ingresso è rappresentato dal parametro  $n$ ,  $n \in \mathbb{N}$  con  $n > 0$ .
- Nella congettura di Cramér i dati in ingresso sono rappresentati da due numeri primi  $\{P_n, P_{n+1}\} \in \mathbb{P}$  dove  $\mathbb{P}$  è l'insieme dei numeri primi,  $P_{n+1}$  è il numero primo consecutivo a  $P_n$ ,  $P_n \geq 11$ .

### 2.2 Dati di Uscita del Problema

- Nella congettura di Beal l'equazione  $a^x + b^y = c^z$  può essere verificata o meno. In entrambi i casi i dati di uscita del problema sono rappresentati dallo stato dell'equazione che può essere verificata oppure non verificata e dall'eventuale presenza o assenza di fattori primi in comune tra  $a, b, c$ .
- Nella congettura di Collatz i dati di uscita del problema sono tutti i numeri  $n \in \mathbb{N}$  generati dall'esecuzione dell'algoritmo che implementa la congettura stessa.
- Nella congettura di Cramér  $|P_n - P_{n+1}| < (\ln P_n)^2$  i dati di uscita del problema sono rappresentati da: il valore assoluto della differenza tra due numeri primi consecutivi  $|P_n - P_{n+1}|$ , il quadrato del logaritmo naturale del più piccolo dei due numeri  $(\ln P_n)^2$  e la condizione della congettura di essere verificata oppure non verificata per i parametri di input.

### 2.3 Relazioni Intercorrenti tra i Dati del Problema

Il software sviluppato ha lo scopo di gestire tre problemi differenti; perciò, fra diversi problemi non intercorrono relazioni tra dati di input e output.

Analizziamo quindi le relazioni intercorrenti tra i dati di input e output singolarmente per ciascuna congettura.

- Nella congettura di Beal l'equazione è di tipo a coefficienti interi in quanto vi figurano solo parametri appartenenti all'insieme dei numeri naturali. L'equazione è verificata quando vi è l'uguaglianza matematica tra le due espressioni che compongono l'equazione.
- Nella congettura di Collatz il numero in ingresso deve essere  $n \in \mathbb{N}$ ,  $n > 0$ . Applicando la funzione presente nella congettura stessa otterremo sempre una sequenza di numeri naturali e positivi superiori a zero.
- Nella congettura di Cramér i numeri  $\{P_n, P_{n+1}\} \in \mathbb{P}$  dove  $\mathbb{P}$  è l'insieme dei numeri primi. Un numero primo è un numero  $n \in \mathbb{N}$ ,  $n > 1$  che è divisibile solo per 1 e per sé stesso.

### 3. Progettazione dell'algoritmo

#### 3.0 Considerazioni iniziali

Nella progettazione dell'algoritmo ci si è avvalso di una metodologia di progettazione di tipo "Top Down":

- Si formula inizialmente una visione generale dell'algoritmo e se ne descrive la finalità senza scendere nel dettaglio delle sue parti.
- Si suddivide l'algoritmo in parti più piccole ciascuna rispondente ad una funzionalità specifica dall'algoritmo complessivo.
- Ogni parte dell'algoritmo è successivamente rifinita aggiungendo maggiori dettagli.
- Ogni nuova parte così ottenuta può essere ulteriormente definita specificando ulteriori dettagli oppure suddivisa in caso di necessità in parti funzionali più piccole.

L'approccio "Top Down" può essere visto anche come la trasposizione informatica della locuzione latina "Divide et impera" che è l'espedito favorito da ogni tirannide per controllare e governare un popolo, nel nostro caso l'algoritmo, ovvero dividerlo in piccole parti facilmente gestibili.

Questa metodologia di progettazione ben si adatta in generale alla programmazione procedurale poiché trae vantaggio dalla suddivisione del problema in parti funzionali più piccole dette procedure e nello specifico, al linguaggio di programmazione ad alto livello C che è procedurale ed imperativo.

#### 3.1 Scelte di Progetto

##### 3.1.1 Scelte generali

Si è scelto l'utilizzo di un menù per la selezione di una delle tre congetture da verificare e di realizzare una funzione per ciascuna congettura richiamata da esso.

Si è scelto anche di lasciar gestire l'input, inteso come acquisizione dei dati dall'utente e l'output, inteso come stampa a schermo dei dati a ciascuna funzione indipendentemente, inclusa la validazione dei dati in ingresso e l'eventuale gestione di over-flow durante i calcoli. Ogni funzione implementa interamente ed indipendentemente la verifica della relativa congettura, ciò ha permesso lo sviluppo del software nel nostro gruppo di lavoro in maniera modulare: le funzioni sono state divise fra i membri del gruppo che hanno provveduto ad implementarle singolarmente per poi integrarle nel software completo una volta validate con estensivi test. Anche la fase di debug finale, svolta collegialmente tra i membri del gruppo ne ha tratto vantaggio per la facilità nel circoscrivere gli errori ed apportare modifiche correttive senza rischio di creare nuovi "bug" involontari nelle altre funzioni già validate.

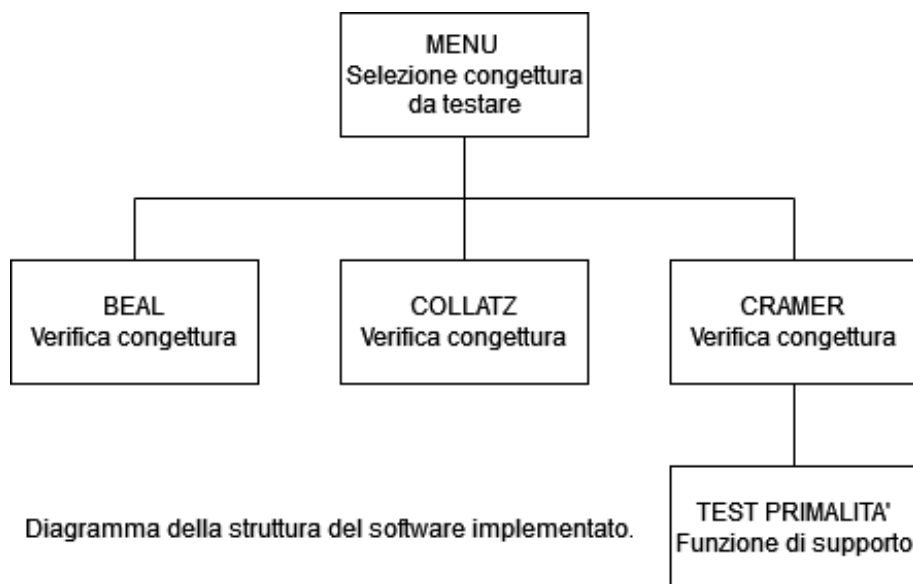


Diagramma della struttura del software implementato.

### 3.1.2 Scelta dei “data type” del linguaggio C da impiegare

Considerando che tutti i valori in ingresso vanno assegnati a parametri  $n \in \mathbb{N}$  per le congetture di Beal e Collatz ed  $n \in \mathbb{P}$ ,  $\mathbb{P} \subset \mathbb{N}$  dove  $\mathbb{P}$  è l'insieme dei numeri primi, per la congettura di Cramér, ma comunque, per tutte e tre le congetture con  $n > 0$  quindi valori non negativi  $\Rightarrow$  le variabili di ingresso delle tre congetture avranno un “data type” del linguaggio C di tipo “unsigned int” che rappresenta ed approssima  $\mathbb{N}$ . La stessa considerazione vale anche per le variabili di output eccetto, dove esplicitamente richiesto l'output di numeri frazionari, di tipo “double”.

Anche le funzioni iterative hanno variabili di controllo con valori interi che non diventano mai negativi e che sono quindi variabili a valori  $n \in \mathbb{N}$ .

“Unsigned Int” rappresenta ed approssima  $\mathbb{N}$ , “Int” rappresenta ed approssima  $\mathbb{Z}$ ,  $\mathbb{N} \subset \mathbb{Z} \Rightarrow$  si sceglie di non utilizzare “unsigned int” in favore di “int” non essendoci una reale necessità di altri “data type” diversi da quelli fondamentali del linguaggio C e si evitano possibili bug causati da operazioni tra numeri “signed” ed “unsigned”.

### 3.1.3 Scelta dei metodi di validazione dei dati in input e dei metodi di gestione overflow

Per la validazione dei dati acquisiti in input si decide di utilizzare la tecnica della validazione stretta del dato acquisito. Ci si avvale del valore restituito dalla funzione “scanf( )”. Tale funzione dopo essere stata chiamata restituisce un valore numerico pari al numero di parametri correttamente acquisiti. Tale valore nel caso dell'acquisizione di un singolo dato, se  $\neq 1$  indica errore, quindi ci permette di agire avvisando l'utente dell'inammissibilità del dato e chiedendo di re-inserirlo. La validazione stretta così implementata ci permette di identificare i casi di valori inseriti fuori dal range ammissibile della variabile a cui va assegnato il dato oppure simboli e lettere digitate al posto di valori numerici. Alcune operazioni matematiche quali moltiplicazione e specialmente elevazione a potenza, possono facilmente causare overflow della variabile alla quale viene assegnato il risultato. Si è quindi ritenuto necessario l'impiego della libreria standard “errno.h” che contiene le macro al preprocessore del compilatore relative ai codici di errore generati dalla funzione “pow( )” per poter gestire tali casi di overflow dell'operazione.

Per quanto riguarda invece l'operazione  $(n * 3) + 1$  presente nella funzione contenuta nella congettura di Collatz, si sceglie di utilizzare una condizione di valore massimo ammissibile per i dati in ingresso, sfruttandola nella fase di validazione del dato in ingresso. Si utilizza la macro “INT\_MAX” presente nella libreria standard “limits.h” per rendere ciò possibile ed indipendente dall'implementazione del valore massimo ammissibile dei data-type del compilatore.

Essendo la condizione  $n < \frac{INT\_MAX}{3} \Rightarrow (n * 3) + 1 \leq INT\_MAX$  evitiamo sempre l'overflow.

### 3.1.4 Criterio di suddivisione in funzioni

Si è scelto di realizzare funzioni a partire da blocchi di codice utilizzati in maniera ricorrente nell'ottica di una scrittura del codice efficiente e dei principi della programmazione procedurale strutturata. Per identificare tali blocchi di codice si è deciso che devono rispettare queste condizioni:

- Essere impiegati più di due volte.
- Avere un significato funzionale proprio indipendente dalle altre funzioni.

Si è identificata la necessità di implementare una singola funzione di supporto alla funzione che implementa la verifica della congettura di Cramér ovvero, una funzione che verifica la primalità di un numero e che viene impiegata sia durante la fase di validazione del dato in ingresso alla congettura che, durante la verifica della consecutività dei due numeri primi inseriti.

## 3.2 Passi dell'Algoritmo

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

### Menù di selezione congettura

- Acquisire dall'utente la scelta della congettura da prendere in considerazione tramite menù.
- In base alla scelta effettuata dall'utente richiamare una delle seguenti funzioni:

#### Congettura di Beal

- Acquisire parametri  $A, B, C, x, y, z$  e controllare che  $A, B, C \geq 1$  e  $x, y, z \geq 3$  e che siano interi e contestualmente calcolare  $A^x, B^y, C^z$ .
- Verificare l'equazione  $A^x + B^y = C^z$  per i parametri calcolati.
- In base al risultato ottenuto:
  - Se l'equazione è verificata:
    - Comunicare l'esito: "Equazione e congettura verificata per i parametri inseriti."
  - oppure:*
    - Se l'equazione non è verificata:
      - Comunicare l'esito: "Equazione e congettura non verificata per i parametri...".
- Verificare se tra  $A, B, C$  vi sono fattori primi in comune.
  - Comunicare l'esito affermativo: "Vi è almeno un fattore primo in comune fra...".
  - Comunicare l'eventuale primo fattore primo in comune trovato.
- oppure:*
  - Comunicare l'esito negativo: "Non vi è alcun fattore primo in comune fra...".

#### Congettura di Collatz

- Acquisire un numero intero  $n > 0$ .
- In base al valore del numero:
  - Se è pari:
    - Calcolare  $n/2$ .
  - oppure:*
    - Se è dispari:
      - Calcolare  $(n * 3) + 1$ .
- Eseguire la procedura sopra riportata fino al raggiungimento di  $n = 1$ .
- Comunicare l'esito: "La congettura è verificata. Ho raggiunto  $n = 1$ "

#### Congettura di Cramér

- Acquisire un numero primo  $P_n \geq 11$  ed il numero primo consecutivo al precedente. Contestualmente verificare che i dati siano numeri primi e che, dato il primo numero primo inserito  $P_n$ , il numero primo consecutivo  $P_{n+1}$  sia effettivamente tale.
- Calcolare e comunicare:
  - Il valore assoluto della differenza tra i due primi.
  - Il quadrato del logaritmo naturale del più piccolo dei due primi.
  - Il rapporto tra la differenza dei due primi ed il quadrato del Logaritmo naturale del minore
- In base al risultato ottenuto:
  - Se l'equazione è verificata:
    - Comunicare l'esito: "La congettura è verificata per i parametri inseriti".
  - oppure:*
    - Se l'equazione non è verificata:
      - Comunicare l'esito: "La congettura non è verificata per i parametri inseriti".

Per evitare ridondanze di codice nel primo passo dell'algoritmo adibito alla verifica della congettura di Cramér, nello specifico nella parte relativa alla verifica di primalità e di consecutività tra numeri primi, è necessario sviluppare un sottoprogramma di supporto che si occupi di verificare la primalità di un numero.

## 4. Implementazione dell'Algoritmo

File sorgente congetture beal collatz cramer.c

```
/* ***** */
/* Esame PP-PPL-PE_PPro-PLPr-CPro 2022/2023 */
/* */
/* Programma per testare le congetture di Beal, Collatz, Cramèr */
/* */
/* Autori: Papadopol Lucian Ioan Matricola: 320648 */
/* Spaccamiglio Luca 322270 */
/* ***** */

/* ***** */
/* Inclusione delle librerie */
/* ***** */

#include <stdio.h>
#include <math.h>
#include <errno.h>
#include <limits.h>

/* ***** */
/* Dichiarazione delle funzioni */
/* ***** */

int congettura_beal(void);
int congettura_collatz(void);
int congettura_cramer(void);
int verifica_nprimo(int);

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

/* definizione della funzione main */
int main(void)
{
    /* dichiarazione delle variabili locali alla funzione */
    int scelta_congettura, /* input: selezione congettura */
        esito_acquisizione, /* lavoro: esito della scanf */
        acquisizione_errata; /* lavoro: esito complessivo acquisizione_errata */

    /* acquisizione scelta */
    do
    {
        /* visualizzazione menù di scelta */
        printf("\n PROGRAMMA DI TEST CONGETTURE DI BEAL - COLLATZ - CRAMÉR\n\n"
            "[1] Beal\n"
            "[2] Collatz\n"
            "[3] Cramér\n\n"
            "Digita il numero della tua scelta e premi 'Invio'. \n\n");

        /* acquisizione scelta e validazione stretta */
        esito_acquisizione = scanf("%d",
            &scelta_congettura);
        acquisizione_errata = esito_acquisizione != 1 ||
            scelta_congettura > 4 ||
            scelta_congettura <= 0;
        if (acquisizione_errata)
            printf("Valore fuori range! \n");
        while (getchar() != '\n');
    } while (acquisizione_errata);

    /* struttura di selezione della congettura */
    switch (scelta_congettura)
    {

```

```

case 1:
    congettura_beal();
    break;
case 2:
    congettura_collatz();
    break;
case 3:
    congettura_cramer();
    break;
}
return (0);
}

/* definizione della funzione per verificare la congettura di Beal */
int congettura_beal(void)
{
    int par_equ[5],          /* input: parametri della equazione*/
        esito_lettura,      /* lavoro: esito della scanf */
        acquisizione_errata, /* lavoro: esito complessivo dell'acquisizione_errata */
        i_primi = 2,        /* lavoro: indice calcolo fattori primi comuni */
        i_parametri,        /* lavoro: indice acquisizione parametri*/
        n_fprimi = 0;       /* output: esito calcolo fattori primi comuni */
    char scelta[6] = {'A', /* output: messaggi per input variabili corrispondenti */
                     'B',
                     'C',
                     'x',
                     'y',
                     'z'};

    double a_esponentex = 1, /* lavoro: risultato potenza A^x */
           b_esponentey = 1, /* lavoro: risultato potenza B^y */
           c_esponentez = 1; /* lavoro: risultato potenza C^z */

    /* messaggio esplicativo dei parametri da inserire*/
    printf("Equazione A^x + B^y = C^z\n\n");

    /* acquisire parametri a,b,c,x,y,z e validazione stretta */
    for (i_parametri = 0;
         i_parametri < 6;
         i_parametri++)
    {
        do
        {
            if (i_parametri <= 2)
                printf("Digita valore letterale %c ≥ 1: ",
                      scelta[i_parametri]);
            else
                printf("Digita esponente %c ≥ 3: ",
                      scelta[i_parametri]);
            esito_lettura = scanf("%d",
                                  &par_equ[i_parametri]);
            if (i_parametri == 0 || i_parametri <= 2)
                acquisizione_errata = esito_lettura != 1 ||
                                      par_equ[i_parametri] < 1;
            else
                acquisizione_errata = esito_lettura != 1 ||
                                      par_equ[i_parametri] < 3;
            if (acquisizione_errata)
                printf("Valore non accettabile! \n");
            else
            {
                if (esito_lettura == 1 &&
                    i_parametri == 3)
                {
                    errno = 0; /* reset errno */
                    a_esponentex = pow(par_equ[0], par_equ[3]);
                    if (errno != 0)
                    {
                        printf("Potenza troppo elevata!\n");
                        acquisizione_errata = 1;
                    }
                }
            }
        }
        while (esito_lettura != 1 ||
              i_parametri < 6 ||
              acquisizione_errata);
    }
}

```



```

    }
}
if (esito_lettura == 1 &&
    i_parametri == 4)
{
    errno = 0; /* reset errno */
    b_esponentey = pow(par_equ[1], par_equ[4]);
    if (errno != 0)
    {
        printf("Potenza troppo elevata!\n");
        acquisizione_errata = 1;
    }
}
if (esito_lettura == 1 &&
    i_parametri == 5)
{
    errno = 0; /* reset errno */
    c_esponentez = pow(par_equ[2], par_equ[5]);
    if (errno != 0)
    {
        printf("Potenza troppo elevata!\n");
        acquisizione_errata = 1;
    }
}
}
while (getchar() != '\n');
} while (acquisizione_errata);
}

/* calcolare quanti primi vi sono in comune tra a, b, c*/
while (par_equ[0] >= i_primi &&
        par_equ[1] >= i_primi &&
        par_equ[2] >= i_primi)
{
    if (par_equ[0] % i_primi == 0 &&
        par_equ[1] % i_primi == 0 &&
        par_equ[2] % i_primi == 0)
    {
        par_equ[0] = par_equ[0] / i_primi;
        par_equ[1] = par_equ[1] / i_primi;
        par_equ[2] = par_equ[2] / i_primi;
        n_fprimi++;
        printf("\nTrovato fattore primo comune: %d",
                i_primi);
    }
    else
        i_primi++;
}

/* avvisare l'utente dell'esito */
if (a_esponentex + b_esponentey == c_esponentez)
{
    printf("\nEquazione e congettura verificata per i parametri inseriti.\n");
}
else
{
    printf("\nEquazione e congettura non verificata per i parametri inseriti.\n");
}

/* indicare all'utente se vi sono fattori primi in comune */
if (n_fprimi > 0)
    printf("Vi è almeno un fattore primo in comune fra A,B e C.\n");
else
    printf("Non vi è alcun fattore primo in comune fra A,B,C.\n");
return (0);
}

/* definizione della funzione per verificare la congettura di Collatz */

```

```

int congettura_collatz(void)
{
    /* dichiarazione delle variabili locali alla funzione */
    int numero_in,          /* input: numero naturale scelto dall'utente */
        esito_lettura,      /* lavoro: esito della scanf */
        acquisizione_errata; /* lavoro: esito complessivo dell'acquisizione_errata */

    /* acquisizione numero da testare e validazione stretta */
    do
    {
        printf("Digita un numero intero > 0: \n");
        esito_lettura = scanf("%d",
                               &numero_in);
        acquisizione_errata = esito_lettura != 1 ||
                               numero_in <= 0 ||
                               numero_in > (INT_MAX / 3);

        if (acquisizione_errata)
            printf("Valore fuori range! \n");
        while (getchar() != '\n');
    } while (acquisizione_errata);

    /* calcolare il valore del numero fino al raggiungimento del valore 1 */
    printf("Numeri generati:\n");
    do
    {
        if (numero_in % 2 == 0)
            numero_in = numero_in / 2;
        else
            numero_in = (numero_in * 3) + 1;

        /* stampare ogni numero ottenuto dai calcoli */
        printf("\n %d", numero_in);
    } while (numero_in != 1);

    /* avvisare l'utente dell'esito */
    printf("\nLa congettura è verificata. Ho raggiunto n = 1\n\n");
    return (0);
}

/* definizione della funzione per verificare la congettura di Cramér */
int congettura_cramer(void)
{
    int valori[] = {0, 0}, /* input: numeri primo in ingresso */
        esito_lettura,     /* lavoro: esito della scanf */
        acquisizione_errata, /* lavoro: esito complessivo dell'acquisizione_errata */
        i_primi,           /* lavoro: indice acquisizione numeri primi */
        i_ricercanp,       /* lavoro: indice ricerca primi tra due estremi */
        a_esito,           /* lavoro: esito verifica numero primo, ciclo di validazione input */
        b_esito = 0,       /* lavoro: conteggio numeri primi */
        delta_np;          /* output: differenza tra i due numeri primi */
    double log_nppn,       /* output: logaritmo del quadrato del numero primo minore */
        rapporto_np;      /* output: rapporto fra delta numeri primi e log quadrato np min */

    printf("\nDigita un numero primo ≥ 11 e premere 'Invio'. \n"
           "Successivamente digita un altro numero primo consecutivo al precedente.\n\n");

    for (i_primi = 0;
         i_primi < 2;
         i_primi++)
    {
        do
        {
            esito_lettura = scanf("%d",
                                   &valori[i_primi]);
            acquisizione_errata = esito_lettura != 1 ||
                                   valori[i_primi] < 11;

            if (acquisizione_errata)
                printf("Valore fuori range! \n");
        }
    }
}

```

```

else
{
    a_esito = verifica_nprimo(valori[i_primi]);

    if (a_esito == 0)
    {
        acquisizione_errata = 1;
        printf("Non è un numero primo! \n");
    }

    if (a_esito == 1 &&
        i_primi == 1 &&
        valori[1] > valori[0])
    {
        i_ricercanp = valori[0] + 1;
        do
        {
            b_esito += verifica_nprimo(i_ricercanp);
            i_ricercanp++;
        } while (i_ricercanp < valori[1]);

        if (b_esito != 0)
        {
            printf("Non è un numero primo consecutivo al primo.\n");
            b_esito = 0;
            acquisizione_errata = 1;
        }
    }

    if (a_esito == 1 &&
        i_primi == 1 &&
        (valori[1] < valori[0] || valori[1] == valori[0]))
    {
        printf("Hai inserito due primi identici oppure il secondo più piccolo\n");
        acquisizione_errata = 1;
    }
}
while (getchar() != '\n');
} while (acquisizione_errata);
}

/* indicare all'utente l'esito dei vari calcoli */
delta_np = valori[1] - valori[0];
printf("\nDifferenza tra i due numeri primi: %d\n",
    delta_np);

log_nnpn = pow(log(valori[0]), 2);
printf("Quadrato del Logaritmo naturale del primo minore: %0.2lf\n",
    log_nnpn);

rapporto_np = delta_np / log_nnpn;
printf("Rapporto tra la differenza dei due primi ed il quadrato del Logaritmo naturale del mi-
nore: %0.2lf\n",
    rapporto_np);

/* indicare all'utente se la congettura è verificata per i parametri inseriti */
if (rapporto_np <= 1)
    printf("La congettura è verificata per i parametri inseriti\n\n");
else
    printf("La congettura non è verificata per i parametri inseriti\n\n");

return (0);
}

/* definizione della funzione per la verifica se un numero è primo */
int verifica_nprimo(int numero_in) /* input: valore da verificare */
{
    /* dichiarazione delle variabili locali alla funzione */
    int i, /* lavoro: controllo ciclo */

```

```
    risultato = 1; /* output: variabile risultato della verifica */

/* ciclo di verifica se un primo è effettivamente tale */
for (i = 2; i < (int)sqrt(numero_in); i++)
{
    if ((numero_in % i) == 0)
    {
        risultato = 0; /* non è primo*/
    }
}

return risultato;
}
```

## Makefile:

```
# Makefile congetture_beal_collatz_cramer #

congetture_beal_collatz_cramer: congetture_beal_collatz_cramer.c Makefile
    gcc -ansi -Wall -O congetture_beal_collatz_cramer.c
    -o congetture_beal_collatz_cramer -lm

pulisci:
    rm -f congetture_beal_collatz_cramer.o

pulisci_tutto:
    rm -f congetture_beal_collatz_cramer congetture_beal_collatz_cra-
mer.o
```

## 5. Testing del Programma

I test effettuati rivelano che il programma accetta solo i tipi di dato richiesti come da specifica di programma, riuscendo a gestire qualsiasi caso di valore immesso inclusi errori di digitazione dell'utente quali immissione di simboli, spazi, caratteri oppure numeri semplicemente troppo grandi. La verifica dei risultati generati dall'esecuzione dei test ha dimostrato la correttezza di essi quindi, la corrispondenza dell'algoritmo alla sua specifica.

### 5.0 Test Menù di selezione congettura

#### Test 5.0.1

PROGRAMMA DI TEST CONGETTURE DI BEAL - COLLATZ - CRAMÉR

- [1] Beal
- [2] Collatz
- [3] Cramér

Digita il numero della tua scelta e premi 'Invio'.

a  
Valore fuori range!

PROGRAMMA DI TEST CONGETTURE DI BEAL - COLLATZ - CRAMÉR

- [1] Beal
- [2] Collatz
- [3] Cramér

Digita il numero della tua scelta e premi 'Invio'.

#### Test 5.0.2

PROGRAMMA DI TEST CONGETTURE DI BEAL - COLLATZ - CRAMÉR

- [1] Beal
- [2] Collatz
- [3] Cramér

Digita il numero della tua scelta e premi 'Invio'.

\$1  
Valore fuori range!

PROGRAMMA DI TEST CONGETTURE DI BEAL - COLLATZ - CRAMÉR

- [1] Beal
- [2] Collatz
- [3] Cramér

Digita il numero della tua scelta e premi 'Invio'.

### 5.1 Test congettura di Beal

PROGRAMMA DI TEST CONGETTURE DI BEAL - COLLATZ - CRAMÉR

- [1] Beal
- [2] Collatz
- [3] Cramér

Digita il numero della tua scelta e premi 'Invio'.  
Operazione scelta: 1

### Test 5.1.1

Equazione  $A^x + B^y = C^z$

Digita valore letterale  $A \geq 1$ : 3

Digita valore letterale  $B \geq 1$ : 6

Digita valore letterale  $C \geq 1$ : 3

Digita esponente  $x \geq 3$ : 3

Digita esponente  $y \geq 3$ : 3

Digita esponente  $z \geq 3$ : 5

Trovato fattore primo comune: 3

Equazione e congettura verificata per i parametri inseriti.

Vi è almeno un fattore primo in comune fra A,B e C.

### Test 5.1.2

Equazione  $A^x + B^y = C^z$

Digita valore letterale  $A \geq 1$ : 7

Digita valore letterale  $B \geq 1$ : 7

Digita valore letterale  $C \geq 1$ : 98

Digita esponente  $x \geq 3$ : 6

Digita esponente  $y \geq 3$ : 7

Digita esponente  $z \geq 3$ : 3

Trovato fattore primo comune: 7

Equazione e congettura verificata per i parametri inseriti.

Vi è almeno un fattore primo in comune fra A,B e C.

### Test 5.1.3

Equazione  $A^x + B^y = C^z$

Digita valore letterale  $A \geq 1$ : 7

Digita valore letterale  $B \geq 1$ : 7

Digita valore letterale  $C \geq 1$ : 14

Digita esponente  $x \geq 3$ : 3

Digita esponente  $y \geq 3$ : 4

Digita esponente  $z \geq 3$ : 3

Trovato fattore primo comune: 7

Equazione e congettura verificata per i parametri inseriti.

Vi è almeno un fattore primo in comune fra A,B e C.

### Test 5.1.4

Equazione  $A^x + B^y = C^z$

Digita valore letterale  $A \geq 1$ : 3

Digita valore letterale  $B \geq 1$ : a

Valore non accettabile!

Digita valore letterale  $B \geq 1$ : 6

Digita valore letterale  $C \geq 1$ : 3

Digita esponente  $x \geq 3$ : d

Valore non accettabile!

Digita esponente  $x \geq 3$ : 3

Digita esponente  $y \geq 3$ : 67

Digita esponente  $z \geq 3$ : 1029374

Potenza troppo elevata!

Digita esponente  $z \geq 3$ : 10

Trovato fattore primo comune: 3

Equazione e congettura non verificata per i parametri inseriti.

Vi è almeno un fattore primo in comune fra A,B e C

### Test 5.1.5

Equazione  $A^x + B^y = C^z$

Digita valore letterale  $A \geq 1$ : 1

Digita valore letterale  $B \geq 1$ : 2

Digita valore letterale  $C \geq 1$ : 3

Digita esponente  $x \geq 3$ : 3

Digita esponente  $y \geq 3$ : 3

Digita esponente  $z \geq 3$ : 3

Equazione e congettura non verificata per i parametri inseriti.

Non vi è alcun fattore primo fra A,B,C.

### Test 5.1.6

Equazione  $A^x + B^y = C^z$

Digita valore letterale  $A \geq 1$ : 1

Digita valore letterale  $B \geq 1$ : 2

Digita valore letterale  $C \geq 1$ : 3

Digita esponente  $x \geq 3$ : 3

Digita esponente  $y \geq 3$ : 3

Digita esponente  $z \geq 3$ : 3

Equazione e congettura non verificata per i parametri inseriti.

Non vi è alcun fattore primo fra A,B,C.

### Test 5.1.7

Equazione  $A^x + B^y = C^z$

Digita valore letterale  $A \geq 1$ : -12

Valore non accettabile!

Digita valore letterale  $A \geq 1$ : -

Valore non accettabile!

Digita valore letterale  $A \geq 1$ : 0

Valore non accettabile!

Digita valore letterale  $A \geq 1$ : 67

Digita valore letterale  $B \geq 1$ : +

Valore non accettabile!

Digita valore letterale  $B \geq 1$ : 12

Digita valore letterale  $C \geq 1$ : >C

Valore non accettabile!

Digita valore letterale  $C \geq 1$ : 12A

Digita esponente  $x \geq 3$ : A12

Valore non accettabile!

Digita esponente  $x \geq 3$ : 12A

Digita esponente  $y \geq 3$ : 23

Digita esponente  $z \geq 3$ : 2

Valore non accettabile!

Digita esponente  $z \geq 3$ : 1

Valore non accettabile!

Digita esponente  $z \geq 3$ : 0

Valore non accettabile!

Digita esponente  $z \geq 3$ : accetta

Valore non accettabile!

Digita esponente  $z \geq 3$ : 1 2

Valore non accettabile!

Digita esponente  $z \geq 3$ : 12

Equazione e congettura non verificata per i parametri inseriti.

Non vi è alcun fattore primo fra A,B,C.



### Test 5.1.8

Equazione  $A^x + B^y = C^z$

Digita valore letterale  $A \geq 1$ : 1

Digita valore letterale  $B \geq 1$ : 1

Digita valore letterale  $C \geq 1$ : 1

Digita esponente  $x \geq 3$ : 3

Digita esponente  $y \geq 3$ : 3

Digita esponente  $z \geq 3$ : 3

Equazione e congettura non verificata per i parametri inseriti.

Non vi è alcun fattore primo fra A,B,C.

### Test 5.1.9

Equazione  $A^x + B^y = C^z$

Digita valore letterale  $A \geq 1$ : 11

Digita valore letterale  $B \geq 1$ : 11

Digita valore letterale  $C \geq 1$ : 1

Digita esponente  $x \geq 3$ : 3

Digita esponente  $y \geq 3$ : 3

Digita esponente  $z \geq 3$ : 3

Equazione e congettura non verificata per i parametri inseriti.

Non vi è alcun fattore primo fra A,B,C.

### Test 5.1.10

Equazione  $A^x + B^y = C^z$

Digita valore letterale  $A \geq 1$ :

12

Digita valore letterale  $B \geq 1$ : 239828392839

Valore non accettabile!

Digita valore letterale  $B \geq 1$ : 11

Digita valore letterale  $C \geq 1$ : 11

Digita esponente  $x \geq 3$ : 11

Digita esponente  $y \geq 3$ : 11

Digita esponente  $z \geq 3$ : 11

Equazione e congettura non verificata per i parametri inseriti.

Non vi è alcun fattore primo fra A,B,C.

## 5.2 Test congettura di Collatz

PROGRAMMA DI TEST CONGETTURE DI BEAL - COLLATZ - CRAMÉR

[1] Beal

[2] Collatz

[3] Cramér

Digita il numero della tua scelta e premi 'Invio'.

Operazione scelta: 2

### Test 5.2.1

Digita un numero intero  $> 0$ :

0

Valore fuori range!

Digita un numero intero  $> 0$ :

1

Numeri generati:

2

1

La congettura è verificata. Ho raggiunto  $n = 1$

### Test 5.2.2

Digita un numero intero  $> 0$ :

-12

Valore fuori range!

Digita un numero intero  $> 0$ :

12

Numeri generati:

6

3

10

5

16

8

4

2

1

La congettura è verificata. Ho raggiunto  $n = 1$

### Test 5.2.3

Digita un numero intero  $> 0$ :

2

Numeri generati:

1

La congettura è verificata. Ho raggiunto  $n = 1$

### Test 5.2.4

Digita un numero intero  $> 0$ :

\$

Valore fuori range!

Digita un numero intero  $> 0$ :

\$ \$ &

Valore fuori range!

Digita un numero intero  $> 0$ :

>2

Valore fuori range!

Digita un numero intero  $> 0$ :

2A

Numeri generati:

1

La congettura è verificata. Ho raggiunto  $n = 1$

### Test 5.2.5

Digita un numero intero  $> 0$ :

34

Numeri generati:

17

52

26

13

40

20

10

5

16

8  
4  
2  
1

La congettura è verificata. Ho raggiunto  $n = 1$

### Test 5.2.6

Digita un numero intero  $> 0$ :

a

Valore fuori range!

Digita un numero intero  $> 0$ :

A

Valore fuori range!

Digita un numero intero  $> 0$ :

A

Valore fuori range!

Digita un numero intero  $> 0$ :

a

Valore fuori range!

Digita un numero intero  $> 0$ :

1

Numeri generati:

4  
2  
1

La congettura è verificata. Ho raggiunto  $n = 1$

### Test 5.2.7

Digita un numero intero  $> 0$ :

11

Numeri generati:

34  
17  
52  
26  
13  
40  
20  
10  
5  
16  
8  
4  
2  
1

La congettura è verificata. Ho raggiunto  $n = 1$

### Test 5.2.8

Digita un numero intero  $> 0$ :

-99999

Valore fuori range!

Digita un numero intero  $> 0$ :

-99

Valore fuori range!

Digita un numero intero  $> 0$ :

+99

Numeri generati:

298  
149

448  
224  
112  
56  
28  
14  
7  
22  
11  
34  
17  
52  
26  
13  
40  
20  
10  
5  
16  
8  
4  
2  
1

La congettura è verificata. Ho raggiunto  $n = 1$

### Test 5.2.9

Digita un numero intero  $> 0$ :

+1

Numeri generati:

4  
2  
1

La congettura è verificata. Ho raggiunto  $n = 1$

### Test 5.2.10

Digita un numero intero  $> 0$ :

(12)

Valore fuori range!

Digita un numero intero  $> 0$ :

(12

Valore fuori range!

Digita un numero intero  $> 0$ :

12)

Numeri generati:

6  
3  
10  
5  
16  
8  
4  
2  
1

La congettura è verificata. Ho raggiunto  $n = 1$

### 5.3 Test congettura di Cramér

PROGRAMMA DI TEST CONGETTURE DI BEAL - COLLATZ - CRAMÉR

- [1] Beal
- [2] Collatz
- [3] Cramér

Digita il numero della tua scelta e premi 'Invio'.  
Operazione scelta: 3

#### Test 5.3.1

Digita un numero primo  $\geq 11$  e premere 'Invio'.  
Successivamente digita un altro numero primo consecutivo al precedente e premere 'Invio'.

11  
13

Differenza tra i due numeri primi: 2  
Quadrato del Logaritmo naturale del primo minore: 5.75  
Rapporto tra la differenza dei due primi ed il quadrato del Logaritmo naturale del minore: 0.35

La congettura è verificata per i parametri inseriti

#### Test 5.3.2

Digita un numero primo  $\geq 11$  e premere 'Invio'.  
Successivamente digita un altro numero primo consecutivo al precedente e premere 'Invio'.

11  
11  
Valore non accettabile, hai inserito due primi identici oppure il secondo più piccolo  
13

Differenza tra i due numeri primi: 2  
Quadrato del Logaritmo naturale del primo minore: 5.75  
Rapporto tra la differenza dei due primi ed il quadrato del Logaritmo naturale del minore: 0.35

La congettura è verificata per i parametri inseriti

#### Test 5.3.3

Digita un numero primo  $\geq 11$  e premere 'Invio'.  
Successivamente digita un altro numero primo consecutivo al precedente e premere 'Invio'.

11  
23  
Valore non accettabile, non è un numero primo consecutivo al primo.  
13

Differenza tra i due numeri primi: 2  
Quadrato del Logaritmo naturale del primo minore: 5.75  
Rapporto tra la differenza dei due primi ed il quadrato del Logaritmo naturale del minore: 0.35

La congettura è verificata per i parametri inseriti

#### Test 5.3.4

Digita un numero primo  $\geq 11$  e premere 'Invio'.  
Successivamente digita un altro numero primo consecutivo al precedente e premere 'Invio'.

23  
24  
Non è un numero primo!  
25

29

Rapporto tra la differenza dei due primi ed il quadrato del Logaritmo naturale del minore: 0.61

Differenza tra i due numeri primi: 10

Quadrato del Logaritmo naturale del primo minore: 36.51

Rapporto tra la differenza dei due primi ed il quadrato del Logaritmo naturale del minore: 0.27

La congettura è verificata per i parametri inseriti

### Test 5.3.8

Digita un numero primo  $\geq 11$  e premere 'Invio'.

Successivamente digita un altro numero primo consecutivo al precedente e premere 'Invio'.

421+431

431+433

Differenza tra i due numeri primi: 10

Quadrato del Logaritmo naturale del primo minore: 36.51

Rapporto tra la differenza dei due primi ed il quadrato del Logaritmo naturale del minore: 0.27

La congettura è verificata per i parametri inseriti

### Test 5.3.9

Digita un numero primo  $\geq 11$  e premere 'Invio'.

Successivamente digita un altro numero primo consecutivo al precedente e premere 'Invio'.

uniurb

Valore fuori range!

B4

Valore fuori range!

la scelta

Valore fuori range!

\$\$\$

Valore fuori range!

3

Valore fuori range!

1

Valore fuori range!

1234567890

1234567890

Non è un numero primo!

Non è un numero primo!

123

Non è un numero primo!

231

Non è un numero primo!

233

247

Non è un numero primo!

239

Differenza tra i due numeri primi: 6

Quadrato del Logaritmo naturale del primo minore: 29.71

Rapporto tra la differenza dei due primi ed il quadrato del Logaritmo naturale del minore: 0.20

La congettura è verificata per i parametri inseriti

### Test 5.3.10

Digita un numero primo  $\geq 11$  e premere 'Invio'.

Successivamente digita un altro numero primo consecutivo al precedente e premere 'Invio'.

[]  
Valore fuori range!  
997  
1010  
Non è un numero primo!  
[1000]  
Valore fuori range!  
1003  
Non è un numero primo!  
999  
Non è un numero primo!  
1000  
Non è un numero primo!  
1001  
Non è un numero primo!  
1002  
Non è un numero primo!  
1003  
Non è un numero primo!  
1004  
Non è un numero primo!  
1005  
Non è un numero primo!  
1006  
Non è un numero primo!  
1007  
Non è un numero primo!  
1008  
Non è un numero primo!  
1009

Differenza tra i due numeri primi: 12

Quadrato del Logaritmo naturale del primo minore: 47.68

Rapporto tra la differenza dei due primi ed il quadrato del Logaritmo naturale del minore: 0.25

La congettura è verificata per i parametri inseriti



## 6. Verifica del Programma

### 6.0 Analisi iniziale

Si vuole effettuare la verifica della correttezza della funzione contenuta nella congettura di Collatz mediante il metodo delle triple di Hoare ed applicando le regole di Dijkstra.

Per verificare la correttezza del programma  $S$  che vuole calcolare un risultato definito come post condizione  $R$ , bisogna determinare il predicato  $Q$  o preconditione, che risolve l'equazione logica  $\{Q\}S\{R\} \equiv \text{vero}$ .  $S$  è noto come  $R$  del resto quindi rimane solo da ricavare  $Q$ .

### 6.1 Brano di codice scelto

```
if (numero_in % 2 == 0)
    numero_in = numero_in / 2;
else
    numero_in = (numero_in * 3) + 1;
```

Il brano di codice scelto può essere riscritto utilizzando variabili brevi ed aggiungendo dei marcatori che ci torneranno utili successivamente durante lo sviluppo della formula logica.

```
if (n % 2 == 0)          β
    n = n / 2;           S1
else
    n = (n * 3) + 1;     S2
```

### 6.2 Proprietà da verificare

La post condizione  $\{R\}$  è formalizzata come  $n = \{n \in \mathbb{N} \mid n \geq 1\}$  cioè  $n$  è sempre un numero intero positivo pari o superiore ad 1.

*Osservazione.* Il predicato  $\beta: ((n \% 2) == 0)$  può essere formalizzato:  $n = n' * 2$

*Osservazione.* Il predicato  $\neg\beta: ((n \% 2) != 0)$  può essere formalizzato:  $n = n' * 2 + 1$

### 6.3 Svolgimento

L'algoritmo  $S$  è rappresentato dall'istruzione di selezione  $S_1$ .

Denotiamo con  $wp(S, R)$  la preconditione più debole rispetto a  $S$  e  $R$ .

Applichiamo le regole di Dijkstra.

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

$$wp(S_1, R) = (n = \{n \in \mathbb{N} \mid n \geq 1\})_{n, \frac{n}{2}}$$

$$wp(S_2, R) = (n = \{n \in \mathbb{N} \mid n \geq 1\})_{n, n*3+1}$$

$$wp(S, R) = ((n = n' * 2 \rightarrow wp(S_1, R)) \wedge (n = n' * 2 + 1 \rightarrow wp(S_2, R))) \equiv (\text{vero} \wedge \text{vero}) \equiv \text{vero}$$

Il programma è quindi sempre corretto rispetto al problema.