

CORSO DI PROGRAMMAZIONE PROCEDURALE

RELAZIONE DEL PROGETTO PER LA SESSIONE INVERNALE

ANNO 2022/2023

Componenti del Team :

ELIA RENZONI e ALESSIO BIAGIOLI

Corso di Laurea in Informatica Applicata, Università degli studi di Urbino Carlo Bo.

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 Cramer asserisce che il valore assoluto della differenza tra due numeri primi consecutivi ≥ 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 ≥ 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 :

2.1 Dati di Ingresso del Problema :

I Dati di ingresso del problema sono rappresentati da numeri appartenenti all'insieme degli interi positivi (\mathbb{Z}^+) o naturali (\mathbb{N}).

Il numero e il valore dei dati di ingresso del problema sono diversi per ogni congettura.

Nel caso della congettura di Beal i dati di ingresso sono delle potenze di numeri interi positivi, dove le basi devono essere maggiori o uguali a 1, mentre gli esponenti maggiori o uguali a 3.

La congettura di Collatz implica come dato di ingresso un singolo numero strettamente maggiore di 0. Invece la congettura di Cramer comporta come dato di ingresso due numeri primi, maggiori o uguali a 11, e consecutivi.

2.2 Dati di Uscita del Problema :

I dati di uscita del problema sono rappresentati dal risultato delle varie congetture. A seconda delle diverse congetture il problema ha i seguenti dati di uscita :

- Congettura di Beal : i dati di uscita sono il fattore primo in comune tra le basi dell'equazione.
- Congettura di Collatz : i dati di uscita sono i numeri generati dalla funzione o un singolo numero.
- Congettura di Cramer : i dati di uscita sono i risultati della formula con la quale si verifica la congettura; quindi sono dati di uscita il valore assoluto della differenza tra i due numeri primi e il quadrato del logaritmo naturale del numero più piccolo.

2.3 Relazioni Intercorrenti tra i Dati del Problema :

Il termine congettura deriva dal latino “conicere”, che significa dedurre. Essa è una proprietà derivante da un’ipotesi che però non è ancora stata dimostrata.

Le relazioni tra i dati di ingresso e quelli di uscita cambiano in base alla congettura a cui si vuole fare riferimento :

- Congettura di Beal : essa ha come relazione tra i dati di ingresso e quelli di uscita l’applicazione della seguente equazione $a^x + b^y = c^z$. Se la somma delle potenze, nel primo membro, è uguale al risultato del secondo membro, allora la congettura è verificata; inoltre se ciò avviene le basi degli esponenti dell’equazione avranno un fattore primo in comune, il quale è un numero primo che se utilizzato come divisore, con dividendo le basi a , b e c , il resto dell’operazione di divisione è pari a 0.
- Congettura di Collatz : tale congettura implica come relazione intercorrente una funzione. La congettura risulta verificata quando una successione di numeri, la quale è determinata dall’applicazione della funzione in base al valore del dato di ingresso, contiene il numero 1, tuttavia se tale successione non contiene il suddetto numero la congettura non è verificata.

La funzione è applicata in base al valore del numero via via incontrato;

$$f(n) = \begin{cases} \frac{n}{2}, & \text{se } n \text{ pari,} \\ 3n + 1, & \text{se } n \text{ dispari.} \end{cases}$$

- Congettura di Cramer : questa congettura risulta essere verificata quando il valore assoluto della differenza tra due numeri primi consecutivi è sempre minore del quadrato del logaritmo del numero primo più piccolo; per cui occorre applicare le seguenti formule :
 - $|p_{n+1} - p_n|$ -> Valore assoluto della differenza tra i due numeri primi consecutivi;
 - $(\ln p_n)^2$ -> Quadrato del logaritmo naturale del numero primo più piccolo.

3. PROGETTAZIONE DELL'ALGORITMO :

3.1 Scelte di Progetto :

Strutture e Rappresentazione dei dati

Il numero scelto dall'utente, che identifica la congettura che si vuole verificare, viene denotato da delle costanti simboliche, in quanto in caso di cambiamenti, nella scelta di tale numero, basterà modificare solo il valore della costante simbolica.

I valori inseriti dall'utente, nel caso della congettura di Beal e di Cramer, sono memorizzati in degli array contenenti elementi di tipo *int*; invece il valore inserito rispetto alla congettura di Collatz viene memorizzato in una variabile di tipo *int*.

Per rendere l'acquisizione dei dati più semplice per l'utente, abbiamo deciso di creare una variabile di tipo stringa e due array contenenti elementi di tipo *char*. Nel primo caso la stringa viene utilizzata per l'acquisizione dei valori relativi alla congettura di Cramer, per comunicare all'utente se deve inserire il primo o il secondo numero, tale funzionalità risulta particolarmente utile nel caso in cui l'utente inserisce dei valori che non rispettano i vincoli posti per i dati di ingresso, per cui dovrà inserire il valore corretto dove ciò non è avvenuto, ad esempio se l'utente al posto del secondo numero inserisce un carattere l'acquisizione del dato ripartirà dal secondo numero e non dal primo, per cui è importante fare distinzione tra i due numeri. Nel secondo caso invece i due array di caratteri sono usati per indicare all'utente che dovrà inserire il giusto esponente per la relativa base; nel momento in cui inserirà un valore non corretto l'acquisizione dei dati ripartirà nell'esatto punto in cui l'utente ha sbagliato.

3.2 Passi dell'Algoritmo :

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

- Acquisire e validare il numero relativo alla congettura scelta dall'utente.
- In base all'operazione scelta dall'utente:
 - Acquisire e validare i dati di ingresso della congettura di Beal.
 - Calcolare se la congettura è verificata e i fattori primi in comune delle basi dell'equazione, sia nel caso in cui la congettura è verificata che nel caso in cui essa non lo è.

Oppure:

- Acquisire e validare i dati di ingresso della congettura di Collatz.
- Sviluppare ricorsivamente la successione di numeri che verifica la congettura:
 - Caso base: se il numero acquisito è uguale a 1 la congettura è verificata.
 - Caso generale: se il numero acquisito è maggiore di 1 e se e' pari si applica $f(n) = n / 2$, mentre se esso e' dispari si applica $f(n) = n * 3 + 1$
Inoltre la funzione si puo' applicare ripetutamente se il numero acquisito ha un valore elevato e necessita di più passaggi.

Oppure:

- Acquisire e validare dati di ingresso della congettura di Cramer.
- Calcolare se la congettura e' verificata tramite la formula.

Al fine di evitare problemi di scarsa leggibilità, il calcolo dei fattori primi in comune, nella congettura di Beal, è inserito all'interno di un'altra funzione anziché dentro la funzione per il calcolo della congettura, inoltre le istruzioni per verificare se un numero è primo o meno vengono inserite all'interno di una funzione apposita per evitare ridondanza di codice.

Il programma è articolato in sottoprogrammi per ogni funzionalità del programma stesso, in modo da rendere il file sorgente più leggibile e composto da sottoprogrammi indipendenti, così che la manutenzione del software sia più facile.

4. IMPLEMENTAZIONE DELL'ALGORITMO :

File sorgente : congetture_beal_cramer_collatz.c

```
/* **** */
/* Programma per la verifica delle congetture */
/* **** */

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

#include <stdio.h>
#include <math.h>
#include <stdlib.h>

/* **** */
/* definizione delle costanti simboliche */
/* **** */

#define MAX_ARR_BEAL 3          /* numero delle basi, per congettura di Beal */
#define MAX_ARR_CRAMER 2       /* numero dei valori, per congettura di Cramer */
#define BEAL_SCELTA 0          /* scelta congettura di Beal */
#define COLLATZ_SCELTA 1       /* scelta congettura di Collatz */
#define CRAMER_SCELTA 2        /* scelta congettura di Cramer */

/* **** */
/* dichiarazioni delle funzioni */
/* **** */

int acquisisci_scelta(void);
void acquisisci_beal(void);
void acquisisci_collatz(void);
void acquisisci_cramer(void);
int verifica_primi(int valori_numeri);
void calcola_beal(const int valore_basi[],
                  const int valore_esponenti[]);
void calcola_collatz(int);
void calcola_cramer(const int valore_numeri[]);
void fattore_primo(int basi_congettura[]);

/* **** */
/* definizioni delle funzioni */
/* **** */

/* definizione della funzione main */
int main(void)
{
    /* dichiarazione della variabile locale alla funzione */
    int confronto_scelta;          /* lavoro : scelta dell'utente */

    /* memorizza la scelta dell'utente */
    confronto_scelta = acquisisci_scelta();

    /* seleziona la congettura in base alla scelta */
    switch (confronto_scelta)
    {
        case BEAL_SCELTA:
            acquisisci_beal();
            break;

        case COLLATZ_SCELTA:
            acquisisci_collatz();
            break;

        case CRAMER_SCELTA:
            acquisisci_cramer();
```

```

        break;
    }

    return (0);
}

/* definizione della funzione per comunicare all'utente le scelte da effettuare
 * e validazione dei dati inseriti dall'utente */
int acquisisci_scelta(void)
{
    /* dichiarazione delle variabili locali alla funzione */
    int numero_scelta;          /* input : numero scelta */
    int acquisizione_errata,    /* lavoro : esito complessivo dell'acquisizione */
        esito_lettura;         /* lavoro : esito della scanf */

    /* acquisire la scelta dell'utente e validare i dati */
    do
    {
        printf(" SELEZIONA L'OPERAZIONE : \n");
        printf("*****\n");
        printf("* 0 - Congettura di Beal    *\n");
        printf("* 1 - Congettura di Collatz *\n");
        printf("* 2 - Congettura di Cramer  *\n");
        printf("*****\n");

        esito_lettura = scanf("%d",
                               &numero_scelta);
        acquisizione_errata = esito_lettura != 1 ||
                               numero_scelta < BEAL_SCELTA ||
                               numero_scelta > CRAMER_SCELTA;

        if (acquisizione_errata)
            printf("Errore, il valore non e' accettabile ! \n");
        while (getchar() != '\n');
    }
    while (acquisizione_errata);

    return (numero_scelta);
}

/* definizione della funzione per l'acquisizione e la validazione dei dati
 * relativi alla congettura di Beal */
void acquisisci_beal(void)
{
    /* dichiarazione delle variabili locali alla funzione */
    int  valore_base[MAX_ARR_BEAL],          /* input : valore delle basi */
        valore_esponenti[MAX_ARR_BEAL];     /* input : valore degli esponenti */
    char caratteri_base[] = {'a',           /* output : caratteri id. delle basi*/
                             'b',
                             'c'},
        caratteri_esponenti[] = {'x',      /* output : caratteri id. degli espo. */
                                  'y',
                                  'z'};

    int esito_lettura,                       /* lavoro : esito della scanf */
        acquisizione_errata,                /* lavoro : esito complessivo dell'acq. */
        i;                                  /* lavoro : variabile di controllo */

    /* acquisire e validare i dati relativi alla congettura di Beal */
    for (i = 0;
         (i < MAX_ARR_BEAL);
         i++)
    {
        do
        {
            printf("Inserisci il valore della base %c (>= 1) e dell'esponente %c (>= 3) : \n",
                   caratteri_base[i],
                   caratteri_esponenti[i]);

```



```

        esito_lettura = scanf("%d%d",
                                &valore_base[i],
                                &valore_esponenti[i]);

        acquisizione_errata = esito_lettura != 2 ||
                                valore_base[i] < 1 ||
                                valore_esponenti[i] < 3;

        if (acquisizione_errata)
            printf("Errore, valori non validi ! \n");
        while (getchar() != '\n');
    }
    while (acquisizione_errata);
}

/* trasferire i valori delle basi e degli esponenti alla seguente funzione
 * affinché verifichi la congettura */
calcola_beal(valore_base,
              valore_esponenti);
}

/* definizione della funzione per l'acquisizione e la validazione
 * dei dati relativi alla congettura di Collatz */
void acquisisci_collatz(void)
{
    /* dichiarazione delle variabili locali alla funzione */
    int numero;                /* input : valore del numero */
    int esito_lettura,          /* lavoro : esito della scanf */
        acquisizione_errata;   /* lavoro : esito complessivo dell'acq. */

    /* acquisizione e validazione del numero */
    do
    {
        printf("Inserisci un numero > 0 : \n");
        esito_lettura = scanf("%d",
                                &numero);
        acquisizione_errata = esito_lettura != 1 ||
                                numero <= 0;
        if (acquisizione_errata)
            printf("Errore, il valore inserito non e' valido ! \n");
        while (getchar() != '\n');
    }
    while (acquisizione_errata);

    /* trasferire il valore del numero acquisito alla seguente funzione
     * affinché verifichi la congettura */
    calcola_collatz(numero);
}

/* definizione della funzione per l'acquisizione e la validazione
 * dei dati relativi alla congettura di Cramer */
void acquisisci_cramer(void)
{
    /* dichiarazione delle variabili locali alla funzione */
    char numeri_ordine[2][15] = {"Primo Numero",          /* output : stringa id. numero */
                                   "Secondo Numero"};

    int valori_numeri[MAX_ARR_CRAMER];    /* input : valore dei numeri */
    int indice,                            /* lavoro : variabile di controllo del ciclo */
        acquisizione_errata,              /* lavoro : ver. il numero dei valori acq. */
        esito_lettura,                    /* lavoro : esito della scanf */
        valori_uguali,                    /* lavoro : ver. numeri uguali */
        valori_consecutivi,               /* lavoro : ver. numeri consecutivi o no */
        valore_minore,                    /* lavoro : ver. se il primo numero e' < del sec. */
        numeri_primi,                     /* lavoro : ver. numeri primi */
        valori_piccoli,                   /* lavoro : ver. numeri troppo piccoli */
        numero_compreso,                  /* lavoro : numero compreso tra i due numeri primi */

```

```

        verifica_numero_compreso,          /* lavoro : ver. se il numero compreso e' primo o meno */
        numero_minore,                    /* lavoro : contiene il numero piu' piccolo fra i due */
        numero_maggiore;                  /* lavoro : contiene il numero piu' grande fra i due */

for (indice = 0;
    (indice < MAX_ARR_CRAMER);
    indice++)
{
    /* acquisizione e validazione dei numeri */
    do
    {
        printf("Inserisci il valore del %s (Numero primo e >= 11) : \n",
            numeri_ordine[indice]);
        esito_lettura = scanf("%d",
            &valori_numeri[indice]);

        acquisizione_errata = esito_lettura != 1;
        valori_uguali = valori_numeri[indice] == valori_numeri[indice - 1];
        valori_piccoli = valori_numeri[indice] < 11;
        valore_minore = valori_numeri[indice] < valori_numeri[indice - 1];

        /* invocazione della funzione per verificare se i numeri sono primi */
        numeri_primi = verifica_primi(valori_numeri[indice]);

        if (valori_piccoli || acquisizione_errata)
        {
            printf("Errore, numero non valido ! \n");
        }

        /* se i numeri inseriti non sono primi */
        if (numeri_primi)
        {
            printf("Errore, il numero non e' primo ! \n");
        }
        /* se i numeri sono primi */
        else
        {
            if (indice == 1)
            {
                /* se i numeri primi sono uguali */
                if (valori_uguali)
                {
                    printf("Errore, i numeri sono uguali ! \n");
                }

                else
                {
                    /* se il secondo valore e' minore del primo */
                    if (valore_minore)
                    {
                        printf("Errore, il secondo numero deve essere maggiore del primo \n");
                    }
                    /* se il secondo valore e' maggiore del primo */
                    else
                    {
                        /* memorizza in due variabili diverse il
                         * numero maggiore e il minore */
                        numero_minore = valori_numeri[indice - 1];
                        numero_maggiore = valori_numeri[indice];

                        /* verifica se i numeri primi inseriti sono consecutivi*/
                        for (numero_compreso = numero_minore + 1,
                            valori_consecutivi = 0;
                            (numero_compreso < numero_maggiore);
                            numero_compreso++)
                        {
                            /* invocazione della funzione per verificare se i numeri compresi

```

```

        * tra i due primi sono anch'essi primi */
        verifica_numero_compreso = verifica_primi(numero_compreso);

        /* se i numeri compresi tra i due numeri primi inseriti
        * sono anch'essi primi*/
        if (verifica_numero_compreso == 0)
            valori_consecutivi = 1;
    }

    /* messaggio all'utente */
    if (valori_consecutivi)
        printf("Errore, i numeri non sono consecutivi ! \n");
}

}

}
while (getchar() != '\n');
}
while (acquisizione_errata ||
        valori_uguali ||
        valori_piccoli ||
        numeri_primi ||
        valori_consecutivi ||
        valore_minore);
}

/* trasferire il valore dei numeri alla seguente funzione
* affinché verifichi la congettura */
calcola_cramer(valori_numeri);
}

/* definizione della funzione per verificare se i numeri sono
* primi o meno */
int verifica_primi(int valori)
{
    /* dichiarazione delle variabili locali alla funzione */
    int valori_primi;          /* output : ver. se il numero e' primo o meno*/
    int conteggio = 0;         /* lavoro : numero di divisori */
    int divisore;              /* lavoro : id. numeri primi */

    /* controlla se i numeri sono primi o meno */
    for (divisore = 2;
        (divisore < valori);
        divisore++)
    {
        if (valori % divisore == 0)
            conteggio++;
    }

    valori_primi = conteggio != 0;

    return (valori_primi);
}

/* definizione della funzione per la verifica della congettura
* di Beal */
void calcola_beal(const int valore_basi[],          /* input : valore delle basi */
                  const int valore_esponenti[])    /* input : valore degli esponenti */
{
    /* dichiarazione delle variabili locali alla funzione */
    int contatore;                /* lavoro : variabile di controllo */
    int risultati[MAX_ARR_BEAL];  /* lavoro : valore delle potenze */
    int somma_risultati = 0;      /* lavoro : somma delle potenze */
    int basi_congettura[MAX_ARR_BEAL]; /* lavoro : valori delle basi della congettura */

    /* elevamento a potenza, somma tra le potenze e verifica

```

```

    /* se la congettura e' verificata o meno */
    for (contatore = 0;
        (contatore < MAX_ARR_BEAL);
        contatore++)
    {
        risultati[contatore] = pow(valore_basi[contatore],
                                   valore_esponenti[contatore]);
        basi_congettura[contatore] = valore_basi[contatore];

        if (contatore < 2)
        {
            somma_risultati += risultati[contatore];
        }
        else if (contatore == 2)
        {
            if (somma_risultati < risultati[contatore] ||
                somma_risultati > risultati[contatore])
            {
                printf("LA CONGETTURA NON E' VERIFICATA ! \n");
            }

            /* trasferire il valore delle basi a,b e c alla seguente funzione
             * per verificare se hanno fattori primi in comune */
            fattore_primo(basi_congettura);
        }
    }
}

/* definizione della funzione per la verifica della congettura
 * di Collatz */
void calcola_collatz(int num) /* input : valore del numero */
{
    /* dichiarazione della variabile locale alla funzione */
    int num_pari; /* lavoro : esito della verifica del numero */

    /* numero pari o dispari */
    num_pari = num % 2;
    /* caso base */
    if (num == 1)
    {
        printf("%d \n",
               num);
    }
    /* caso generale */
    else if (num > 1 && num_pari == 0)
    {
        printf("%d \n",
               num);
        calcola_collatz(num / 2);
    }
    else
    {
        printf("%d \n",
               num);
        calcola_collatz(num * 3 + 1);
    }
}

/* definizione della funzione per la verifica della congettura
 * di Cramer */
void calcola_cramer(const int valore_numeri[]) /* input : valore dei numeri */
{
    /* dichiarazione delle variabili locali alla funzione */
    double num_p; /* lavoro : numero piu' piccolo */
    int i, /* lavoro : variabile di controllo */
        differenza, /* lavoro : differenza tra i due numeri */
        val_ass; /* lavoro : valore assoluto della differenza */

```

```

double log_natu,          /* lavoro : logaritmo naturale del num. piu' piccolo */
      quadr_ln;          /* lavoro : quadrato del logaritmo naturale */

/* calcola la differenza tra i numeri e assegna il numero
 * piu' piccolo alla variabile num_p */
for (i = 0;
     (i < MAX_ARR_CRAMER);
     i++)

    if (i == 1)
    {
        differenza = valore_numeri[i] - valore_numeri[i - 1];
        num_p = valore_numeri[i - 1];
    }

/* calcola e comunica all'utente i risultati delle due
 * operazioni */
val_ass = abs(differenza);
printf("Valore assoluto della differenza : %d \n",
      val_ass);
log_natu = log(num_p);
quadr_ln = pow(log_natu, 2.0);
printf("Quadrato del ln del numero piu' piccolo : %f \n",
      quadr_ln);
}

/* definizione della funzione per calcolare e comunicare
 * all'utente se le basi a,b e c hanno un fattore primo
 * in comune e stamparlo */
void fattore_primo(int basi_congettura[]) /* input : valore delle basi */
{
    /* dichiarazione delle variabili locali alla funzione */
    int contatore,          /* lavoro : variabile di controllo */
        divisore;          /* lavoro : identifica i numeri primi divisori */
    int numero_primo;       /* lavoro : ver. se i divisori sono primi o meno */

    /* controlla se vi sono fattori primi in comune */
    for (contatore = 0;
         (contatore < MAX_ARR_BEAL);
         contatore++)
    {
        divisore = 2;
        while (divisore <= basi_congettura[contatore] &&
              divisore <= basi_congettura[contatore + 1] &&
              divisore <= basi_congettura[contatore + 2])
        {
            /* invocazione della funzione per verificare se la
             * variabile divisore contiene numeri primi o meno*/
            numero_primo = verifica_primi(divisore);

            /* se divisore e' un numero primo */
            if (numero_primo == 0)
            {
                /* fino a quando i valori delle basi sono divisibili
                 * per il divisore, divide il loro valore per il divisore
                 * e stampa il valore della variabile divisore */
                while (basi_congettura[contatore] % divisore == 0 &&
                      basi_congettura[contatore + 1] % divisore == 0 &&
                      basi_congettura[contatore + 2] % divisore == 0)
                {
                    basi_congettura[contatore] /= divisore;
                    basi_congettura[contatore + 1] /= divisore;
                    basi_congettura[contatore + 2] /= divisore;

                    printf("Fattore primo in comune : %d \n",
                          divisore);
                }
            }
        }
    }
}

```

```

        }
    }
    divisore++;
}
}
}

```

Makefile :

```

calcolo_congetture: congetture_beal_cramer_collatz.c Makefile
    gcc -ansi -Wall -O congetture_beal_cramer_collatz.c -o congetture_beal_cramer_collatz -lm
pulisci:
    rm -f congetture_beal_cramer_collatz.o
pulisci_tutto:
    rm -f congetture_beal_cramer_collatz congetture_beal_cramer_collatz.o

```

5. TESTING DEL PROGRAMMA :

Il testing viene effettuato per ogni input che il programma deve ricevere e in base ai vincoli posti per i dati di ingresso. Tali vincoli devono essere rispettati e quindi nel momento in cui l'utente inserisce un dato errato l'acquisizione si ripete.

I test di seguito riportati sono suddivisi in base ai diversi input che il programma deve ricevere affinché produca gli output attesi, senza problemi inerenti alla terminazione e in totale conformità rispetto alla specifica del problema.

- Test per la scelta dell'utente :
Gli input relativi a questa sezione del programma devono coincidere con i seguenti dati :
numero 0 per selezionare la congettura di Beal, numero 1 per selezionare la congettura di Collatz e numero 2 per selezionare la congettura di Cramer; tutti i valori diversi da quelli precedenti sono considerati come non accettabili.

Test 1

SELEZIONA L'OPERAZIONE :

* 0 - Congettura di Beal *

* 1 - Congettura di Collatz *

* 2 - Congettura di Cramer *

a

Errore, il valore non e' accettabile !

SELEZIONA L'OPERAZIONE :

* 0 - Congettura di Beal *

* 1 - Congettura di Collatz *

* 2 - Congettura di Cramer *

0

Inserisci il valore della base a ≥ 1 :

Test 2

SELEZIONA L'OPERAZIONE :

- * 0 - Congettura di Beal *
 - * 1 - Congettura di Collatz *
 - * 2 - Congettura di Cramer *
- *****

1

Inserisci un numero > 0

Test 3

SELEZIONA L'OPERAZIONE :

- * 0 - Congettura di Beal *
 - * 1 - Congettura di Collatz *
 - * 2 - Congettura di Cramer *
- *****

ss23ks

Errore, il valore non e' accettabile !

SELEZIONA L'OPERAZIONE :

- * 0 - Congettura di Beal *
 - * 1 - Congettura di Collatz *
 - * 2 - Congettura di Cramer *
- *****

2

Inserisci il valore del primo numero (Numero primo e ≥ 11) :

Test 4

SELEZIONA L'OPERAZIONE :

* 0 - Congettura di Beal *

* 1 - Congettura di Collatz *

* 2 - Congettura di Cramer *

45

Errore, il valore non e' accettabile !

SELEZIONA L'OPERAZIONE :

* 0 - Congettura di Beal *

* 1 - Congettura di Collatz *

* 2 - Congettura di Cramer *

1

Inserisci un numero > 0

- Test per la congettura di Beal :

I dati di ingresso relativi alla congettura di Beal corrispondono ai valori delle basi a , b , c e dei relativi esponenti x , y e z . I valori devono essere maggiori o uguali a 1, nel caso delle tre basi, e maggiori o uguali a 3 per gli esponenti. Viene inoltre dato per scontato che l'utente inserisca il numero corretto affinché possano essere eseguite le funzioni relative al calcolo della congettura di Beal.

Test 1

Dati di ingresso :

Inserisci il valore della base a (≥ 1) e dell'esponente x (≥ 3) :

24

3

Inserisci il valore della base b (≥ 1) e dell'esponente y (≥ 3) :

0

4

Errore, valori non validi !

Inserisci il valore della base b (≥ 1) e dell'esponente y (≥ 3) :

4

5

Inserisci il valore della base c (≥ 1) e dell'esponente z (≥ 3) :

12

4

Dati di uscita :

LA CONGETTURA NON E' VERIFICATA !

Fattore primo in comune : 2

Fattore primo in comune : 2

Test 2

Dati di ingresso :

Inserisci il valore della base a (≥ 1) e dell'esponente x (≥ 3) :

7

4

Inserisci il valore della base b (≥ 1) e dell'esponente y (≥ 3) :

8

3

Inserisci il valore della base c (≥ 1) e dell'esponente z (≥ 3) :

9

4

Dati di uscita :

LA CONGETTURA NON E' VERIFICATA !

Test 3

Dati di ingresso :

Inserisci il valore della base a (≥ 1) e dell'esponente x (≥ 3) :

25

3

Inserisci il valore della base b (≥ 1) e dell'esponente y (≥ 3) :

5

5

Inserisci il valore della base c (≥ 1) e dell'esponente z (≥ 3) :

-678

4

Errore, valori non validi !

Inserisci il valore della base c (≥ 1) e dell'esponente z (≥ 3) :

20

3

Dati di uscita :

LA CONGETTURA NON E' VERIFICATA !

Fattore primo in comune : 5

Test 4

Dati di ingresso :

Inserisci il valore della base a (≥ 1) e dell'esponente x (≥ 3) :

3

3

Inserisci il valore della base b (≥ 1) e dell'esponente y (≥ 3) :

ciao

Errore, valori non validi !

Inserisci il valore della base b (≥ 1) e dell'esponente y (≥ 3) :

6

3

Inserisci il valore della base c (≥ 1) e dell'esponente z (≥ 3) :

3

5

Dati di uscita :

Fattore primo in comune : 3

Test 5

Dati di ingresso :

Inserisci il valore della base a (≥ 1) e dell'esponente x (≥ 3) :

8

4

Inserisci il valore della base b (≥ 1) e dell'esponente y (≥ 3) :

7

3

Inserisci il valore della base c (≥ 1) e dell'esponente z (≥ 3) :

11

3

Dati di uscita :

LA CONGETTURA NON E' VERIFICATA !

Test 6

Dati di ingresso :

Inserisci il valore della base a (≥ 1) e dell'esponente x (≥ 3) :

8

4

Inserisci il valore della base b (≥ 1) e dell'esponente y (≥ 3) :

11

3

Inserisci il valore della base c (≥ 1) e dell'esponente z (≥ 3) :

22

3

Dati di uscita del problema :

LA CONGETTURA NON E' VERIFICATA !

Test 7

Dati di ingresso :

Inserisci il valore della base a (≥ 1) e dell'esponente x (≥ 3) :

23

3

Inserisci il valore della base b (≥ 1) e dell'esponente y (≥ 3) :

4

8

Inserisci il valore della base c (≥ 1) e dell'esponente z (≥ 3) :

3

9

Dati di uscita :

LA CONGETTURA NON E' VERIFICATA !

Test 8

Dati di ingresso :

Inserisci il valore della base a (≥ 1) e dell'esponente x (≥ 3) :

22

3

Inserisci il valore della base b (≥ 1) e dell'esponente y (≥ 3) :

16

3

Inserisci il valore della base c (≥ 1) e dell'esponente z (≥ 3) :

6

6

Dati di uscita :

LA CONGETTURA NON E' VERIFICATA !

Fattore primo in comune : 2

Test 9

Dati di ingresso :

Inserisci il valore della base a (≥ 1) e dell'esponente x (≥ 3) :

28

3

Inserisci il valore della base b (≥ 1) e dell'esponente y (≥ 3) :

9

4

Inserisci il valore della base c (≥ 1) e dell'esponente z (≥ 3) :

!!!2

Errore, valori non validi !

Inserisci il valore della base c (≥ 1) e dell'esponente z (≥ 3) :

/*llo

Errore, valori non validi !

Inserisci il valore della base c (≥ 1) e dell'esponente z (≥ 3) :

3

5

Dati di uscita :

LA CONGETTURA NON E' VERIFICATA !

Test 10

Dati di ingresso :

Inserisci il valore della base a (≥ 1) e dell'esponente x (≥ 3) :

7

3

Inserisci il valore della base b (≥ 1) e dell'esponente y (≥ 3) :

7

4

Inserisci il valore della base c (≥ 1) e dell'esponente z (≥ 3) :

14

3

Dati di uscita :

Fattore primo in comune : 7

- Test per la congettura di Collatz :

Per calcolare la congettura di Collatz il programma ha bisogno di un numero appartenente all'insieme dei numeri naturali, o interi positivi, e strettamente maggiore di 0. Se l'utente inserisce un valore diverso da quello stabilito il programma stampa un messaggio di errore, e successivamente chiede nuovamente il valore all'utente.

Anche in questo caso si dà per scontato che l'utente inserisca il numero corretto affinché possano essere eseguite le funzioni relative alla congettura di Collatz.

Test 1

Dati di ingresso :

Inserisci un numero > 0

245

Dati di uscita :

245

736

368

184

92

46

23

70

35
106
53
160
80
40
20
10
5
16
8
4
2
1

Test 2

Dati di ingresso :

Inserisci un numero > 0 :

a

Errore, il valore inserito non e' valido !

Inserisci un numero > 0 :

-78

Errore, il valore inserito non e' valido !

Inserisci un numero > 0 :

12

Dati di uscita :

12

6

3

10

5

16

8

4

2

1

Test 3

Dati di ingresso :

Inserisci un numero > 0 :

2500

Dati di uscita :

2500

1250

625

1876

938

469

1408

704

352

176

88

44

22

11

34

17

52

26

13

40

20

10

5

16

8

4

2

1

Test 4

Dati di ingresso :

Inserisci un numero > 0 :

|

Errore, il valore inserito non e' valido !

Inserisci un numero > 0 :

456

Dati di uscita :

456

228

114

57

172

86

43

130

65

196

98

49

148

74

37

112

56

28

14

7

22

11

34

17

52

26

13

40

20

10

5

16

8
4
2
1

Test 5

Dati di ingresso :

Inserisci un numero > 0 :

15

Dati di uscita :

15

46

23

70

35

106

53

160

80

40

20

10

5

16

8

4

2

1

Test 6

Dati di ingresso :

Inserisci un numero > 0 :

4

Dati di uscita :

4

2

1

Test 7

Dati di ingresso :

Inserisci un numero > 0

67

Dati di uscita :

67

202

101

304

152

76

38

19

58

29

88

44

22

11

34

17

52

26

13

40

20

10

5

16

8

4

2

1

Test 8

Dati di ingresso :

Inserisci un numero > 0 :

1

Dati di uscita :

1

Test 9

Dati di ingresso :

Inserisci un numero > 0 :

70

Dati di uscita :

70

35

106

53

160

80

40

20

10

5

16

8

4

2

1

Test 10

Dati di ingresso :

Inserisci un numero > 0 :

104

Dati di uscita :

104

52

26

13

40

20

10
5
16
8
4
2
1

- Test per la congettura di Cramer :
Per verificare la congettura di Cramer occorre inserire due numeri primi maggiori o uguali a 11 e consecutivi. Se l'utente inserisce dei numeri non primi, non consecutivi, uguali, non numeri o minori di 11, il programma stampa un messaggio di errore e chiede all'utente di inserire i dati corretti.
Anche in questo caso si dà per scontato che l'utente abbia inserito il numero corretto affinché possano essere eseguite le funzioni relative al calcolo della congettura di Cramer.

Test 1

Dato di input :

Inserisci il valore del Primo Numero (Numero primo e ≥ 11) :

23

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

37

Errore, i numeri non sono consecutivi !

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

29

Dati di uscita :

Valore assoluto della differenza : 6

Quadrato del ln del numero piu' piccolo : 9.831324

Test 2

Dati di ingresso :

Inserisci il valore del Primo Numero (Numero primo e ≥ 11) :

827

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

799

Errore, il numero non e' primo !

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

853

Errore, i numeri non sono consecutivi !

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

839

Errore, i numeri non sono consecutivi !

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

829

Dati di uscita :

Valore assoluto della differenza : 2

Quadrato del ln del numero piu' piccolo : 45.128900

Test 3

Dati di ingresso :

Inserisci il valore del Primo Numero (Numero primo e ≥ 11) :

7

Errore, numero non valido !

Inserisci il valore del Primo Numero (Numero primo e ≥ 11) :

17

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

17

Errore, i numeri sono uguali !

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

19

Dati di uscita :

Valore assoluto della differenza : 2

Quadrato del ln del numero piu' piccolo : 8.027098

Test 4

Dati di ingresso :

Inserisci il valore del Primo Numero (Numero primo e ≥ 11) :

2

Errore, numero non valido !

Inserisci il valore del Primo Numero (Numero primo e ≥ 11) :

47

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

1069

Errore, i numeri non sono consecutivi !

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

23

Errore, il secondo numero deve essere maggiore del primo

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

53

Dati di uscita :

Valore assoluto della differenza : 6

Quadrato del ln del numero piu' piccolo : 14.823637

Test 5

Dati di ingresso :

Inserisci il valore del Primo Numero (Numero primo e ≥ 11) :

709

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

719

Dati di uscita :

Valore assoluto della differenza : 10

Quadrato del ln del numero piu' piccolo : 43.084199

Test 6

Dati di ingresso :

Inserisci il valore del Primo Numero (Numero primo e ≥ 11) :

313

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

7

Errore, numero non valido !

Errore, il secondo numero deve essere maggiore del primo

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

317

Dati di uscita :

Valore assoluto della differenza : 4

Quadrato del ln del numero piu' piccolo : 33.018851

Test 6

Dati di ingresso :

Inserisci il valore del Primo Numero (Numero primo e ≥ 11) :

1009

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

983

Errore, il secondo numero deve essere maggiore del primo

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

1013

Dati di uscita :

Valore assoluto della differenza : 4

Quadrato del ln del numero piu' piccolo : 47.840947

Test 7

Dati di ingresso :

Inserisci il valore del Primo Numero (Numero primo e ≥ 11) :

5

Errore, numero non valido !

Inserisci il valore del Primo Numero (Numero primo e ≥ 11) :

11

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

5

Errore, numero non valido !

Errore, il secondo numero deve essere maggiore del primo

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

13

Dati di uscita :

Valore assoluto della differenza : 2

Quadrato del ln del numero piu' piccolo : 5.749902

Test 8

Dati di ingresso :

Inserisci il valore del Primo Numero (Numero primo e ≥ 11) :

643

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

647

Dati di uscita :

Valore assoluto della differenza : 4

Quadrato del ln del numero piu' piccolo : 41.811028

Test 9

Dati di ingresso :

Inserisci il valore del Primo Numero (Numero primo e ≥ 11) :

535

Errore, il numero non e' primo !

Inserisci il valore del Primo Numero (Numero primo e ≥ 11) :

541

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

547

Dati di uscita :

Valore assoluto della differenza : 6

Quadrato del ln del numero piu' piccolo : 39.607126

Test 10

Dati di ingresso :

Inserisci il valore del Primo Numero (Numero primo e ≥ 11) :

10

Errore, numero non valido !

Errore, il numero non e' primo !

Inserisci il valore del Primo Numero (Numero primo e ≥ 11) :

104

Errore, il numero non e' primo !

Inserisci il valore del Primo Numero (Numero primo e ≥ 11) :

89

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

53

Errore, il secondo numero deve essere maggiore del primo

Inserisci il valore del Secondo Numero (Numero primo e ≥ 11) :

97

Dati di uscita :

Valore assoluto della differenza : 8

Quadrato del ln del numero piu' piccolo : 20.147856

6. VERIFICA DEL PROGRAMMA :

6.1 Brano di Codice Scelto :

```
valori_uguali = valori_numeri[indice] == valori_numeri[indice - 1];  
valori_piccoli = valori_numeri[indice] < 11;  
valore_minore = valori_numeri[indice] < valori_numeri[indice - 1];
```

Il seguente brano di codice fa parte della funzione per l'acquisizione e la validazione dei dati relativi alla congettura di Cramer.

6.2 Proprieta' da Verificare :

Tramite le regole di Dijkstra e la tripla di Hoare si vuole verificare che il brano di codice scelto esegua correttamente le istruzioni in esso contenute.

Il brano di codice scelto viene utilizzato per valutare se i valori degli elementi dell'array sono uguali, minori di 11 o se il secondo elemento è minore del primo. Le tre variabili conterranno dei valori di verità che verranno assegnati in seguito alla valutazione delle espressioni.

Il brano di codice scelto è composto da tre diverse istruzioni S , le quali comportano tre diverse postcondizioni R .

Le tre istruzioni possono essere indicate nel seguente modo :

- 1) S_1
- 2) S_2
- 3) S_3

Le postcondizioni, quindi i valori di verità che le variabili conterranno a un certo momento dell'esecuzione del programma, dipendono dalla valutazione delle tre seguenti espressioni :

- 1) `valori_numeri[indice] == valori_numeri[indice - 1];`
- 2) `valori_numeri[indice] < 11;`
- 3) `valori_numeri[indice] < valori_numeri[indice - 1];`

Il nome della variabile di tipo array, per formalizzare le postcondizioni più facilmente, sarà denotata con *valn*, con pedice i o $i - 1$. Inoltre i

nomi delle variabili, in cui verrà assegnato il valore di verità, saranno annotate nel seguente modo : $valU$, $valP$, $valM$.

Il valore di verità, che sarà contenuto nelle variabili $valU$, $valP$ e $valM$, viene indicato con T (verità = 1) o F (falsità = 0).

Le postcondizioni possono essere formalizzate nel seguente modo :

$$R_1 = uguaglianza(valn_i, valn_{i-1}) = T \vee F$$

$$R_2 = min(valn_i, 11) = T \vee F$$

$$R_3 = min(valn_i, valn_{i-1}) = T \vee F$$

6.3 Svolgimento :

La preconditione più debole $wp(S, R)$ viene posta diversa da *vero*, in questo modo la correttezza del programma rispetto al problema potrebbe non dipendere dallo stato iniziale della computazione.

Applicando meccanicamente le regole di Dijkstra otteniamo :

$$\begin{aligned} wp(S_1, R_1) &= (valU = T \vee valU = F)_{valU, uguaglianza(valn_i, valn_{i-1})} \\ &= (uguaglianza(valn_i, valn_{i-1}) = T \vee uguaglianza(valn_i, valn_{i-1}) = F) \end{aligned}$$

$$\begin{aligned} wp(S_2, R_2) &= (valP = T \vee valP = F)_{valP, min(valn_i, 11)} \\ &= (min(valn_i, 11) = T \vee min(valn_i, 11) = F) \\ wp(S_3, R_3) &= (valM = T \vee valM = F)_{valM, min(valn_i, valn_{i-1})} \\ &= (min(valn_i, valn_{i-1}) = T \vee min(valn_i, valn_{i-1}) = F) \end{aligned}$$