

```

/*****
/* programma per verifica congetture di Beal, Collatz e Cramer */
*****/

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

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

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

#define FRASE1 1 /* stampare prima frase */
#define FRASE2 2 /* stampare seconda frase */
#define FRASE3 3 /* stampare terza frase */

/*****/
/* definizione dei tipi */
/*****/

typedef enum {uscita,
             congett_beal,
             congett_collatz,
             congett_cramer,
             }congett_t; /* tipo congettura */

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

int leggi_congettura(void);

int acquisisci_naturale(char *,
                       int);

void beal(int,
          int,
          int,
          int,
          int,
          int);

void collatz(int);

void cramer(int,
            int);

bool primo(int);

bool primo_consec(int,
                  int);

char uscita_programma(void);

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

```

```

/* definizione della funzione main */
int main(void)
{
    /* dichiarazione delle variabili locali alla funzione */
    int      a,          /* input: valore inserito dall'utente utilizzato
                           per le congetture di Beal, Collatz e Cramer */
             b,          /* input: valore inserito dall'utente utilizzato
                           per le congetture di Beal e Cramer */
             c,          /* input: valore inserito dall'utente utilizzato
                           per la congettura di Beal */
             x,          /* input: valore inserito dall'utente utilizzato
                           per la congettura di Beal */
             y,          /* input: valore inserito dall'utente utilizzato
                           per la congettura di Beal */
             z;          /* input: valore inserito dall'utente utilizzato
                           per la congettura di Beal */
    congetture_t cong;    /* input: congettura da verificare */

    do
    {
        /* acquisire congettura da verificare */
        cong = leggi_congettura();

        /* verificare congettura acquisita */
        switch (cong)
        {
            /* verifica congettura di Beal */
            case congett_beal:
                printf("\nCONGETTURA DI BEAL\n");

                a = acquisisci_naturale("primo",
                                         FRASE1);
                x = acquisisci_naturale("primo",
                                         FRASE2);
                b = acquisisci_naturale("secondo",
                                         FRASE1);
                y = acquisisci_naturale("secondo",
                                         FRASE2);
                c = acquisisci_naturale("terzo",
                                         FRASE1);
                z = acquisisci_naturale("terzo",
                                         FRASE2);

                printf("\n");

                beal(a,x,b,y,c,z);

                printf("\n\n");
                break;

            /* verifica congettura di Collatz */
            case congett_collatz:
                printf("\nCONGETTURA DI COLLATZ\n");

                a = acquisisci_naturale("",
                                         FRASE1);

                printf("\n");

                if (a == 1)
                {
                    printf("%d ",
                            a);
                    collatz(3 * a + 1);
                }
                else

```

```

        collatz(a);

        printf("\n\n");
        break;

/* verifica congettura di Cramer */
case congett_cramer:
    printf("\nCONGETTURA DI CRAMER\n");

    a = acquisisci_naturale("primo",
                           FRASE3);
    b = acquisisci_naturale("secondo",
                           FRASE3);
    printf("\n");

    cramer(a,b);

    printf("\n\n");
    break;

default:
    break;
}

/* scelta dell'utente se vuole continuare a testare le congetture o
uscire dal programma */
cong = uscita_programma();

printf("\n");
}
while (cong != uscita);

return(0);
}

/* definizione della funzione per acquisire un numero naturale */
int acquisisci_naturale(char *messaggio, /* input: messaggio specifico */
                       int frase)      /* input: frase specifica */
{
    /* dichiarazione delle variabili locali alla funzione */
    int n; /* output: numero naturale da acquisire */
    int esito_lettura, /* lavoro: esito della scanf */
        acquisizione_errata; /* lavoro: esito complessivo
dell'acquisizione */

    /* leggere e validare il numero naturale */
    do
    {
        /* acquisire il numero naturale n > 0 utilizzato per la congettura
di Beal e Collatz */
        if (frase == FRASE1)
        {
            printf("Digita il %s numero (naturale >0): ",
                   messaggio);
            esito_lettura = scanf("%d",
                                &n);
            acquisizione_errata = esito_lettura != 1 || n < 1;
        }

        /* acquisire il numero naturale n >= 3 (esponente) utilizzato per la
congettura di Beal */
        else if (frase == FRASE2)
        {
            printf("Digita l'esponente (naturale >=3) del %s numero: ",

```

```

        messaggio);
        esito_lettura = scanf("%d",
                               &n);
        acquisizione_errata = esito_lettura != 1 || n < 3;
    }

    /* acquisire il numero naturale n >= 11 utilizzato per la congettura
    di Cramer */
    else if (frase == FRASE3)
    {
        printf("Digita il %s numero naturale (>=11): ",
               messaggio);
        esito_lettura = scanf("%d",
                               &n);
        acquisizione_errata = esito_lettura != 1 || n < 11;
    }

    if (acquisizione_errata)
        printf("Valore non accettabile!\n");
    while (getchar() != '\n');
}
while (acquisizione_errata);
return(n);
}

/* definizione della funzione per leggere la congettura da verificare */
int leggi_congettura(void)
{
    /* dichiarazione delle variabili locali alla funzione */
    int congettura_acquisita;    /* output: congettura da verificare */
    int esito_lettura,          /* lavoro: esito della scanf */
        acquisizione_errata;    /* lavoro: esito complessivo
                                dell'acquisizione */

    /* stampare le possibili congetture da verificare */
    printf("Congetture disponibili:\n");
    printf("1 Congettura di Beal\n");
    printf("2 Congettura di Collatz\n");
    printf("3 Congettura di Cramer\n");

    /* leggere e validare la congettura acquisita */
    do
    {
        printf("Digita la congettura da verificare (%d-%d): ",
               congett_beal,
               congett_cramer);
        esito_lettura = scanf("%d",
                               &congettura_acquisita);
        acquisizione_errata = esito_lettura != 1 ||
                               congettura_acquisita < congett_beal ||
                               congettura_acquisita > congett_cramer;
        if (acquisizione_errata)
            printf("Valore non accettabile!\n");
        while (getchar() != '\n');
    }
    while (acquisizione_errata);
    return(congettura_acquisita);
}

/* definizione della funzione per verificare la congettura di Beal */
void beal(int a, /* input: prima base dell'equazione */
          int x, /* input: esponente della prima base dell'equazione */
          int b, /* input: seconda base dell'equazione */)

```

```

    int y, /* input: esponente della seconda base dell'equazione */
    int c, /* input: terza base dell'equazione */
    int z) /* input: esponente della terza base dell'equazione */

{

    /* dichiarazione delle variabili locali alla funzione */
    int fatt_primo; /* output: fattore primo in comune tra le basi
                    dell'equazione */
    int min;        /* lavoro: valore minimo tra le basi dell'equazione */
    int i;          /* lavoro: indice di scorrimento */

    /* verifica valore minimo tra le basi dell'equazione */
    min = a;
    if (b < min)
        min = b;
    if (c < min)
        min = c;

    fatt_primo = 0;
    i = 2;

    /* verifica esistenza di fattori primi in comune tra le basi
    dell'equazione */
    while (i <= min && fatt_primo == 0)
    {
        if (primo(i) == 0 && a % i == 0 && b % i == 0 && c % i == 0)
            fatt_primo = i;
        if (i == 2)
            ++i;
        else
            i+=2;
    }

    /* verifica della congettura e stampa dell'esito */
    if (pow(a,x) + pow(b,y) == pow(c,z) && fatt_primo != 0)
    {
        printf("Congettura di Beal verificata!");
        printf("\nL'equazione (a^x + b^y = c^z) vale.\n");
        printf("Esiste il seguente fattore primo in comune: %d",
            fatt_primo);
    }
    else if (pow(a,x) + pow(b,y) != pow(c,z) && fatt_primo != 0)
    {
        printf("Congettura di Beal non verificata!");
        printf("\nL'equazione (a^x + b^y = c^z) non vale.\n");
        printf("Esiste il seguente fattore primo in comune: %d",
            fatt_primo);
    }
    else if (pow(a,x) + pow(b,y) != pow(c,z) && fatt_primo == 0)
    {
        printf("Congettura di Beal non verificata!");
        printf("\nL'equazione (a^x + b^y = c^z) non vale.\n");
        printf("Non esiste alcun fattore primo in comune!");
    }

    else if (pow(a,x) + pow(b,y) == pow(c,z) && fatt_primo == 0)
    {
        printf("Congettura di Beal confutata!");
        printf("\nL'equazione (a^x + b^y = c^z) vale.\n");
        printf("Non esiste alcun fattore primo in comune!");
    }
}

```

```

/* definizione della funzione per verificare se un numero e' primo */
bool primo(int val)      /* input: valore da verificare */
{
    /* dichiarazione delle variabili locali alla funzione */
    int i;                /* lavoro: indice di scorrimento */
    bool num_prim;        /* output: restituisce 0 se il valore e' primo, 1 se il
                           valore non e' primo */

    num_prim = 0;

    /* verifica primalita' del valore */
    if (val != 2 && ((val % 2) == 0 || val < 2))
    {
        num_prim = 1;
    }
    else
        for (i = 3;
              (i <= sqrt(val));
              i += 2)
        {
            if ((val % i) == 0)
                num_prim = 1;
        }
    return (num_prim);
}

/* definizione della funzione per verificare se due numeri sono primi
consecutivi */
bool primo_consec(int val1, /* input: primo valore */
                  int val2) /* input: secondo valore */
{
    /* dichiarazione delle variabili locali alla funzione */
    bool trov_prim; /* output: restituisce 0 se i valori sono primi
consecutivi, 1 se i valori non sono primi consecutivi */
    trov_prim = 0;

    /* verifica se i valori sono primi consecutivi */
    if (primo(val1) == 0 && primo(val2) == 0 && val1 < val2)
    {
        for (val2 = val2 - 1;
              (val2 > val1);
              val2-- )
        {
            if (primo(val2) == 0)
            {
                trov_prim = 1;
                val1 = val2;
            }
        }
    }
    else
        trov_prim = 1;

    return(trov_prim);
}

```

```

/* definizione della funzione per verificare la congettura di Cramer */
void cramer(int val1, /* input: primo valore */
            int val2) /* input: secondo valore */
{
    /* dichiarazione delle variabili locali alla funzione */
    int diff; /* output: valore assoluto della differenza tra i
               due numeri */
    double risult_log_esp; /* output: quadrato del logaritmo naturale del
                           piu' piccolo dei due numeri */

    /* verificare della congettura */
    if (primo_consec(val1, val2) == 0)
    {

        /* calcolo valore assoluto della differenza tra i due numeri */
        diff = abs(val1 - val2);

        /* calcolo quadrato del logaritmo naturale del piu' piccolo
        dei due numeri */
        risult_log_esp = pow(log(val1), 2);

        if (diff < risult_log_esp)
        {
            printf("Congettura verificata!\n");
            printf("|%d - %d| < (log %d)^2\n",
                    val1,
                    val2,
                    val1);
            printf("%d < %lf",
                    diff,
                    risult_log_esp);
        }
        else
        {
            printf("Congettura confutata!\n");
            printf("|%d - %d| > (log %d)^2\n",
                    val1,
                    val2,
                    val1);
            printf("%d > %lf",
                    diff,
                    risult_log_esp);
        }
    }
    else
        printf("I valori inseriti non sono due numeri primi consecutivi\n");
}

/* definizione della funzione per verificare la congettura di Collatz */
void collatz(int n) /* input: valore da verificare */
{
    printf("%d ",
            n);

    /* esecuzione congettura */
    if (n > 1)
    {
        if (n % 2 != 0)
            collatz(3 * n + 1);
        else
            collatz(n / 2);
    }
}

```

```

/* definizione della funzione per uscire dal programma */
char uscita_programma(void)
{
    /* dichiarazione delle variabili locali alla funzione */
    char exit;          /* output: valore di uscita */
    int esito_lettura,   /* lavoro: esito della scanf */
    acquisizione_errata; /* lavoro: esito complessivo
                        dell'acquisizione */

    /* leggere e validare l'uscita dal programma */
    do
    {
        printf("Vuoi uscire ? [y/n] ");
        esito_lettura = scanf("%c",
                               &exit);
        acquisizione_errata = esito_lettura != 1 ||
                               (exit != 'y' && exit != 'n');
        if (acquisizione_errata)
            printf("Valore non accettabile!\n");
        while (getchar() != '\n');
    }
    while (acquisizione_errata);
    if (exit == 'y')
        exit = uscita;

    return(exit);
}

```