

```
/*
 * Corso di Fondamenti di Informatica
 * Esercitazione 1
 * Esercizio 2:
 * programma che deriva da un processo di progettazione incrementale;
 * sfrutta le proprietà della conversione automatica di tipo da float a int
 */

/* Programma per convertire in formato binario numeri decimali compresi tra
 * 0 ed 1, overosia aventi dunque la forma 0.(...) dove (...) indica una o
 * più cifre dopo la virgola (rappresentata dal punto).
 * Il valore restituito viene troncato se il numero di cifre binarie dopo la
 * virgola supera quello specificato dal parametro MAX_NUM_CIFRE definito nel
 * seguito.
 */

/* Nota: le righe "commentate via" e precedute da "DEBUG" sono state usate
 * in fase di debugging e poi eliminate trasformandole in commenti. Sono
 * state lasciate nel programma a fini didattici. */

#include <stdio.h>
#define MAX_NUM_CIFRE 8
#define SOGLIA 1e-6

int main()
{
    float NumDaConv;
    /* il numero decimale compreso tra 0 (compreso) ed 1 (escluso) da
     * convertire in binario */
    float Doppio;
    /* contiene il doppio di NumDaConv (variabile non strettamente necessaria,
     * introdotta per rendere il programma piu' chiaro) */
    int ParteIntera;
    /* usato per estrarre la parte intera di Doppio sfruttando la conversione
     * automatica da float ad int (si veda il seguito) */
    int ContCifra;
    /* numero d'ordine della cifra (a destra della virgola) del numero binario
     * attualmente valutata */

    printf("Inserisci un numero decimale compreso del tipo '0.cifre decimali': ");
    scanf("%f", &NumDaConv);
}
```

```
/** inizio conversione ***/
printf("\n\nIl corrispondente valore binario e': 0.");
ContCifra = 1;

while ( (ContCifra <= MAX_NUM_CIFRE) && (NumDaConv >= SOGLIA) )
/* NOTA: se NumDaConv diventa nullo, overosia se la seconda delle due
 * condizioni in AND diventa vera, cio' implica che tutte le cifre
 * successive del numero binario saranno nulle ed è dunque inutile
 * calcolarle; per tale ragione se la condizione e' vera il ciclo viene
 * interrotto */
{
    Doppio = NumDaConv*2;

    ParteIntera = Doppio;
    /* sfruttiamo la conversione automatica di tipo per eliminare la parte
     * frazionaria (quella dopo la virgola) di Doppio */

    /* DEBUG: printf("\nDoppio=%f ", Doppio); */

    printf("%d", ParteIntera);
    /* stampa la cifra binaria corrente; notare che vale necessariamente 0
     * oppure 1 */

    NumDaConv = Doppio - ParteIntera;
    /* lascia solo la parte dopo la virgola; notare che se Doppio e' <1
     * NumDaConv coincide con Doppio */

    /* DEBUG: printf(" Ora NumDec=%f", NumDaConv); */

    ++ContCifra;
}

printf("\n\n");

return(0);
}

/* .....
 * Esercizio: chiedendo la conversione di 0.999999999999
```

- * il programma fornisce come risultato (binario) 0.2.
- * Come è possibile?
- * Suggerimento: il problema scompare se NumDaConv e Doppio sono definiti come double...
- *



*/

www.unidocs.it

www.unidocs.it

www.unidocs.it

www.unidocs.it

www.unidocs.it

www.unidocs.it

www.unidocs.it

www.unidocs.it