

```
//lista concatenata --> esercitazione
```

```
#include <stdio.h>
#include <stdlib.h>    //devo includere questa libreria per usare la CALLOC

//#define dimensione_lista 10

struct complex{        //Creo un TIPO    -->    un contenitore    -->    mi
    //permette di costruire un singolo numero complesso (parte reale + parte immaginari)

    float reali;
    float immaginari;
};

struct complex_list{    //Creo un TIPO    -->    mi permette di costruire una
    //lista di complessi

    struct complex numero;
    struct complex_list *next;    //qttraverso la lista in unico verso --> verso il
    //successivo
};

int main (void){

    int i = 0;    //questa variabile verrà usata nel ciclo per riempire la lista di
    //numeri complessi

    int dimensione_lista;

    printf("Quanti numeri?: ");
    scanf("%i", &dimensione_lista);

    struct complex_list lista_numeri;    //lista_numeri = variabile --> questo sarà il
    //primo elemento della lista

    //lista_numeri.numero.reali = i;    //prendo il valore di i e lo do alla parte
    //reale
    //lista_numeri.numero.immaginari = i;    //prendo il valore di i e lo do alla parte
    //immaginaria

    struct complex_list *coda = &lista_numeri;    //un puntatore alla struct
    //(all'intero blocchetto) --> punta all'indirizzo di lista_numeri
    //il puntatore (*coda) punta al 1 elemento della lista (&lista_numeri) --> coda sarà
    //usato per costruire la lista elemento per elemento

    //questo ciclo crea "dimensione_lista" (10) elementi della lista
    for( i = 0; i < dimensione_lista ; i++){

        //assegniamo alla parte reale e immaginaria del numero complesso il valore di i e
        //i

        coda -> numero.reali = i+1;    //num è un puntatore ( -> )    numero NON è un
        //puntatore ( . )
        // i + 1    -->    per stampare a partire dall'1
        // usiamo i per vedere che numeri sono (per visualizzare qualcosa)

        coda -> numero.immaginari = -(i+1);    //mettiamo -i solo per vedere una
        //differenza quando stampiamo, non per altri motivi
        // - ( i + 1 )    -->    per stampare a partire da 1 ma in
```

negativo

```

        //per allocare memoria dinamica tramite calloc

        // coda -> next      è il puntatore al prossimo blocchetto (il successivo)
        //collego il blocchetto con il blocchetto successivo -->      (perchè è un
puntatore al blocchetto Completo)

        struct complex_list *nuovo_blocchetto;      //dichiarazione di un puntatore per un
nuovo blocco della lista concatenata --> ccrea un nuovo nodo della lista
        nuovo_blocchetto = (struct complex_list *) calloc (1, sizeof(struct
complex_list)); //alloco spazio per un nuovo blocchetto completo (include anche il
puntatore al blocchetto successivo)
        coda -> next = nuovo_blocchetto;      //collega il nodo corrente (coda) nel nuovo
nodo      -->      coda -> next ora punta a "nuovo_blocchetto"

        //...sino a qui

        coda = coda -> next;      //così facendo, al prossimo giro del ciclo for, possiamo
aggiungere un altro blocco ancora

    }

    coda = &lista_numeri;      //per resettare coda      -->      coda ora punta al 1
indirizzo      --> così possiamo scorrere nuovamente coda (usando next)
    // coda è il 1 blocchetto ora      (puntatore)

    while( coda -> next != (void*) 0 ){      // (void*)0      =      puntatore nullo --> ha
l'indirizzo 0

        printf("%.2f + %.2fi\n", coda -> numero.reali, coda -> numero.immaginari);

        coda = coda -> next;

    }

    return 0;
}

```