

23/1/2019 ftp://ftp.elet.polimi.it/users/ALESSANDRO.GABRIELLI/Didattica/Fondamenti\_di\_informatica/2018\_2019/laboratori/Lab5/codice\_lab05/lab...

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

typedef struct EL
{
    int      Dato;
    struct EL* Prossimo;
} TipoElemento;
/* tipo del singolo elemento della lista dinamica usata nel programma */

typedef enum {false, true} boolean;

TipoElemento* Riempilista(void);
/* crea una lista dinamica contenente interi positivi o nulli letti da
 * tastiera e restituisce la testa di tale lista. L'input da tastiera termina
 * se l'utente inserisce un numero negativo. */

void StampaLista(TipoElemento* Testa);
/* stampa a schermo gli interi contenuti nella lista avente testa Testa */

boolean ContieneDoppioni(TipoElemento* Testa);
/* restituisce true se almeno uno dei numeri contenuti nella lista avente testa
 * Testa compare due o più volte nella lista, altrimenti restituisce false */

int main(void)
{
    TipoElemento* TestaListaInteri;

    printf("\n\n");
    TestaListaInteri = Riempilista();

    printf("\n\n");
    StampaLista(TestaListaInteri);

    printf("\n\nLa lista ");
    if (false == ContieneDoppioni(TestaListaInteri))
    {
        printf("non ");
    }
    printf("contiene doppioni.");

    printf("\n\n");

    return(0);
}

TipoElemento* Riempilista(void)
{
    TipoElemento* Testa;
    /* la testa della lista che la funzione crea */
    int Input;
    /* numero intero letto di volta in volta da tastiera */
    TipoElemento* NuovoElemento;
    /* punta ad un nuovo elemento della lista appena allocato */

    Testa = NULL;

    printf("Inserisci una serie di interi positivi o nulli; per terminare inserire un intero
negativo.\n");
    do
    {
        printf("Inserire un numero: ");
```

```
23/1/2019      ftp://ftp.elet.polimi.it/users/ALESSANDRO.GABRIELLI/Didattica/Fondamenti_di_informatica/2018_2019/laboratori/Lab5/codice_lab05/lab...
scanf("%d", &Input);

if (Input >= 0)
    /* aggiunta di un nuovo elemento in testa alla lista */
    {
        NuovoElemento = malloc(sizeof(TipoElemento));

        NuovoElemento->Dato = Input;
        NuovoElemento->Prossimo = Testa;

        Testa = NuovoElemento;
    }
} while (Input >= 0);

return(Testa);
}

void StampaLista(TipoElemento* Testa)
{
    TipoElemento* Cursore;
    /* scorre gli elementi della lista */

    printf("Stampa degli elementi della lista:");

    Cursore = Testa;
    while (Cursore != NULL)
    {
        printf("\n%d", Cursore->Dato);
        Cursore = Cursore->Prossimo;
    }
}

boolean ContieneDoppioni(TipoElemento* Testa)
{
    TipoElemento *Cursore, *Cursore2;
    /* scorrono gli elementi della lista */

    Cursore = Testa;
    while (Cursore != NULL)
    {
        Cursore2 = Testa;
        while (Cursore2 != NULL)
        {
            if ((Cursore2 != Cursore) &&
                (Cursore2->Dato == Cursore->Dato))
                /* se esiste un doppione */
                {
                    return(true);
                }

            Cursore2 = Cursore2->Prossimo;
        }

        Cursore = Cursore->Prossimo;
    }

    return(false);
}
}
```