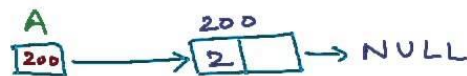
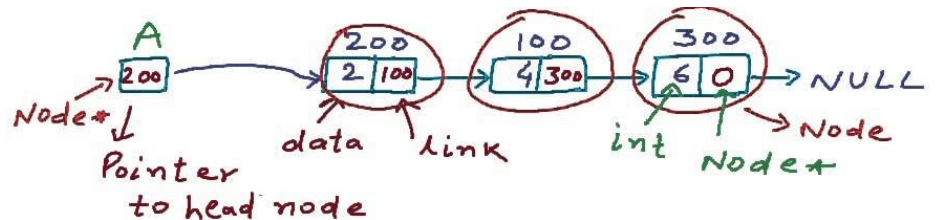


ESERCIZI SULLE LISTE

in C

```
Struct Node
{
    int data;
    Node* link;
}
Node* A;
A = NULL;
Node* temp =
    (Node*)malloc (sizeof(Node))
(*temp).data = 2; +
(*temp).link = NULL;
A = temp;
```



INDICE

1. Invertire una lista.....	3
2. Fusione di due liste ordinate	5
3. Media di una lista e nodi maggiori della media	7

1. Invertire una lista

Leggere una lista e costruire una seconda lista con gli elementi disposti al contrario rispetto alla lista originale (es.: se la lista letta in input è $1 \rightarrow 8 \rightarrow 4$ viene costruita la lista $4 \rightarrow 8 \rightarrow 1$).

```
#include <stdio.h>

#include <stdlib.h>

struct nodo
{
    int info;
    struct nodo *next;
};

struct nodo *leggi_lista(void)
{
    struct nodo *p, *primo=NULL;
    int i, n;
    printf("Numero di elementi: ");
    scanf("%d", &n);
    for (i=0; i<n; i++)
    {
        printf("Inserisci elemento %d: ", i);
        p = malloc(sizeof(struct nodo));
        scanf("%d", &p->info);
        p->next = primo;
        primo = p;
    }
    return(primo);
}
```

```

void stampa_lista(struct nodo *p)
{
    while (p != NULL)
    {
        printf("%d --> ", p->info);

        p = p->next;
    }

    printf("NULL\n");

    return;
}

struct nodo *inverti(struct nodo *p)
{
    struct nodo *q, *q1 = NULL;

    while (p != NULL)
    {
        q = malloc(sizeof(struct nodo));

        q->info = p->info;

        q->next = q1;

        q1 = q;

        p = p->next;
    }

    return(q1);
}

int main(void)
{
    struct nodo *p, *q;

    p = leggi_lista();

    stampa_lista(p);

    q = inverti(p);

    stampa_lista(q);

    return(0);
}

```

2. Fusione di due liste ordinate

Lette in input due liste di numeri interi ognuna delle quali ordinata, costruire una terza lista di numeri interi ordinata, ottenuta mediante la “fusione” delle prime due.

```
struct nodo *fusione(struct nodo *p1, struct nodo *p2)
{
    struct nodo *p, *primo = NULL;

    while (p1 != NULL && p2 != NULL)
    {
        p = malloc(sizeof(struct nodo));

        p->next = primo;

        primo = p;

        if (p1->info > p2->info)
        {
            p->info = p1->info;

            p1 = p1->next;
        }
        else
        {
            p->info = p2->info;

            p2 = p2->next;
        }
    }

    while (p1 != NULL)
    {
        p = malloc(sizeof(struct nodo));

        p->info = p1->info;

        p->next = primo;

        primo = p;

        p1 = p1->next;
    }
}
```

```

while (p2 != NULL)
{
    p = malloc(sizeof(struct nodo));

    p->info = p2->info;

    p->next = primo;

    primo = p;

    p2 = p2->next;
}

return(primo);
}

int main(void)
{
    struct nodo *p, *q, *r;

    p = leggi_lista();

    q = leggi_lista();

    r = fusione(p, q);

    stampa_lista(r);

    return(0);
}

```

3. Media di una lista e nodi maggiori della media

Creare una lista di nodi *floating point*. Costruire una seconda lista composta dai soli nodi maggiori della media.

```
#include <stdlib.h>

#include <stdio.h>

struct nodo
{
    float info;
    struct nodo *next;
};

struct nodo *leggi_lista(void)
{
    struct nodo *p, *primo;
    int i, n;
    printf("Inserisci il numero di elementi: ");
    scanf("%d", &n);
    primo = NULL;
    for (i=0; i<n; i++)
    {
        printf("Inserisci elemento %d: ", i);
        p = malloc(sizeof(struct nodo));
        p->next = primo;
        scanf("%f", &p->info);
        primo = p;
    }
    return(primo);
}
```

```

void stampa_lista(struct nodo *p)
{
    while (p != NULL)
    {
        printf("%3.2f --> ", p->info);

        p = p->next;
    }

    printf("Null\n");

    return;
}

```

```

float media(struct nodo *p)
{
    float s=0.0, n=0.0;

    while (p != NULL)
    {
        s = s + p->info;

        n = n + 1;

        p = p->next;
    }

    return(s/n);
}

```



```

int main(void)
{
    struct nodo *p, *q, *r, *s;

    int conta = 0;

    float m;

    p = leggi_lista();
    stampa_lista(p);
    m = media(p);
    r = NULL;

    q = p;
    while (q != NULL)
    {
        if (q->info > m)
        {
            conta++;

            s = malloc(sizeof(struct nodo));

            s->info = q->info;

            s->next = r;

            r = s;
        }

        q = q->next;
    }

    printf("\nElementi maggiori della media (%d) \n", conta);
    stampa_lista(r);
    return(0);
}

```