

typedef

```

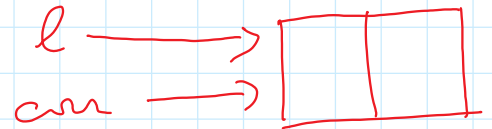
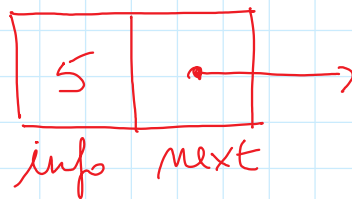
struct node
{
    int info;
    struct node * next;
}

```

Tipo

Elemento di lista ;

) nome dato al tipo



main ()

```

{ ElementoDiLista * l = NULL, con;

```

```

    int n, i;

```

```

    scanf ("%d", &n);

```

```

    if (n > 0)

```

```

    { l = malloc (sizeof (ElementoDiLista));

```

```

        l -> info = 1;
        con = l;

```

```

        for (i = 2; i <= n; i++)

```

```

        { con -> next = malloc (sizeof (EDL));

```

```

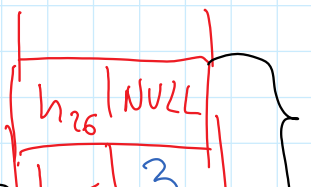
            con = con -> next;

```

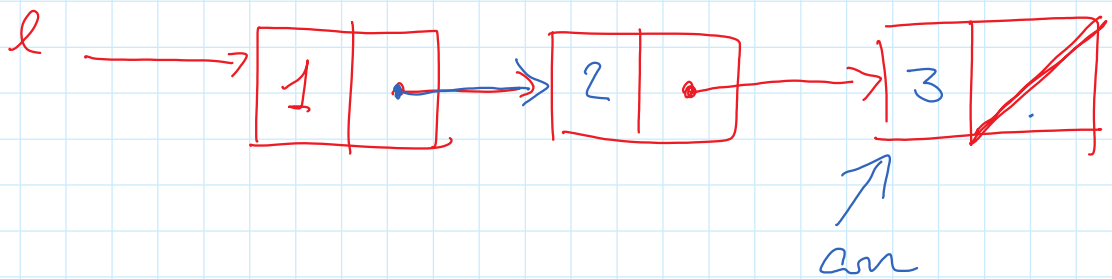
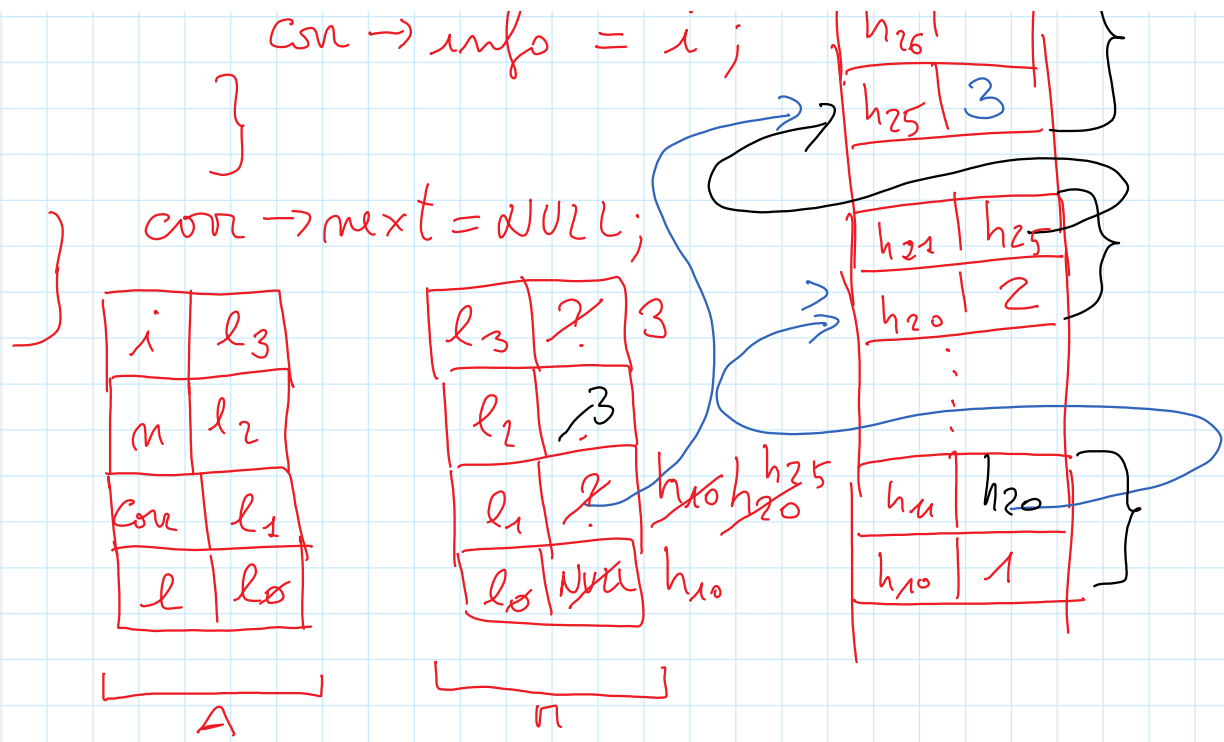
```

            con -> info = i;

```



l == NULL vuol dire che la lista puntata da l è vuota



```

void crealista (ElementoDiListe **l, int n)
{
  ElementoDiListe * con; int i; *l = NULL;
  if (n > 0)
  {
    *l = malloc (sizeof (ElementoDiListe));
    con = *l;
    con -> info = 1;
    for (i = 2; i <= n; i++)
    {
      con -> next = malloc (sizeof (EDL));
      con = con -> next;
      con -> info = i;
    }
    con -> next = NULL;
  }
}

```

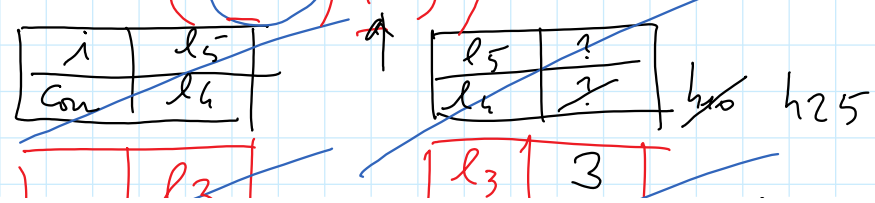
l ha tipo

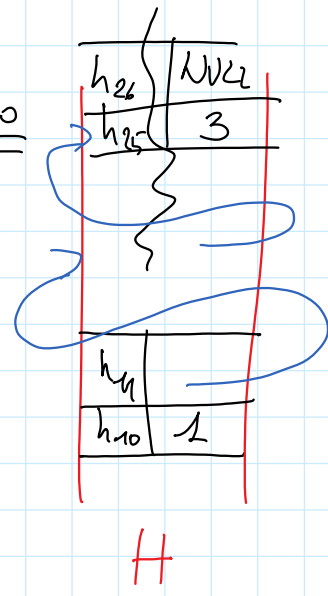
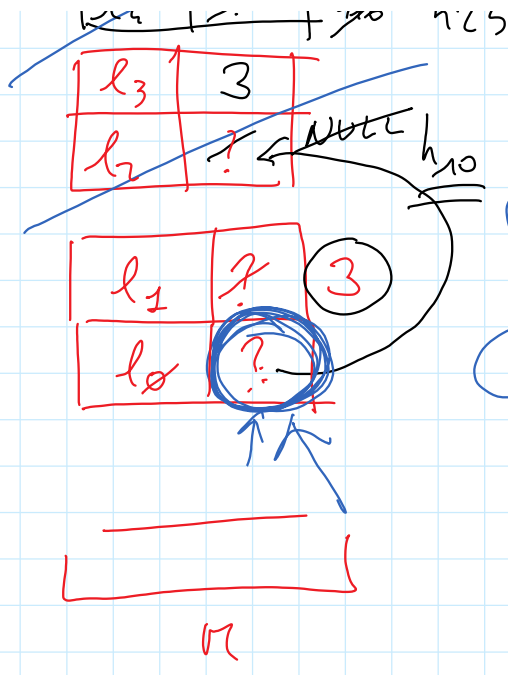
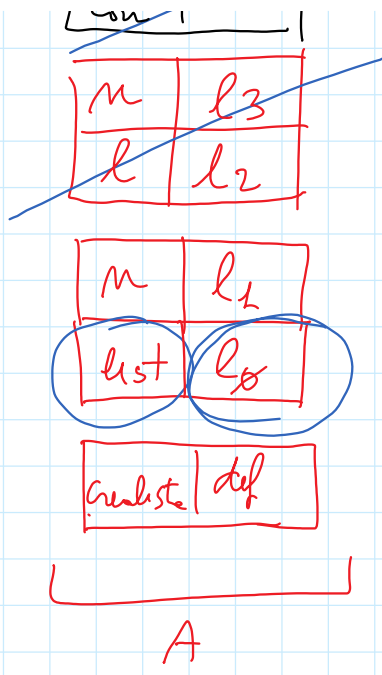
```

main()
{
  ElementoDiListe * list;
  int n;
  scanf ("%d", &n);
  crealista (list, n);
}

```

&list



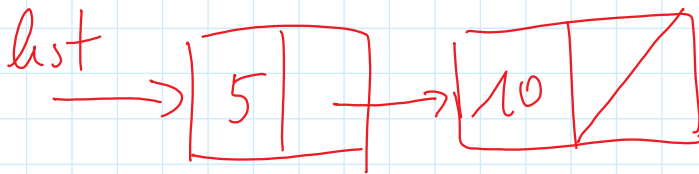
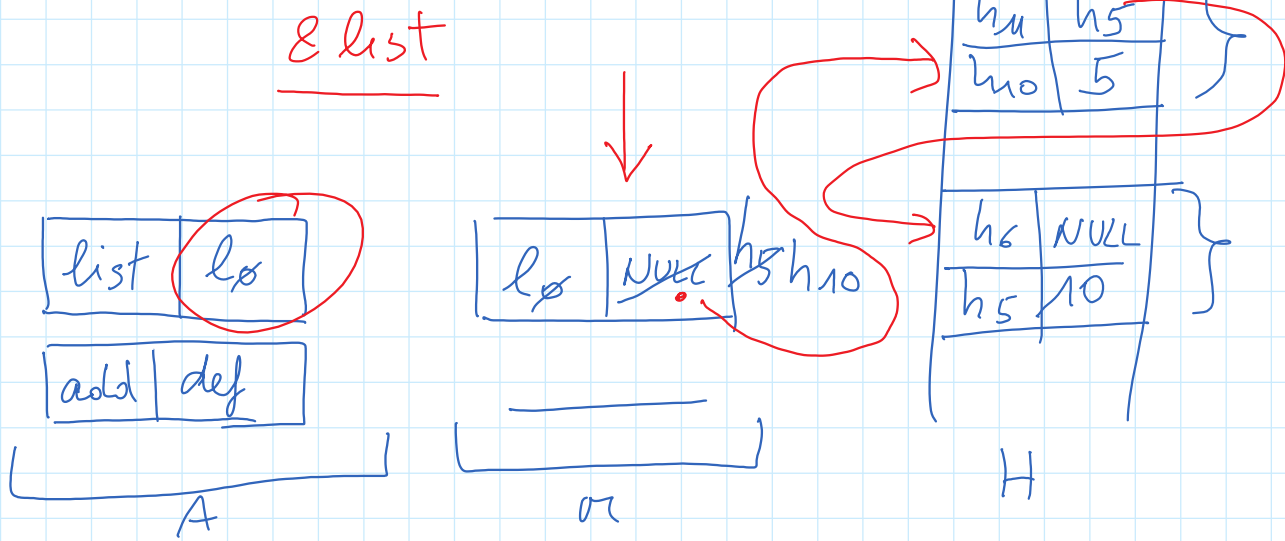


Si scriva una procedura che aggiunge un elemento in testa a una lista

```
void add (ElementoDiLista ** l, int m)
{
    ElementoDiLista * new = malloc (sizeof (ElementoDiLista));
    new -> info = m;
    new -> next = * l;
    * l = new;
}
```

```
main ()
{
    ElementoDiLista * list = NULL;
    add (&list, 10);
    add (&list, 5);
}
```

Elementi di Liste **



list

Elementi di Liste *

& list

Elementi di Liste **

Scrivere una funzione che aggiunge un elemento in testa a una lista e restituisca il puntatore al primo elemento della nuova lista.

```

ElementoDiLista * add (ElementoDiLista * l, int n)
{
  ElementoDiLista * new = malloc (sizeof (ElementoDiLista));
  new -> info = n;
  new -> next = l;
  return new;
}

```

main()

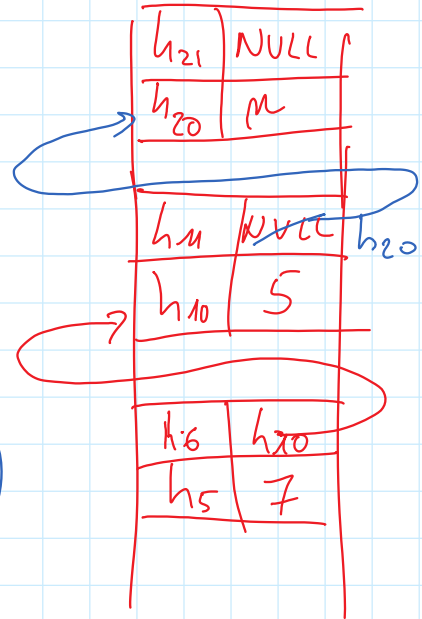
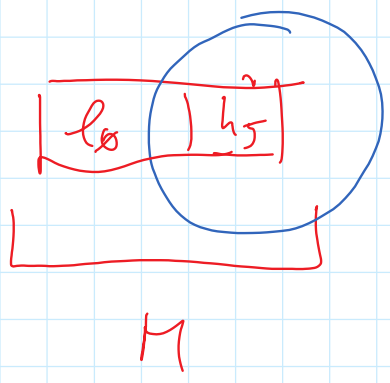
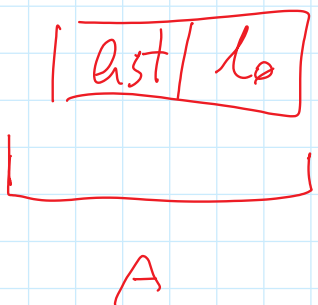
```

{
  ElementoDiLista * list = NULL;
  list = add (list, 10);
  list = add (list, 5);
}

```

mercoledì 7 novembre 2018 12:41

Aggiungere la struct, elemento vuoto
void addlast (Elemento di lista * l, int n)




```
void addlast (Elementi Di Liste * l, int n)
```

```
{ Elementi Di Liste * new = malloc (sizeof (EDL)),
```

```
new -> info = n; cor; }
```

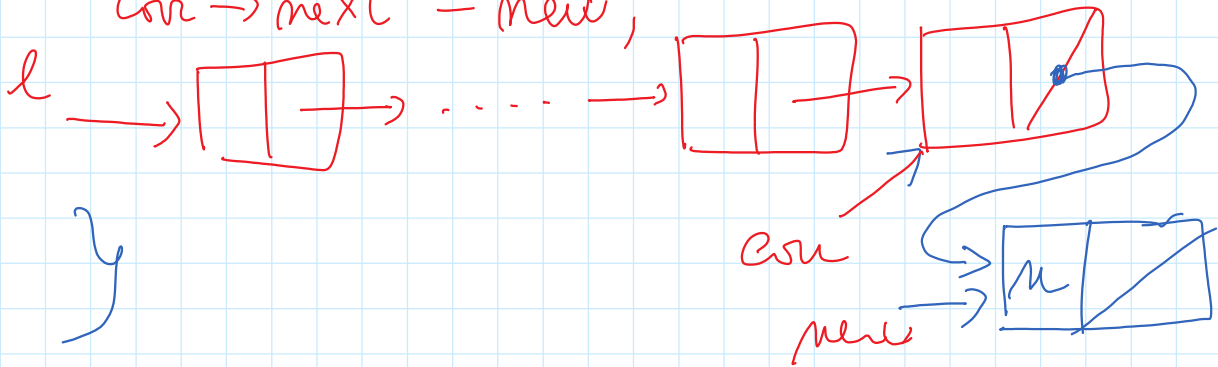
```
new -> next = NULL;
```

```
cor = l;
```

```
while (cor -> next != NULL)
```

```
cor = cor -> next;
```

```
cor -> next = new;
```



```
void add_list (Elemento Di lista ** l, int m)
```

la lista può essere vuota!

```
{ Elemento Di lista * new = malloc ( ..... );
  new -> info = m;
  new -> next = NULL;
```

```
if (*l == NULL)
    *l = new;
```

```
else,
```

```
{ Elemento Di lista * cur = *l;
  while (cur -> next != NULL)
      cur = cur -> next;
```

```
  cur -> next = new;
}
```

```
}
```