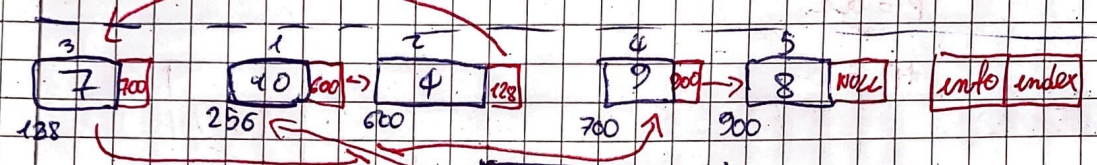


10, 4, 7, 9, 8

Head = 256

Fiecare element va avea 2 componente {info de tip T, link de tip T* -> memorează adresa următorului element din listă

[ULTIMUL ELEMENT ARE LINK CĂTRE POINTERUL NULL. {Nu punctează}]

Head: Pointer = adresa primului element din listă

info	link

```
struct Nod {
    T info;
    Nod* link;
};
```

struct -> ne permite să grupăm mai multe tipuri de date sub un același nume.

```
struct Data {
    int zi;
    int luna;
    int an;
};
```

în C -> variabile de tip data, d ->
-> struct Data d;

în C++ -> Data d;

d

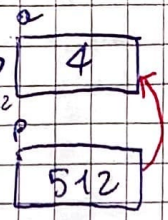
p -> | 11 (zi) | 10 (luna) | 2022 (an) |

d.zi = 11;
d.luna = 10;
d.an = 2022; Data *p;
p = &d; adresa lui d

p->zi este de tip int -> este 11

Pointeri

```
int a = 4;
int *p;
p = &a
```



[ADRESA LOCATIEI SE ACCESEAZĂ: P' sau &a]

```
cout << "a = " << a << endl   // a = 4
cout << "*p = " << *p << endl   // *p = 4
```

↳ conținutul zonei de memorie către care pointează p

p = &a
≠
*p = a

int *p	int *p
p = &a;	*p = a;
p → [4] (a)	[4] (a) p → [4]

NO!

```
int a = 2, b = 3;
int *p, *q;
```

p = &a; [300] p → [2] (a, 400)

q = &b; [300] q → [3] (b, 300)

p = q; Atribuire de pointeri // p și q pointează în același loc

cout *p = 3	→ *p = 9 atribuire	→	cout *p = 9
cout *q = 3			cout *q = 9

(*p)++ ⇒ *p = *p + 1 ⇒ 9 + 1 = 10

p++ // p = p + sizeof(int)

cout << *p // valoare reziduală

cout << *q // *q = 10

Lista în alocare înlănțuită

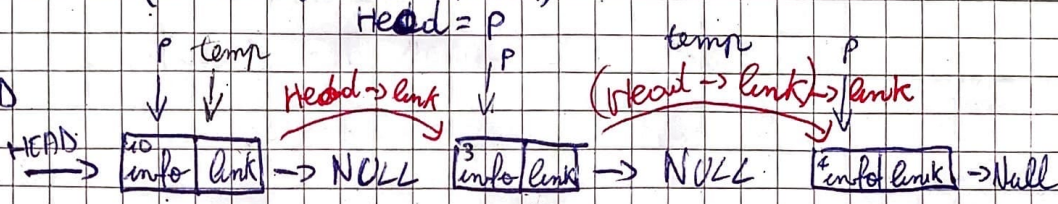
HEAD -> [10] ——> [3] ——> [4] ——> NULL

Crearea listei

1. // alocă zonă de memorie pentru primul element. Fie p adresa zonei de memorie

HEAD ⇒ p (în C++: new Nod;
Head = p

Nod* HEAD



HEAD -> info = 10; HEAD -> link = NULL
(Este echivalent cu p -> link = Null)

Alocă o nouă zonă de memorie pentru un nou Nod
Fie p pointer la zona de memorie
(p = new Nod // C++)

Pas următor la fel

p -> info = 3;
p -> link = NULL;
} Head -> link = p

p = new Nod

p -> info = 4
p -> link = NULL
} (Head -> link) -> link = p
temp -> link = p

temp = p // temp memorează poziția ultimului Nod creat

// Să se creeze o listă alocată înlănțuit cu 10 elemente introduse de la tastatură.

```
Nod *Head, *p, *temp=Null; Head = Null;
for (int i=1; i<=10; i++){
    p = new Nod;
    cin >> p->info
    p->link = Null;
    if (Head == Null) {
        Head = p;
    } else {
        temp -> link = p
    temp = p;
}
```

!!

// Să se afișeze o listă sub forma 3 -> 4 -> 8 -> Null
(Nu știm décât Head, nu știm câte elemente)

```
Nod *p = HEAD;
while ( p -> link ≠ NULL){
        cout << p -> info << "->";
        p = p -> link
}
cout << "NULL" << endl;
```

Var 1

```
for (Nod * iter = HEAD; iter != NULL; iter = iter -> link){
        cout << iter -> info << "->";
}
Cout << "NULL" << endl;
```

Var 2