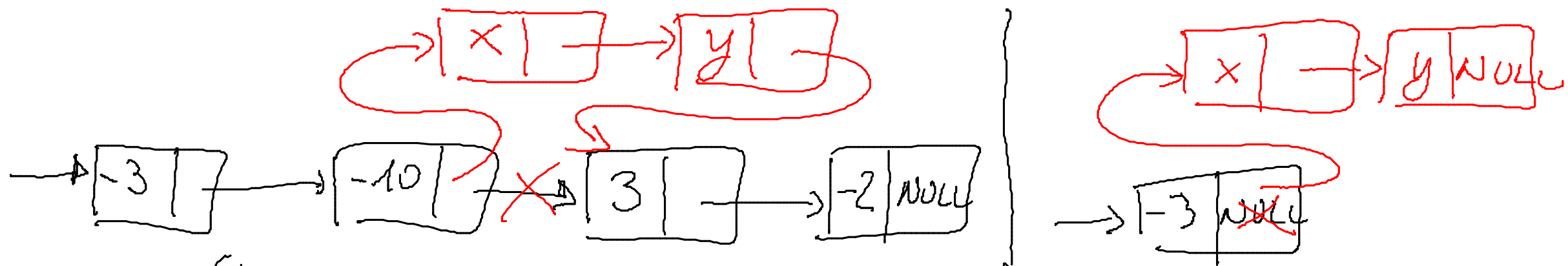


Scrivere una procedura C che presi due parametri x e y e una lista di interi, inserisca x e y immediatamente prima del primo valore positivo nella lista.

Se la lista non contiene valori positivi x e y vanno inseriti in coda alla lista.



```
void ms (ListoElement, * l, int x, int y)
```

```

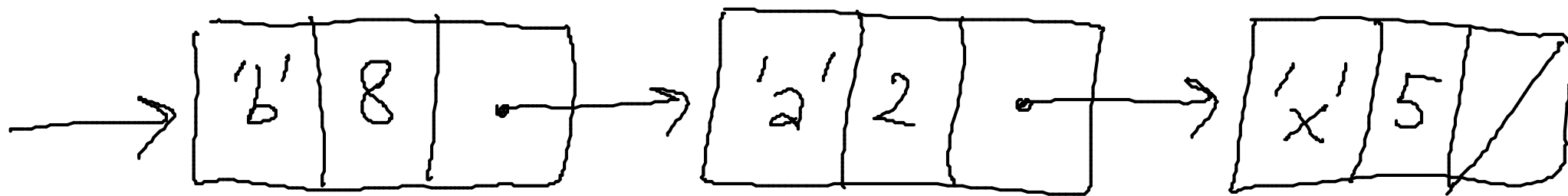
void ins (ListDiElementi * l; int x, int y)
{
  ListDiElementi aus1 = malloc(...); ListDiElementi aus2 = malloc(...);
  aus1->info = x; aus2->info = y; aus1->next = aus2;
  if (*l == NULL) { *l = aus1; aus2->next = NULL; }
  else if (*l->info > 0) { aus2->next = *l; *l = aus1; }
  else { ListDiElementi prec = *l; ListDiElementi cor = *l->next;
        int trovato = 0;
        while (cor != NULL && !trovato)
          if (cor->info > 0) trovato = 1;
          else { prec = cor; cor = cor->next; }
        prec->next = aus1;
        aus2->next = cor;
  }
}

```

```
struct el { char codice; int quantita;  
            struct el * next; }
```

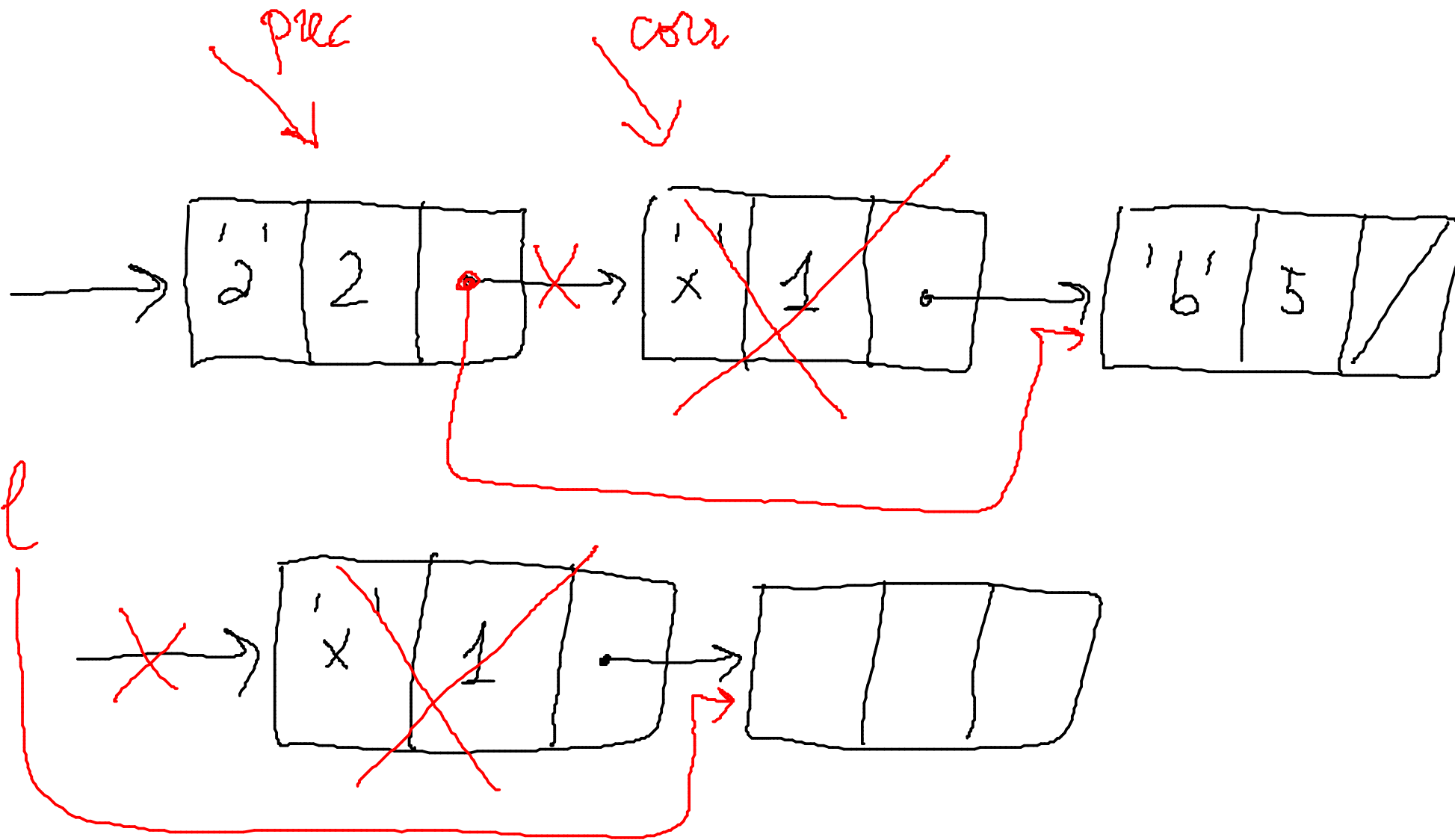
```
typedef struct el Prodotto;
```

```
typedef Prodotto * Magazzino;
```



Scrivere una procedura che dato un magazzino e dato un prodotto aggiorna il magazzino a seguito della vendita di una unità del prodotto

void remove (Magazzino *m, char c)



```
void vendita (Magazzino *m, char c)
```

```
{ Magazzino prec, corr; int trovato = 0;
```

```
prec = NULL; corr = *m;
```

```
while (! trovato)
```

```
    if (corr -> codice == c)
```

```
        trovato = 1;
```

```
    else { prec = corr; corr = corr -> next; }
```

```
    corr -> quantita = corr -> quantita - 1;
```

```
    if (corr -> quantita == 0)
```

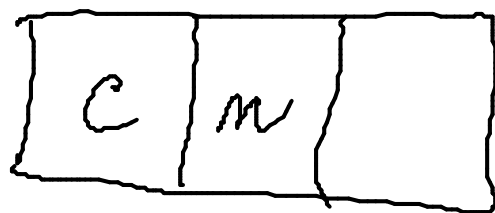
```
        { if (prec == NULL) *m = corr -> next;
```

```
          else prec -> next = corr -> next; free (corr); }
```

```
}
```

Scrivere una procedura che, dati un magazzino m
un codice prodotto c e una quantità n , aggiunge
al magazzino n elementi del prodotto c

- 1) se c non è presente in m , bisogna
"aggiungere a m un elemento



- 2) se c è già presente in m →

| | | |
|---|--------------|---|
| c | k | → |
|---|--------------|---|

void aggiungi (Magazzino *m, char c, int n)

```
void aggiungi (Magazzino *m, char c, int n)
```

```
{ Magazzino corr = *m; int trovato = 0;
```

```
  while (corr != NULL && !trovato)
```

```
    if (corr->codice == c) trovato = 1;
```

```
    else corr = corr->next;
```

```
  if (trovato) corr->quantita = corr->quantita + n;
```

```
  else { Magazzino new = malloc (sizeof (Prodotto));
```

```
    new->codice = c; new->quantita = n;
```

```
    new->next = *m;
```

```
    *m = new; }
```

```
}
```