Consider a = [1, 2, 3]
b = [4, 5, 6] and merge a and b
and get c = [1, 2, 3, 4, 5, 6]

a = size(a) / size of (int)

b = size(b) / size of (int)

```
int a[ ] = {1, 2, 3}
int b[ ] = {3, 4, 5}
int c[ a+b ]

for ( i=0 ; i<a ; i++)
{
  c[i] = a[i]
}
for (j=0 ; j<b ; i++)
{
  c[a+j] = b[j]
}
For (k=0 ; k<(a+b) ; k++)
{

}
```

9003093079

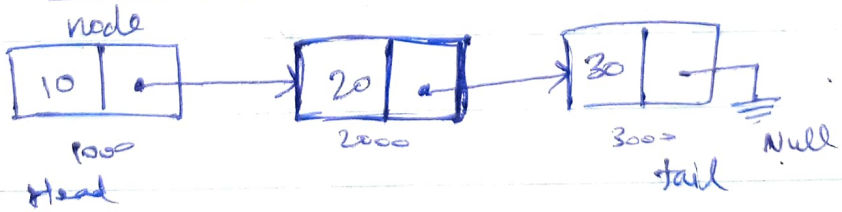
a: 

for (i=0; i<10; i++)

Get a input from the user using dynamic allocation at runtime for n=1

```
int main()
{
int *ptr = malloc (N * sizeof(int));
for (i=0; i<N; i++)
{
  scanf ("%d", ptr+i);
}
  printf ("%d", *(ptr+i));
```

List using linked list




```
Struct node
{
  int data;
  Struct node *link;
};
```

```
#include <stdlib.h>
Struct node
{
 int data;
 Struct node * ptr
};
int main()
{
Struct node * head = malloc (size of (struct n
//      Head -> 10
Struct node * tail = malloc (size of (struct node))
    tail -> data = 20;
    tail -> ptr = Null
    Head -> ptr = Tail

Struct node * temp = malloc (size of (struct node));
      tail -> data = 30;
      tail -> ptr = Null
```

$$F(n) = \begin{cases} T(n/2) + 1 & n > 1 \\ 1 & n == 1 \end{cases}$$

Find TP ?

$$T(n) = T(n/2) + 1$$

$$= T\left(\frac{n/2}{2}\right) + 1 + 1$$

$$= T(n/4) + 2$$

$$= T\left(\frac{n/4}{2}\right) + 2 + 1$$

$$= T(n/8) + 3$$

$$\vdots$$

$$= T(n/2^3) + 3$$

$$= T(n/2^k) + k$$

$$n/2^k = 1$$

$$n = 2^k$$

$$k = (\log_2 n)$$

$$T(n) = \begin{cases} T(n/2) + n^2 & n > 1 \\ 1 & n == 0 \end{cases}$$