\* Raw text to - Component Papagraphs , Sentences & words . Struck passagraph { Struct word { struct sentence { Struck Gentence \* data; chaon \* data; Struck word \* data; Pot Sentence\_count; 9nt Cword-Count; Struct document { Learning ( 26 fun. Struct passagraph \*data; Leconning pointers les mosse fun. It so good to have pointers Int panagraph - count; Struck Sentence\_ forst\_ Sentence\_ fn\_first\_ Panagraph; frost \_ sentence \_ 9n\_ first \_ pacagraph.data = { executing ? rec", ee 26", edata" }; Struck panagraph first - Panagraph; 2 first - panagraph. data = { { Learning ? .... ? fun ? } }; 1) Skruck genkence first\_sentence\_9n\_second pagraph; forst\_sentence\_gn\_second\_pagraph.data= { eleanning", "pointeens, ... fun; 3; 1) Struck Berkence Second\_Sentence\_9n\_Second\_Pagraph; Second\_Sentence\_In\_Second\_Pagraph.data=gerIt", ..., epounteres"; Second Paragraph. data = { { " Learning", "pointers", "fun" y, " etu", " pointers" }; @ Struck & ocument doc; doc. data = { { { e Learning ? ... "fun" } }, { { e Learning ... }, { e It" ... } }; Same thing - chan\* Learning C % fun. In Learning pointers es more fun. It les good to have C. Apponoach: Just give addonoses

Sentence - going to have -> Strings!

```
* Void Installed _ document (chan * text) - Installe Doc - Struct document
* Struck passagraph Kth - place agraph (Pot K) -> sneturn Kth passa
* Struck Sentence Kth_ Sentence _9n_mth_ Poonagraph (9nt k, 9nt m) -> thought
               in an passa!
 Struck Word Kth_ word_ 9n_m_th_Sentence_Of_h_th_pooragraph (9nt k,
                          9nt m, 9nt n); - 1 kth word, mth Sent, hth passa.
   Franagraph Count.
                        kext (chan *)
       Vuentes - number.
            1 K: Kth paora
               k m: Kth Sentence in the powa
               K m n: k th word mth about n th passa.
                    Kth - Panagraph (Struck document Doc, Int K)
 @ Struck Renkence Kth-Sent - m-th- Foona (Struct document Doc, 9nt Ky Potm)
   greaum (Doc. data [M-1]. data [M-1]);
3 Struck word Kth_word_Pn_mth_Sentene_Of_nth_pagna (Struct
                                 Pot Ka Pot ma Pot n
     greturn (Doc. data [m-1] . data [m-1] . data [k-1]);
                              c > next blood
                                  - nex Sontence
```

Last line of the papa (lost) doesn't have \n.

in next passa.

```
- Coneate word
           6. ? - Coreate Sentence, coreate word
10 whenever (10?, --- cheate pana, Sentence, word,
            - no reed goon new dentence - coreate new paina
                             -> e10? -> end of string!
        make make
                         Same as penevious one! - do with Struct
                            atruck document
                                              Doc;
 foroblem faced:
                           Rook cause !!
 and respect orner subside
why? : Cgreated in stack - dies when function gets done!
Solution: Use pointon à dynamic memory allocation!
        document get-document (chaos *text) {
 Struck
        9nt papa = 1, Sent = 1, words = 1, 9;
         okruck document * Doc;
         Doc = (Struck document *) malloc (38280) (Struck document));
         Doc - data = (Struck passagraph *) malloc (3820 of (Struct passagraph));
         Doc - passagraph_count = 1;
         Doc - data [0] . data = (struct Sentence *) malloc (size of (struct
                                                     Sentence));
         Doc - data [o]. Sentence _ count = 1;
         Doc - data [o]. data [o]. data [o]. data = (chaon *) kext;
         poc - data [o]. data [o]. word_count = 1;
         fog( ( = 0; Lext [ + 1] ) = ( ) ; 9++)
           (kex + [9+1] == (n2)
            { text[1] = 610;
```

```
Switch (rext [9])
     case:
           Doc - data = (Struck passagraph*) 91ealloc (Doc - data 1 passa * 512e of
            pagra ++;
                                                         (Struct paragraph));
            Sent=0;
             Doc → data [pana-i], Sentence _ Count =0;
      Case
              Sent++; (Doc-) data [Paona-1). Sentence_count)++;
              (Doc - data [Paga - 1]. data = (Struck Sentence *) greavoc
                  (DOC-) data [paona-i]. data, Sent * SizeOf (struct Sentence));
              Doc-data [pana - 1]. data [sent -1]. word _count = 0;
              words = 0;
      Case
              words ++;
              (Doc -) data [paga - 1]. data [sent -1]. word_coupt) ++;
               Doc - data [paga - 1). data [sent -1]. data = (struct word +)
                   greater (Doc -) data [passa - 1).data [sent -1].data,
                                  words * size of (Struck Word));
               Doc-data [pana-1).data [sent-1].data [words-1].data =
                              (chaon *) & text [9+1];
  : ((60,000 text [7] = 6,0, ; (00000) - 0000 = [0] 0300 = 0000
                               some a gast [a] & source - come
text[7]= 101; 2010) = arch. (d) Auch. (d) prob. (d) each.
return (Doc.data [n-1].data [m-1].data [x-1]);
```

use: document \* Doc

Then allocate Pro heap?