```
1. Stack Implementation using Linked List
                                                    29/01/20,
Hinchnotest dio. hs
.# include (stdlib.h)
                                Long-serge serge
Stand Node {
  int data;
 Stanit Node & nent;
Stand Stack {
  Struct Node a top;
  85
Struct Node " create Node(int data) {
    stant Node * ohn Node = (strut
   Node*) malloc (sørze of (stmil Node));
       new Node -> data: data;
       new Node -> nent = NULL;
     retnen new Node;
Struct Stack " create Stack(){
   Stmit Stack * Stack : (stmit
 Stack ") malloc (saigo (st mil Stack));
     Stack -> top= NULL;
  return stack;
  int is Empty (stouct Stack * stack) {
   return stack -> top== NULL,
  Void push (stmit Stack * stack, intolata) {
   Stomet Node nen Node = coreale Node (donta);
    nenoNode -> nent = stack -> top;
  stack -> top = newNode;
 int pop(stanct Stack stack) {
      if ( is Empty (stack)) {
```

prille return -1; stand Node popped Node = stack - stop; int popped Data = popped Node - odata; Stack - top = popped Node - nent; force (popped No de); SLL Stender, Burnel Coveredor return popped Data; int peek (stant Stack stack) { return is Empty (stack)?-1: stack -> top-> data; 2/3 void display (stonit Stack stack) englism. stmit rode " corrent = stack -> top; while (convent != NULL) { printf (" fod", corrent - data); Greek impless current = current -> nent; ey corpses sprojection printf("/m"); At forchide cetallite is Strong mode? int main(){ stonct Stack stack = (neateStack(); Frit dista Wirnet Nod prohlstack,1); Storest Girens E proh(stact, 2); Browet Node a front: prohlotack, 3); Storet Wede orear. printf("Stack:"); stored node . on o'a Ned display (stack); printf (" Peek: Tod", peek (stack)); printf ("Pop: 7. dinspop(stack)); Node ) malloc (. 200) printf ("Stack after pop: "); news de -> news display(stack); etwen maple returno;

strict Qu Chtput: Exter 1. Push 2. Pg) 3. -1 to stop stmo : Enter operation. some stades, spood Breezewa qu 1. Enter dement to push 23 Stack elements are: 23

Enter operation:

Enter operation: 45

Enter Element to prosh: 45

Stack Elements are: 45 23 re int is void e Enter operation: 2 1 great ( stand Stad " street) } { s popped Element: 45 23. atsbegate tosts 186 Stack Elements are 1 Happy Esternit Steel to stock Jel 2. Quene implementation naing hinkedhirt 3 #include < stdio. h> int # include < Stalib. h> 16 Struct Node { int data; > stonet Node = nent; stones starter 1 3 brigg ( ctart 1); str Struct Quene { prohistal, 2): Struct Node = front; publistack, 3); Struct No de & rear; " Stack: Strutt Node « create Node (int data) { (7) +52 long dsg. Struct Node\* nenvode-Estonet Node\*) malloc (sigeof (stmct Node)) int Signaf ( Lab : N. greet new Node - data datas energy ( Stock ofter nersono de -> nent = NULL; 3 o shoul (stack)? return new Node? Voi ( a morals.

struct Grand stmit Quene quene - (stomet Quene ) malloc(siger quelle sfront = quene > rear = NULL; retran quene int is Empty (storet Quene of greene) Ereturn quene front ==NULL;{ void enquene (stmet anene quene, int data). { stond Node new Node = create Node (data); if list mpty (quenes) { quene -> front = quene -> rear = New No de ; quene -> rear -> nent = new Node; quene -> rear = New Node; int dequene (stonct anene + quenel if lis Empty (quene)) { printf ("Quene un derflow"); 7 retran -1; storet Node \* dequened Node = quene > front; int dequened Data = dequened No de -> data; quene - front : deg nened Node -> nent; free (dequene d No de); return dequened Data; int front (stanct Quene , quene) { return is Empty (quene)?-1; quere - front - data; displey (stand Quene , greene) { stmit Node " averent: quene - front; while (overheart!=NVLL)? Printf ("-1.d", convent -> data);

Corrent - ment - ment : 3. Sin printf(")"); issue assermant somite assert e)#in i sharp grave; # inc int main () mine Course great fiction () primme this Stric stonct Quene quene : create Quene (); 8 stand node, words e orgnene (quene, 2); roid enguene (quene, 3); E st printf ("Quene:"); obsplay(quene); printf("Front: "/.d/n", front(quene)); printf ("Dequene: 1.d/n", dequene(quene)); prints (" anene after dequere: ); voi display (quene); 18 ( Emply Grens ) E printf("Grene undeflow"); return o; of a marp & about being a pool & done, police Output int dequenced Data : dequemed No de-sidata greene spront des neredrode sout, free (degrene drode): 1 turn degreered Date int operationat Buenes guene) ! 1. 6 (mont) by (drone) j. 1 grene - front deta; gestend (Exang Quere, drove) & strong node o consent grand fort while ( owners !: NVEL) { Dist ( one and !: NVEL) { Dist ( one and !: NVEL) }