9/01/204 WEEK-B su-sort, reverse, concatenation Per) o concatenate is struct as node * conca_ lts (struct rode * lists, struct node * lists) if (list == NULL) { 4 roturn lists; struct rode ptr = list1;
while (competr-> next != NULL) { ptrzptr->neut; ptr > neute list 2; return list1; void sortlist (struct node ** head) (2) sort o struct node & ptr, & node next; int temp; while (ptr:=NULL) nodement = ext > next; while (nodewest ! = NULL) {
if (ptr > data > nodewest >)
data tempz ptrzdata; ptr-> data = noderent-> data; nodered anderest - vent; y ptr eptr > varti

stack stop a stack stop swent, void reverse (struct node ** head) ! free (temp); struct rode *prev, *current, *neutrodo veturn pop-V; Preva NULL; current: *icade; b) Implement Queue using Single linked List. while (carrent 1 = NULL) { restribble a current-scent; void Insort (etvet queue queue, introduct) current - rent = prev; struct rade * newnader create node (value); prev c current; y current event Node; if (is Empty (aveva)) queue > front = new Node; 4 * heads = previ queul = rear = new Node; a) Implement stack using linked lists. void initialize stack (struct Stack & stack) quoies rear -s neut 2 new Node; struct indode *top; , que s vear z www. Wode; void initistack (struct stack *stack) dequel struct queue queue) stack-stop & NULL; if (is Empty (queues) void push (struct Stack stack, int value) printf (" Under flow"); the struct node "rewnoder create rode (while); int dequeued value equeue strout sodata; newnode > next estade > top; struct Node * tempe queue sfront; stack-top = newnode; if (queue > front iz que ue > rear) I queue => front = NULLi int pop (struct Stack* stack) & zqueul > rear = NULL; if (is Empty (stack)) { else { queue > front c queue > front rent; printf (" Under flow " \n"); 6 enit(1)+; free (temp) poper = stack stop sdata return (is dequaled value) stract node temp : stack stop;

off Enter your choice: 1) 1) 1. create list 2 areate list 2 3 concatenate & emprist ent enter -1 to ent: enter dota: 1 enter data 1 2 enter data: -1 Enter your choice -1 ere ale list 2. ereate list 2 3. concatonate & print 50 enit enter -1 to ent? enter data: 3 enter data : 4 enter data: 5 exter data: -1 Ender your choice+ 1 execute list 1 2-create list 2 3 concatenate & print 5 . enit Lists are concatenated Enter your choice+ 1. exects but 2-create lit 2 3. concectorate topics 5 \$ aut 1 2 3 4 5 345 Enter choice !-1 create 2 sort 3 print 4 enit enter -1 to enit: enter data: 5 enter data: 2 enter data: 3 enter data : 9 Gater choice+ 1. create 2 sort 3. print 4. exit

1. create 2. sort 3. print 4. enet 2359 3) Enter choice + i. create 2. sofreverse 3- print & emit enter -1 to evit: enter data: 1 enter data: 2 enter data 10 enter data : -1 Enter choice + 1. ereate 2 reverse 3 print 4 evit enter Unicet 11 create 2-reverse 3-print 4 enit 0 2 I) a) Enter choice i) push 21) pop 3) display 4) suit No. of elements to be pushed: 3 element 1: 50 element 2: 60 element 3: 70 1) push 2) pop 3) display 4) out Stack is 70 60 50 Euler choice+ 1) push 2) pop 3) display 4) aut

Enter choicet

Enter choice: Denqueue 2) deque idisplay de \$ 3) det 4) Enit \$ DE 1. Exter -1 to evit: Enter data: 1 Enter data: 2 Exter data: 3 Enter data: -1 Enter chaice: Nenqueue 2) deque 3) display 4) Exit Enter choice: i) enque 2) deque 3) display 4) Ent Enter choice 1 1) enque à deque 3) display 4) bit 05/02/2024 WEEK - 07