```
29/1/2024
LAB-05
or) Stack implementation noing SU.
# in clude estatio. b7
# include <stalib. b>
struct Node &
  int data;
  struct Node * next;
struct Node + top = NUL;
roid push (int data) {
   struct Node new Mode = (struct Node ) malloc (size of (struct Node))
    if (newmode == NULL) {
      printf (" oreeflow in");
                                                    19510AS 243/2 13605
      retuen;
   newnode - data = data;
   neunode -> next = top;
   top = nevonode;
                                                          TION SID
int popl) {
   If (top == NULL) {
     printf ("Underflow 1,");
      return -1;
                                                     Ster positions a
  struct Node temp = top;
   int templ = temp + date;
   top: temp - next;
   free (temp);
    retnen templ;
roid displayer &
  Ff (top = = NULL) &
     printf (" Empty 19");
     retuen;
  struct Node current = top)
  printf ("Stack: ");
  while (current) = NULL) {
     printf (" " d", current - dota);
     current = current - , next;
```

```
Print ("In")
Int masn () {
 int choice, data)
 white (1) E
   printif (" Stack operation Menu: In");
   grintif (" 1. Puch It 2. Pop It 3. Display It
                                              4. exit \n");
   printf (" Enter your choice: ");
   scant [" " d", & choice);
   switch (choice) {
     cased: printf ("Enter data to puch: ");
             scant ("Y.d", & data);
             puch (data);
             breek;
      cased: if (top == NULL) {
                  Printf ("Underflow In"); 3
                 printf (" Popped element: Y-d in", pop());
               break;
                         struct Node" neconode: (struct Node") mallor
      case 3: display ();
               break;
                                                print ( "Overflow (n")
      case 4: printf ("Exiting ... ");
               exit(0);
      default: printf ("Anvalid choice In");
                                                  3 ( Company by brings) &
 retueno;
Output:
Stack operation Menu:
1. Prich 2. Pop 3. Display O. Exit
Enter your choice: 1
Exter data: 3
                                                  Mortingues " ) fred.
Enter your choice: 1
Enter data: 4
Enter your choice: 2
                                                  Francis of payment of or fi
Popped element: 4.
                                                it as contracted to be
enter your choice: 3 stact: 3
```

```
# include <stdio. b7
   It include estallib. b>
   Struct Node &
                                 Car charte blesses (a).
    int data;
     Struct Node next;
                                 struct Queue &
    struct Node' front;
    struct Node rear;
  3;
  roid initialise (struct Queue queue) {
                                             (stok ) do it
     queue - front = NULL;
     quelle -> rear= NULL;
                                        3 (1100 == 40+) } : 5
  Int is Empty (struct Queue queue) {
     return ( queue - front== NULL)
  3
        prints (" Ropped elements 4d 10", pape ())!
 roid enqueue (struct Queue queue, int data) {
    struct Node new new nede = (struct Node ) malloc (size of Catruct Node);
                                               cose 3 display ();
    if ( new node = = NULL) {
      Print ( Overflow (n'))
                                    case of printy constituy. ")
      retuen;
    newnode - data: data; , l'al lastada brance 10 prince : Mustice
    hewnode + next = Albert next NULL;
   if CisEmpty (quene) ) &
      queue - front = new node;
      queue recr: new node;
   else {
      queuer rear next = newnode;
      queue - reer = newnode;
  printf (" Enquened! 10");
int dequeue ( struct Queue - queue) {
  if cis Empty (queue))[
    printf ("Indertow (");
    metues -1;
```

02) Queue implementation using SLL.

```
struct Node temps quelle , frem)
   IN date: temp = data;
   of cqueue - front = queue - rear) Ed .
     queue - front = NULL;
     queue - rear = MULL;
  else E
     queue - front = queue - front - next;
 free (temp);
  printf ( " Dequeued Yd In", data);
  retnen datas
rold display (struct Queue' queue) ?
  of che Empty (queues) &
    printf (" Empty 10");
    retuen;
                                                              13 = house
  struct Mode + ourrent = queue - front;
  printf ("Quene: ");
   while ( ourrent! - MULL) {
     printf ("Yid", autrent > data);
     current - current -> next;
  printf("(n");
At mainc) {
  struct Queue q;
  hitialise ( fa);
  int choice, data;
  while (1) ?
  prints (" Queue Operation Menus 10");
  printf ("1. Enqueue It &. Dequeue It 3. Display It 4 Prit (");
  p. Not (" Enter your choice; "))
  scant (" "d", &choice);
  switch (choice) {
     case! printf ("Enter data to enqueue: ");
            scent (" v.d", +detal;
            enquene ( fg, , deta);
            break;
     Case 2: dequener = 1: break.
```

case3: display(9); breck; case 4: printf c"exiting....")) exitto); default: printf (" Anvalld choice in"); retuen 0; atput: clastets and tox hourses Queue Operation Menu. 1 Enqueue 2. Dequeue 3. Display 4. Exit Enter your choice: 1 g canno "anaco tonto o guerra) g Enter data: 34 Granewed: 3 Enter your choice! 1 Enter dota: \$4 Enqueued: 27 Enter your choice: 1 Enter data: 5 Enqueued: 5 enter your choice: a Dequened: 3 Enter your chorce: 3 Juene: 4 5 Enter your choice: 0 Exiting house Course Chamber Weren 10.11 enging the consumable or Proposed to a supply lit or ent last

201 7000

3 ((SITALE) FINAS

```
00) Sorting, Reversing & concatenation
of thetude estatio. La
st include estallib. b
Struct Node &
  but date;
  struct Node " next;
struct Node head = MULL;
raid Ensert beg (Int data) {
   struct Node newnode =
                             (struct Node") malloc (size of (struct Node));
   newnode - deta = deta;
   newnode - next = head;
   head = newnode;
roid insertend (int data)?
   struct Node new node = (struct Node) malloc ( size of (Struct Node));
   new node -> data = data;
   new node - next = NULL;
   if (head == NULL) {
      head = new mode;
   else {
      Struct Node temp = head;
      while (temp - next! = NULL) {
         temp = temp -> next;
      temp - next = new node;
roid insertat middle (int data, int pos) {
    struct Node' newnode = (struct Mode') malloc(size of (struct Mode))
    new node - deta = deta;
    new node + next = NULL;
    if Cheed == NULL) {
       head = newnode;
    else f
      struct Node temp : head;
      for cintiz1; "kpos-1; int) {
         if (temp) = NULL) {
           temp = temp + next)
```

```
if Ctemp = NULL) &
       new node - next = temp + next;
      temp + next = new node;
    else ?
      Printf (" Anvalid pos In");
   (shuret world) madles (stroot (struct made)
roid sortlist () {
   if ( head == NULL) return;
   struct Node ti, "1;
    int temp;
   for (i = head; i-1 next! = NULL; i=i-) next) f
       for Cj = i + next; jl = NULL; j=j + next) ?
           if (i -> data > j -> data) {
                                                     3 Carrie & A
           temp = i - data;
             i + data = j + data;
             J- date = temp;
                                         of Moder temps heads
                                            tromp remp - next;
 void reverselist() ?
    struct Node prer = NULL,
    Struct Node" current=head;
    struct Node * next node = NULL;
                                       tor suitable that charge help pro-
                                       stade' necounder (drue
     while (current! = NULL) {
        next node = current - next;
        current - next = prev;
        prer = ourrent;
         current = nextnode;
      head = prev;
```

```
roid concatenate lists (struct
                              Node + second) }
   if Cheed == NULL) {
      head = second;
   else ?
     Struct Node + temp = head;
      while (temp + next! = NULL) {
        temp: temp - next;
     temp - next = second;
                          affects saids (" There's charge stunish
(nt man () {
  it choice, data, position;
                                                         ( Halland
   smuct mode + 11st 2 = NULL;
   while (1) {
     print f (" In 1. Insert at beginning 2 - Ansert at End 3. Ansert at
               middle 4. Sort 5. Reverse 6. Concetenate 0. fort (0");
     printf (" enter your choice: "); was a palmained to heart.
     scanf (" "d", & choice)
   subswitch (choice) {
      · basel: printf ( Venter data: ");
               scant (" "Id", Adata);
               msert beg (data);
                break;
        case 2: printf(" Enter dota: ");
                 scenf ("Yd") &date);
                 insertend ( flist 2, date);
        case 3: printf ("Enter date: ");
                 scenf (" y'd" fdde);
                printf ("Enter position; ");
                sconf (" f.d') fiposition);
                 insert at middle (data, position);
                breaks
```

case 4 : Sortlist (); breaks cases; reverselat(); break; case 6; insert at End (list 2, 2); inserter end (Mix 2, 4); concetenate (list 2); preak; case o: printf ("exiting ... "); break! default: printf ("Annaled choice "In); printy ("current list: "); Output. 1. Ansert at Beginning 2. Ansert at End Reverse 6. Concatenate Enter your choice: 1 Onter data: 2 easely profit (Yenley date. 6% Current list: 2 (atala , " har ") faces Enter your choice: 1 Enter data: 3 Current list: 3 2 Enter your choice: 1 Enter data: 4 state S. A. H.) had reserve current list: 4 3 2 Enter your choice: 4 List sorted successfully. current list: 2 3 4 Enter your choice:5 List reversed successfully. aurrent list: 4 8 2 Enter your choice: 6 Lists concertenated successfully. Current list: 43 2 24