Date No.

A.

```
sort() {
  struct node * ptr = * head
  struct node * temp = NULL
  int i
  if (head == NULL) return
  else {
    while (ptr != NULL) {
      temp = ptr -> next
      while (temp != NULL) {
        if (ptr -> data > temp -> data) {
          i = ptr -> data
          ptr -> data = temp -> data
          temp -> data = i
        }
        temp = temp -> next;
      }
      ptr = ptr -> next;
    }
  }
}
```

Date No.

```
reverse() {
    struct node * prev = NULL
    struct node * next = NULL
    struct node * ptr = * head
    while (ptr != NULL) {
        next = ptr -> next
        ptr -> next = prev
        prev = ptr
        ptr = next
    }
    * head = prev
}

concatenate ( struct node * ptr1, struct node * ptr2)
{
    if (ptr1 != NULL && ptr 2 != NULL)
    {
        if (ptr -> next == NULL)
            ptr -> next = ptr 2
        else
            concatenate (ptr 1 -> next, ptr2)
    }
    else { print -> " Either ptr1 or ptr 2 is NULL"}
}
```

Date No.

```
struct node * concat (struct node * start1, struct node
                      * start2)
{
  struct node * ptr
  if (start1 == NULL) { start1 = start2
                        return start1 }
  if (start2 == NULL) return start1
  ptr = start1
    while (ptr -> next != NULL)
      ptr = ptr -> next
      ptr -> next = start2
      return start1
  }

push ( struct node * * head, int new_data)
  {
    struct node * new_node = (struct node *)
                malloc (sizeof (struct node))
  new_node -> data = new_data
  new_node -> next = * head
   * head = new_node

pop()<
    struct node * ptr
```

Date No.

```
if (head == NULL)
    printf("List is Empty").
else {
    ptr = head
    head = ptr -> next
    free (ptr)
  }
}

Enqueue(item){
  struct node * ptr, * temp
  ptr = (struct node*) malloc (sizeof(struct
        node)).
  ptr -> data = item
  ptr -> next = NULL

  if (head == NULL)
  {
    head = ptr
  }
else {
  temp = head
  while (temp -> next != NULL)
  {
    temp = temp -> nal
  }
temp -> nal = ptr }}
```

Date No.

```
Dequeue() {
    struct node * ptr
    if (head == NULL)
    {
        print "List is Empty"
    }
    else
    {
        ptr = head
        head = ptr -> next
        free(ptr)
    }
}
```