**NAMA : SHABRINA QOTTRUNNADA**

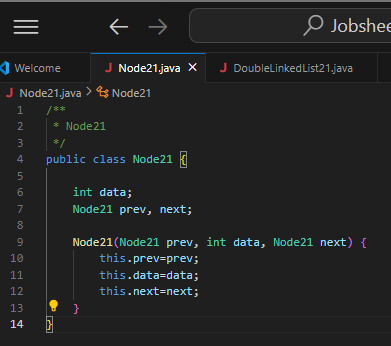
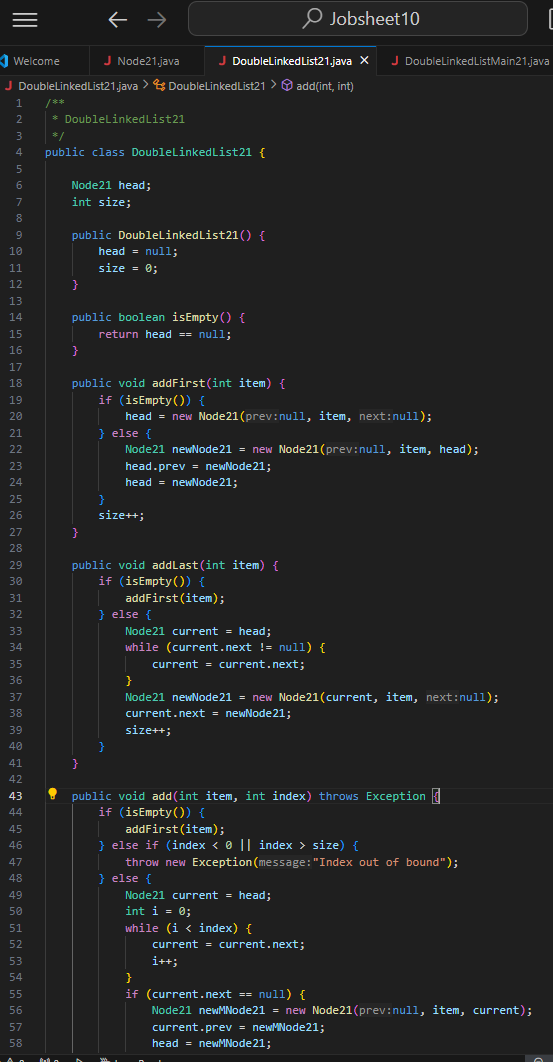
**KELAS : 1G**

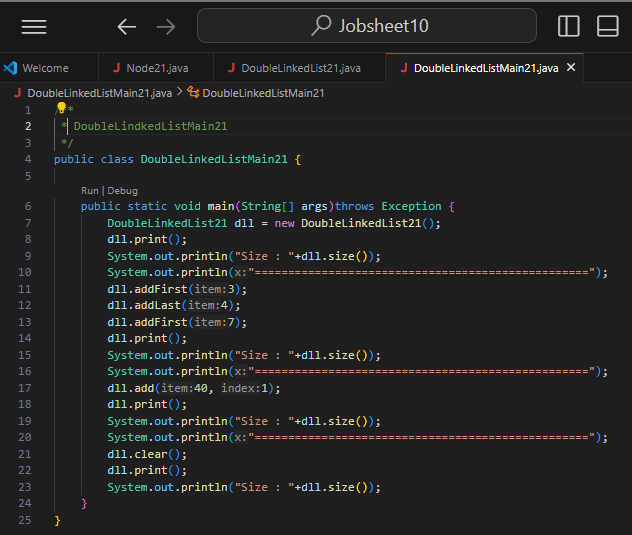
**NIM : 2341760160**

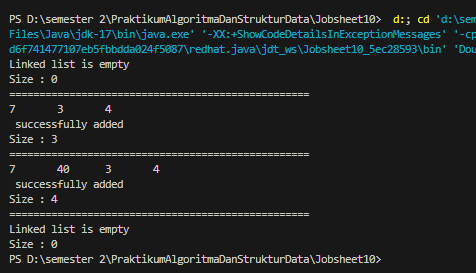
**MATKUL : ALGORITMA DAN STRUKTUR DATA (DOUBLE LINKED LIST)**

**JOBSHEET 10**

**Github :** [**https://github.com/shabrinaq/smstr2\_Jobsheet10/tree/main**](https://github.com/shabrinaq/smstr2_Jobsheet10/tree/main)

********



****

* + 1. **Questions**
       1. What’s the difference between single linked list and double linked list?
* Perbedaan antara single linked list dengan double linked list adalah :
* Single Linked List yaitu untuk penggunaan memori yang lebih sederhana dari double linked list yang dimana untuk memplementasikan program, sedangkan
* Double Linked List yaitu lebih ke fleksibel dalam mengoperasikan penyisipan serta penghapusan, namun lebih banyak membutuhkan memori.
  + - 1. In **Node class**, what is the usage of attribute next and prev ?
* Penggunaan atribut next dan prev yaitu :
* Next 🡺 untuk menyimpan referensi ke Node selanjutnya dalam urutannya yang mana nantinya bila kita ingin membuat program pada file ini nantinya kitab isa menggunakan next ini untuk melewati beberapa node ke node lain.
* Prev 🡺 untuk menyimpan referensi ke Node selanjutnya dalam urutannya yang mana nantinya bila kita menggunakan prev pada program, nantinya bisa lebih memudahkan kita dalam mengoperasikan penyisipan serta penghapusan.
  + - 1. In constructor of **DoubleLinkedList class.** What’s the purpose of head and size attribute in this following code?



* Tujuannya dalam kode program tersebut itu jika ‘head’ digunakan sebagai menyimpan referensi ke node awal di double linked list, sedangkan untuk ‘size’ digunakan untuk mencari jumlh node yang ada pada double linked list.
  + - 1. In method **addFirst(),** why do we initialize the value of Node object to be null at first?

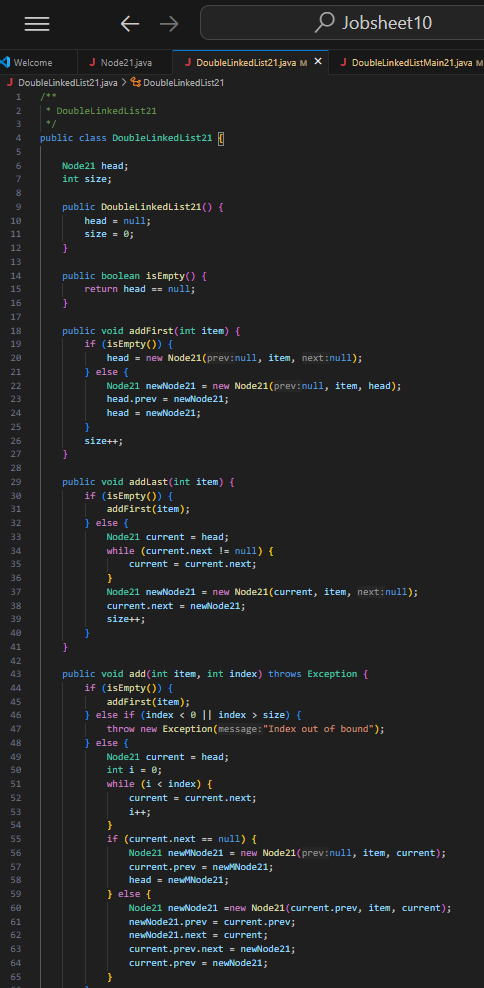
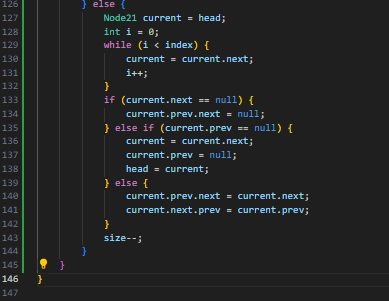
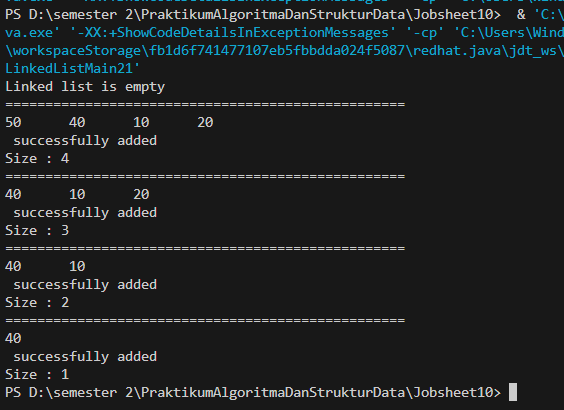
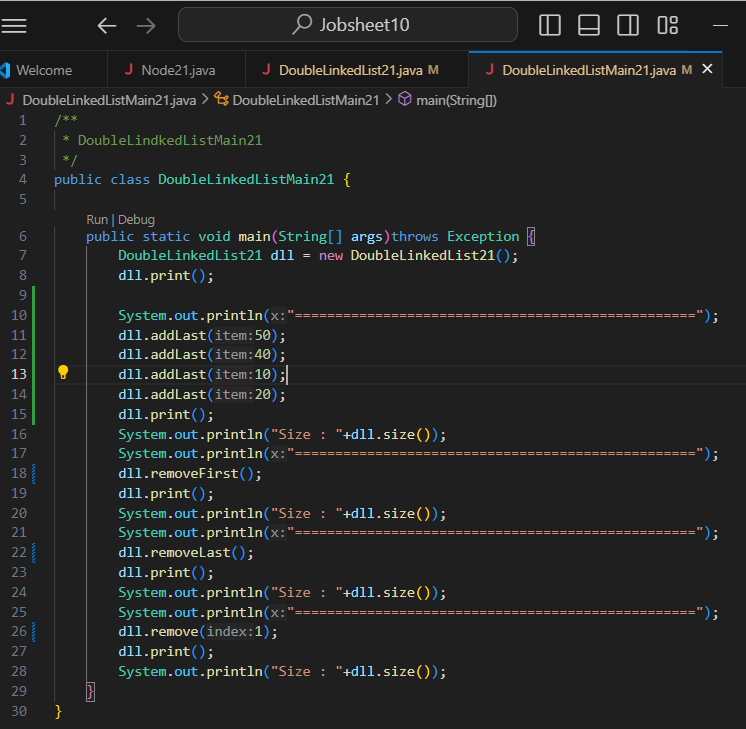
Node newNode = new Node(**null**, item, head);

* Karena inisialisasi ini untuk memastikan node baru itu terhubung dengan benar dan tepat ke daftar dengan ‘head’ menunjukkan sebagai node awal. Lalu ‘null’ itu karena untuk awal daftar.
  + - 1. In method **addLast(),** what’s the purpose of creating a node object by passing the **prev**

parameter with **current** and **next** with **null** ?

Node newNode = new Node(**current**, item, **null**);

* Tujuannya untuk menghubungkan node awal atau bisa disebut baru di akhir daftar dengan menggunakan node terakhir yang ada pada saat ini.



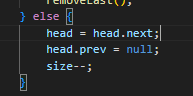
* + 1. **Questions**
       1. What’s the meaning of these statements in **removeFirst()** method?



Untuk memeriksa daftar yang kosong dengan metode ‘isEmpty’, bila daftar tersebut kosong maka linked listnya itu nantinya tidak bisa dihapus



Untuk memeriksa jika satu elemen atau ‘size == I’ maka akan memanggil ‘removeLast()’



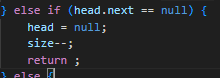
Untuk menghapus elemen pertama yang ada di double linked list jika nantinya ada beberapa elemen

* + - 1. How do we detect the position of the data that are in the last index in method

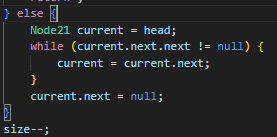
**removeLast()**?



Untuk mengetahui apakah daftar tersebut kosong menggunakan metode ‘isEmpty’, nah jika daftar tersebut kosong maka linked didalamnya itu tidak bisa di hapus



Untuk memeriksa jika ada satu elemen yang ada di dalam daftar ‘head.next == null’, dan head.next akan memeriksa apakah itu null, bila hasilnya iya maka head hasilnya akan null. Lalu memberikan ukuran daftar dengan  dan nanti dikeluarkan dengan ‘return;’



Untuk menghapus elemen terakhir jika nantinya ada elemen yang lebih dari satu yang ada di daftar

* pertama untuk menginisialisasikan variable ‘current’ dengan menjadi ‘head’
* ke dua untuk menjalankan baris tersebut atau loop dengan mencari 2 elemen akhir dalam daftar tersebut hingga menghasilkan null
* ke tiga untuk menjalankan current ke node berikutnya
* ke empat itu untuk mengatur next yang tadinya dari node kedua terakhir atau current menjadi null hasilnya
* yang terakhir itu untuk ukuran daftar bisanya itu sebanyak apa yang diinginkan soal atau output
  + - 1. Explain why this program code is not suitable if we include it in **remove** command!



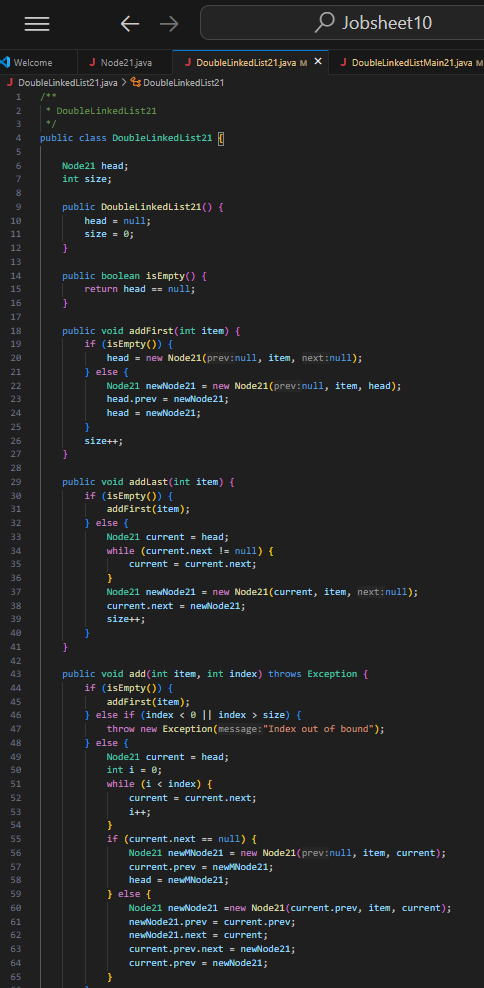
* Mengapa pint tersebut tidak cocok jika kita masukkan ke remove, itu karena kan di remove tujuannya kita untuk menghapus node tertentu atau yang terpilih saja pada daftar
* Pertama itu untuk menetapkan node yang sedang mengikuti head yang menjadi variable tmp
* Kedua itu untuk mengupdate head.next yang akan melewati tmp
* Ketiga itu untuk memperbarui prev dari node kedua yang baru menuju ke node head
  + - 1. Explain what’s the function of this program code in method **remove**!

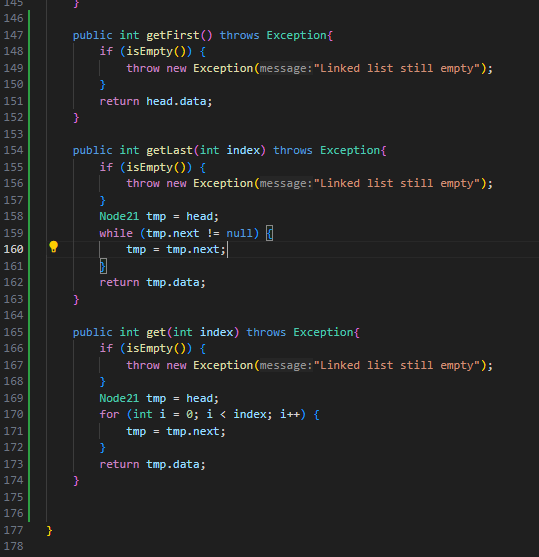
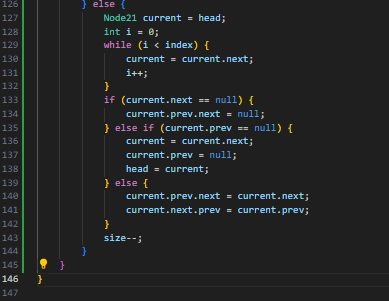


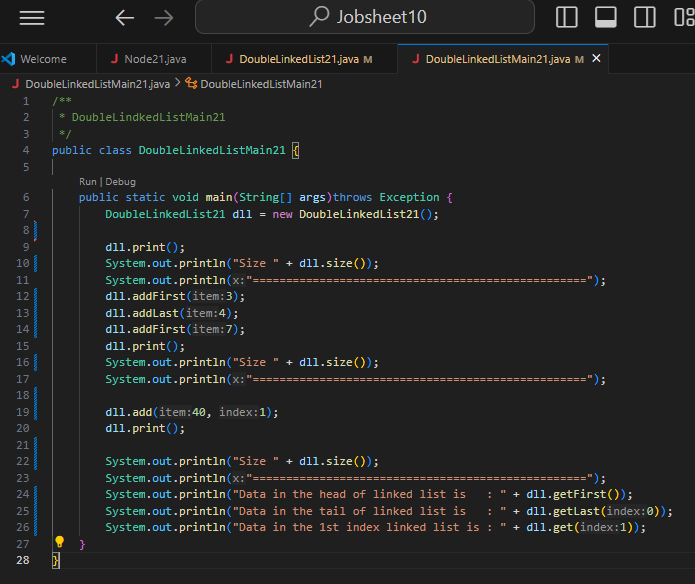
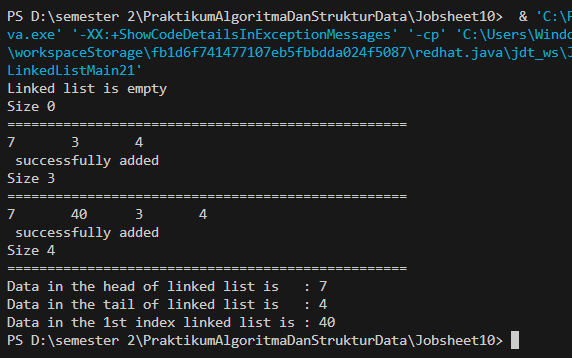
🡺 fungsi dari kode program yang ditampilkan yaitu untuk memperbarui atau mengupdate petunjuk dari node yang saling berdekatan untuk melewat node yang akan kita hapus.

- untuk baris pertama itu mengupdate referensi ‘next’ dari node sebelumnya

- untuk baris kedua itu mengupdate referensi ‘prev’ dari node setelahnya atau berikutnya



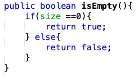
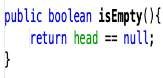




* + 1. **Questions**
       1. What is the function of method **size()** in **DoubleLinkedList** class ?
* Method size() yang ada di class DoubleLinkedList digunakan untuk menentukan berapa banyak angka yang dimasukkan pada output, bila sudah ditentukan angkanya maka tidak bisa menginput data lebih dari angka angka yang telah di outputkan pada size().
  + - 1. How do we set the index in double linked list so that it starts from 1st index instead of 0th index?
* Langkah untuk melakukannya yaitu :

1. Kita sesuaikan metode penyisipan ataupun penghapusan elemennya dari daftar, kita harus memastikan bahwa indeks tersebut dimulai dari angka 1
2. Lalu kita harus memodifikasikan metod get(int indeks), kia bisa mengurangi satu indeks yang diberikan oleh pengguna kalua kita mau mengakses elemennya
3. selanjutnya, kita bisa memperbarui perhitungan ukuran yang dimana kit akita ingin indeks itu tetap untuk mengetahui jumlah elemennya, kita harus memastikan dulu bahwa indeks tersebut itu bisa terlihat oleh pengguna
   * + 1. Please explain the difference between method **Add()** in double linked list and single linked list !

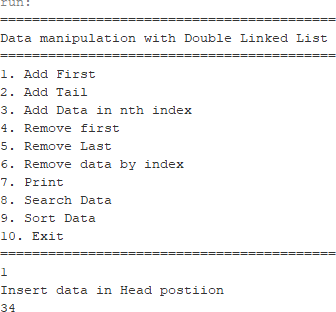
* Perbedaan dari method Add() di DoubleLinkedList dan SingleLinkedList adalah :
* Method Add() di DoubleLinkedList adalah memiliki point untuk ke node sebelumnya ‘prev’ sehingga kita dapat lebih mudah untuk menyisipkan elemen pada akhir daftar tanpa melewati daftar tersebut dari awal, sedangkan
* Method Add() di SingleLinkedList adalah memiliki point ‘next’ sehingga kita harus melewati semua daftar dari awal untuk menemukan elemen penyisipan yang baru
  + - 1. What’s the logic difference of these 2 following codes?



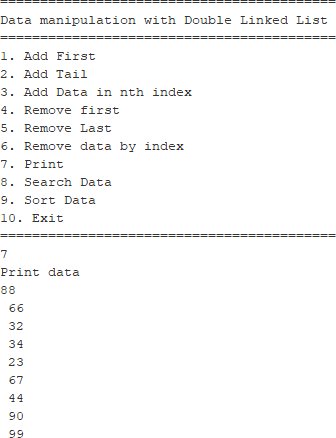
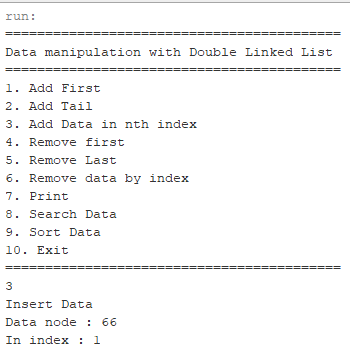
(a) (b)

1. dalam kode program ini terdapat pernyataan if-else yang digunakan untuk memeriksa apakah ‘size’ tersebut sama dengan 0, jika sama maka koleksi akan kosong dan nilai yang akan muncul true, begitupun sebaliknya jika salah maka koleksi tidak kosong dan nilai yang akan muncul false
2. dalam kode program ini merupakan kode singkatnya dari kode program atas yang dimana nantinya langsung dilakukan pengecekan apakah ‘size’ sama dengan 0. Jika benar maka nilai yang akan muncul true, begitupun sebgaliknya
   1. Assignment
3. Create a program with double linked list implementation that allows user to choose a menu as following image! The searching uses sequential search approach and the program should be able to sort the data in descending order. You may any choose sorting approach you prefer (bubble sort, selection sort, insertion sort, or merge sort)

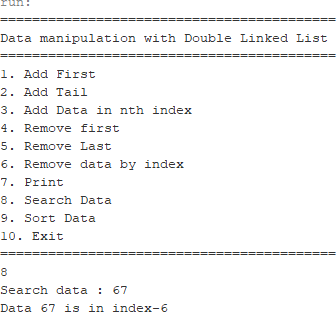
**Adding a data**



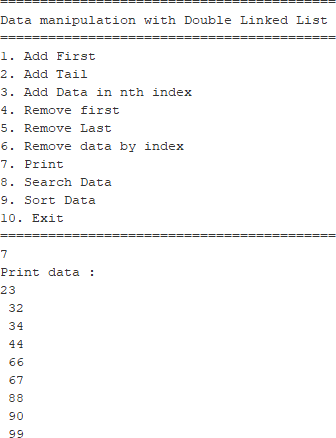
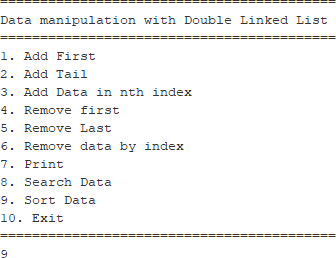
**Add data in specified index and display the result**



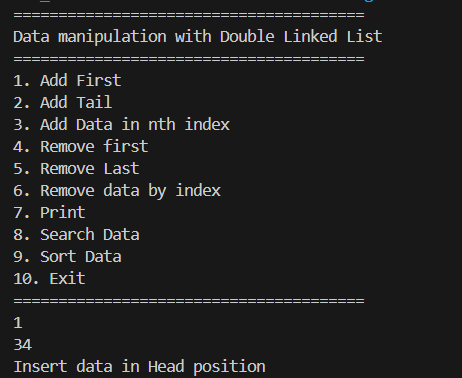
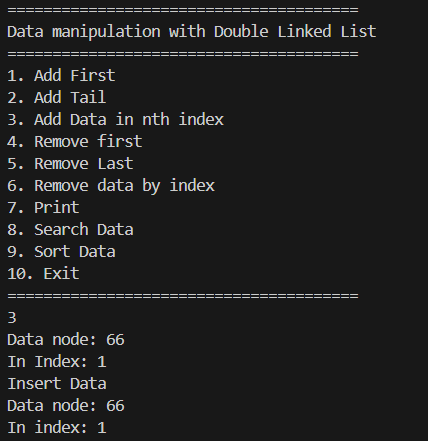
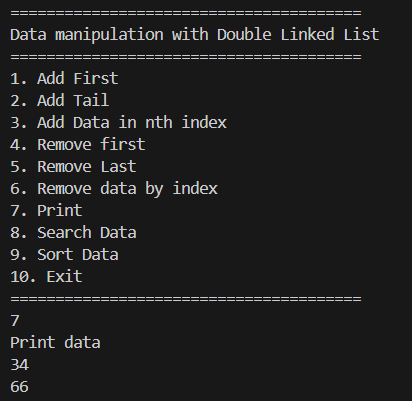
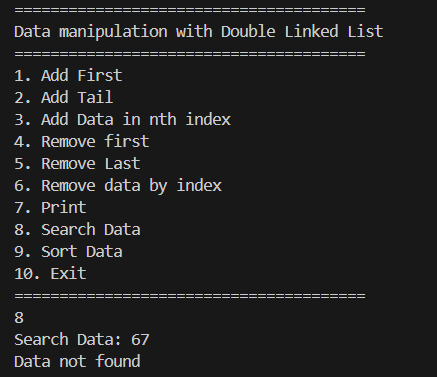
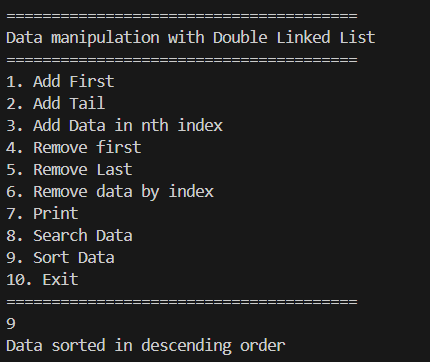
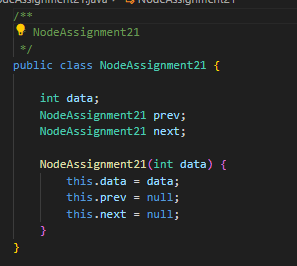
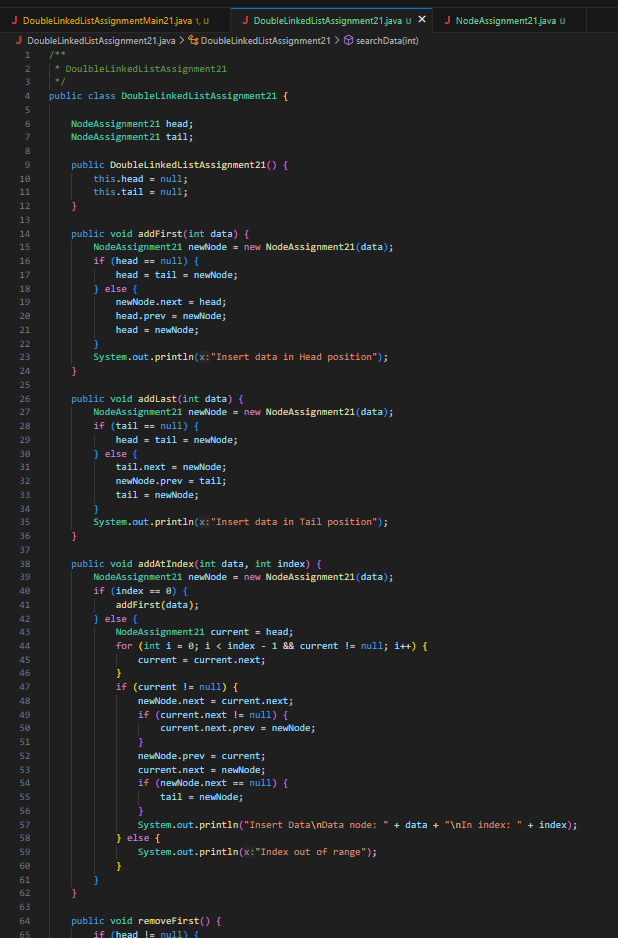
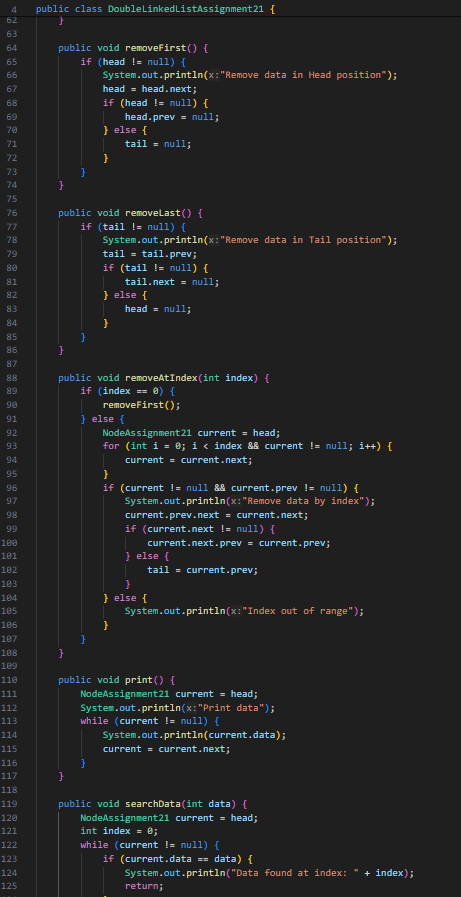
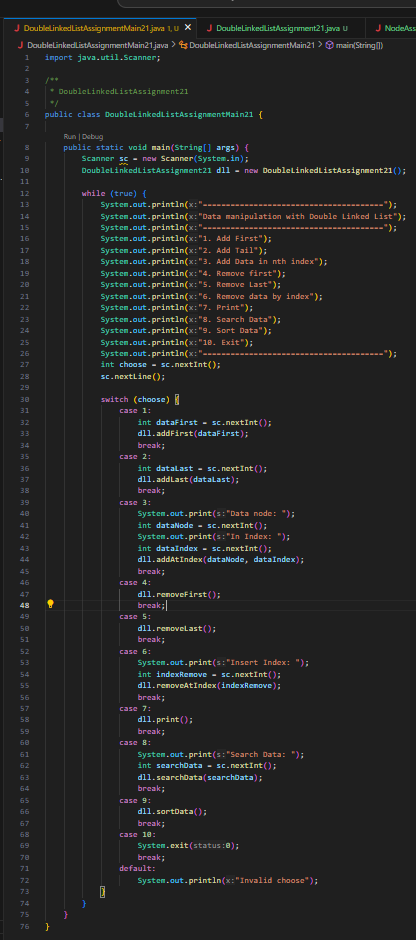
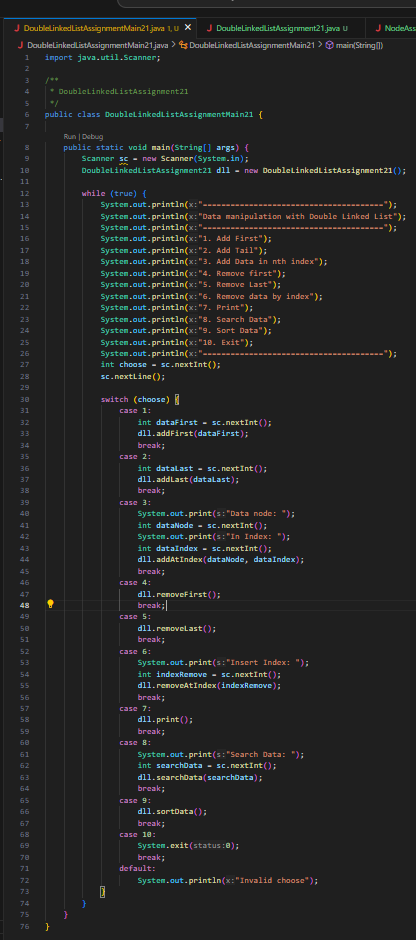
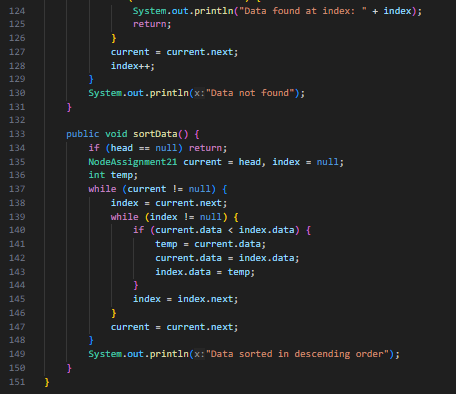
**Search Data**



**Sorting Data**

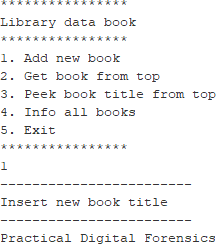


Answer:

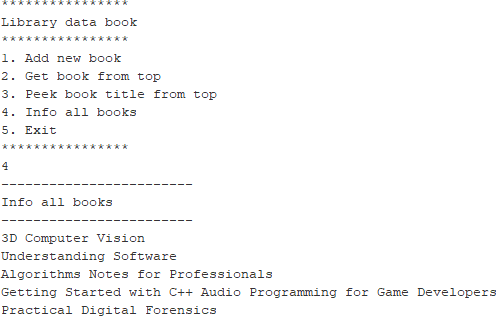
        

1. We are required to create a program which Implement Stack using double linked list. The features are described in following illustrations:

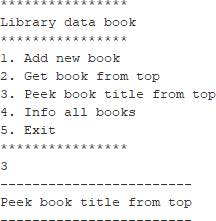
**Initial menu and add Data (push)**



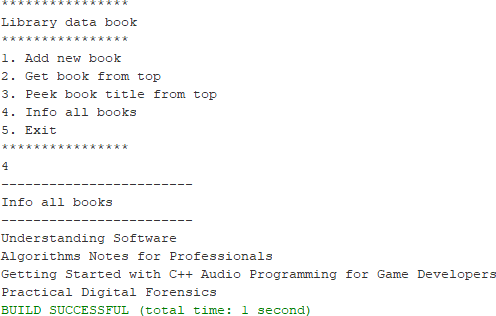
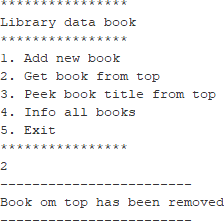
**Print All Data**

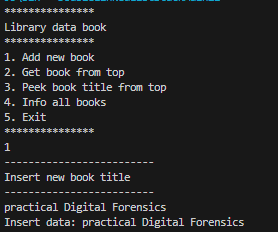
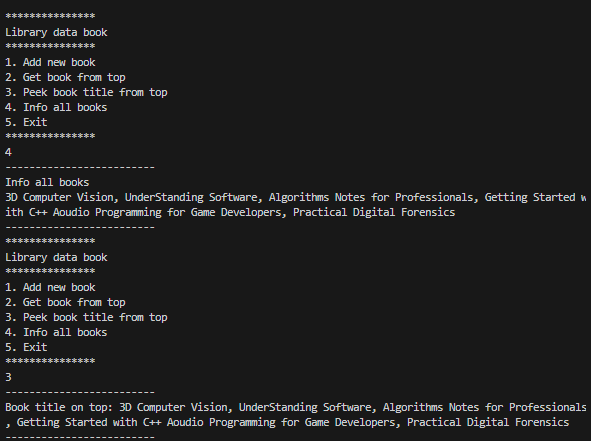


**See the data on top of the stack**



**Pop the data from the top of the stack**



Answer :

