LAB QUIZ

Build a class called **LinkedListDecoder**.

Following this, there are 5 stages of challenges, each with increasing levels of difficulty. Quiz marks will be calculated as follows. Note that the tasks must be completed in order.

1. Complete challenge 1 = ⅕
2. Complete challenge 2 = ⅖
3. Complete challenge 3 = ⅗
4. Complete challenge 4 = ⅘
5. Optimization of algorithm = Full marks

After every challenge, you may check your answer at the link below to validate your code. If your code outputs the correct sequence, you will get 'CORRECT!' alert, and if your answer is wrong, you will get 'Wrong.. Try Again!'

https://desmondyeoh.github.io/ta-datastruct-quiz/index3
https://drive.google.com/file/d/1iZ66a-D-IjKTN4avDmMxsq0VDR37TkZP/view (if can't view, download this and view in browser)
(NOTE: CHECK THAT IT IS WRITTEN SET C IN THE TITLE)

Restrictions

- You must implement your answer in LinkedList data structure.
- You must not use String manipulation techniques to get the final answer.

CHALLENGE 1

`addString(String str);`

- Just split `str` into characters and add each character to the linked list.
- Let the head points to first character of `str` and tail points to last character of `str`.

`toString();`

- Override the toString method to print out all the nodes in the linked list to a one-line string, with each character separated by a underscore (_)
- For example, `'a' -> 'b' -> 'c'` will print out `a_b_c`

Run the following code and check your answer.

CODE:

```
lld = new LinkedListDecoder();
lld.addString("pvhomjhlaiwcibhapsytrqrsvtquwjntxjlxtajjgveiddinxicxlarnjqrjozsuqvbblntreskidyv
yxbskfcjewayfbkfabefswoujnjzjkvwqtdffksclvrzpoaldnslujhnzgebwwewhjfhpxkcpooraxknbtedygjnuxcfme
ocdjitqxtnyjkumscdejqcdbspjrzdmqunosiydlnxqmvifqddgpauzqywxwmtrwcxrwatteixrusbaglooisiyhvnscdm
gjnwcxpaxpngbynoqfyuizhomzfbbqygugsuhoftkyzmcjlwnccglufadzwlpouifaeptawdfficqdtmoadzhznrvtqiqu
nzkpreozoppjfkjmcvakgiilpxkfgkhbgnqfejvdkrvkqgicrdyudmwmsgfttisciqgmetlvicabohheefnbrpdbxtmzgr
sztxldwuwoqrpnppsiedcnvtbbpsedcokxgksoxqmjskzebzvcgvfbmggpvpjcqbklbquqgiqfjislfcwwepqvdacvbcoh
pjaskdeauxombiwvjfnnvxgswfpzrhzilczdycdyahsohugoneulrlshkaiotpiwhmdlphxiywjplsmvkduqcuaqmucxlr
zofqeazytzhuejyikjilgcqbryaefqolmrmgyhheoimkwomkgtzqnxfrfryrwztxjjpojodzsgycvlyejvohatknuawtft
skzlevnqepmsmgppenxfemnofeigtorqvroojgfzjwusqzcenkuybzvzbuztwfekdaylluaxhykxurifksbnmdhpbeyrya
bblhxpmlwcgquaedlhucghhqdcvkkbnpdyixbmuwxyikpfsaqahhgworhtarmnuzwutjxxrtvclggwnrahlfyzdmbcvfvz
ynpylltlaztskitxxghtkuheuphsmmbdzmvwbeyvwyzvovsnmuaigwdejxbsrgxpkngthkmbhrmovucyxnpazhqlrxqevr
rylryassmfblkwjivvtyw");
System.out.println(lld);
```

HINT: Your answer will start with `p_v_h_o_m_` and end with `v_v_t_y_w`

CHALLENGE 2

`A(int n, char a, char b);`

- This method will add character `b` after the `n`-th occurrence of character `a`.
- Eg: [‘f’, ‘c’, ‘f’, ‘g’] ⇒ A(2, ‘f’, ‘z’); ⇒ [‘f’, ‘c’, ‘f’, ‘z’, ‘g’]
    - Add ‘z’ after the 2nd ‘f’.
- Run the following code and check your answer.

CODE:

```
lld = new LinkedListDecoder();
lld.addString("kuwszxcwcjwjilqwzeenkgzyggsxubtwxuxvkqylmwctilqzwwknmgpsrsshfdxygvjcfhpszneuoen
cgsgdcudkbnefsvxzpqjugixshgfocmfxysqklkpbvkvunkleuvlfrlvpwgmhvgoegegwstzbfmgnagrumtertbiiuoxrg
gowurlhmaqsljkzczivqzlkrhtirjoubzztyiijnaqowwupwpylycatnpqbzbsjdfezjjglnetjcfnldfowckphiannirq
vqqrobywshmwblssqydzqsifapqeukjpelvysjevyapqkffemmwsaarxyzbkqvxmmiqycgynfryxdistofjdzblxmnwcmt
ffmosjlliiqqnuvlexnhzqiypuzupsyetojmmtqgwecqdvibosvmcdzjjemnlknmwjnstbtemwpxkcabohaejsdbsvlxbb
rvnfkrffrxaxwqrcwthvorixekafxpuibhgaduxtrjfajabpjoovbjpirhvdebzwtkoamghpfutcbqcbkwptcosxncvtuz
irduhfetjaipeqaarkqfuyjhvwhslpmyhfnkyhysblajnwhwixepzwdxapylrggdehkknydudnninbnvcyvxuzcjylasfy
zskparnwojmxnzuzmycbepmdumnlfijwvlamqcerjoheqgezhhgairxqqcjvurknrrsfkmbirdqabuomiflgcwdhmupegr
hulovtwlateshepagghswfxhbolahdtgcclkxbjcrduyyqbhhasbfyfahnxljeoxhpqrympdlqkiddqcdifdiasimhpfed
acuxsgvtkzzhnjpyhwbzwudktevxqoboktuturogzyuooxspdcitxhczeptsniuqrvckfkadgjmnzrvrxzxalbmxzvgctp
vttovtvmzvfzdkyakwozymcyqoplabnogrvgribtovhbmtzuihoitnkdcdflgdyxyciwtfdjqpwyitfepugngljjpdmroc
klborsmdepoxjdavaklyo");
lld.A(39,'o','s');lld.A(21,'u','c');lld.A(30,'h','w');lld.A(8,'k','t');lld.A(24,'x','i');lld.A
(17,'j','x');lld.A(29,'y','n');lld.A(11,'w','g');lld.A(4,'t','q');lld.A(33,'c','k');lld.A(18,'
n','g');lld.A(14,'g','l');lld.A(38,'c','n');lld.A(33,'y','m');lld.A(16,'q','p');lld.A(32,'y','
i');lld.A(28,'y','o');lld.A(5,'r','d');lld.A(33,'q','d');lld.A(14,'m','t');lld.A(8,'e','s');ll
d.A(18,'m','a');lld.A(15,'e','b');lld.A(21,'w','v');lld.A(38,'s','z');lld.A(14,'g','r');lld.A(
36,'h','b');lld.A(9,'o','f');lld.A(15,'p','x');lld.A(2,'p','l');lld.A(31,'s','f');lld.A(38,'k'
,'h');lld.A(26,'i','e');lld.A(28,'p','v');lld.A(28,'m','p');lld.A(5,'g','h');lld.A(7,'j','d');
lld.A(7,'p','a');lld.A(31,'f','k');lld.A(2,'a','o');lld.A(13,'a','n');lld.A(13,'d','t');lld.A(
8,'b','x');lld.A(8,'h','m');lld.A(34,'f','n');lld.A(40,'v','h');lld.A(29,'e','h');lld.A(8,'z',
'i');lld.A(41,'a','o');lld.A(2,'q','q');
System.out.println(lld);
```

HINT: Your answer will start with `k_u_w_s_z_` and end with `a_k_l_y_o`

CHALLENGE 3

`B(int i, char a, int j);`

- This method will remove `j` letters after the `i`-th occurrence of character `a`.
- Eg: [‘a’, ‘a’, ‘a’, ‘c’, ‘b’, ‘c’, ‘c’] ⇒ B(2, ‘a’, 4) ⇒ [‘a’, ‘a’, ‘c’]
    - Remove 4 characters (‘a’, ‘c’, ‘b’, ‘c’) after the second ‘a’.
- You may assume that `i+j` will always be smaller than the size of linked list - 1. That is to say, you can assume the following case **will not happen**.
    - Eg: [‘a’, ‘a’, ‘a’, ‘c’, ‘b’] ⇒ B(2, ‘a’, 3)
- Run the following code and check your answer.

CODE:

```
lld = new LinkedListDecoder();
lld.addString("xfiqvquwtqvnfkfirfxvdyqkwhafrsoqqskxchrngbbknkxraveuuzilccojnzptilsmspzqxuatnpd
xlqpvydbiggikkuvfocumzryzjkatwcllglikccblscmweekvnovvoypdemrxfwjvwyxrmqgstrjymhmbzyofmaizrziaj
geymefkukarzdebzijisregphbxirwtythfgvoiqlzfmspplqmqsjgpueijkfromlksmzgtdfddnzghybvexjhcohladjx
rtsaagcnfhvojxbtkywkctcvclclzgosrhklmatchyobhszzorivzlcqeonrxdykdegpcxyvqyxtajmooieudbadsntusp
qxwxjxbdsgbdnpmstnnvtpbhwukxrarrxbdpnzlmxvpvruojdfutygqbpqodnpudurcmriluofdlswniyheumwqkltjmvl
xjadybgtcefahoqudhqalgpqbgsfwhcwaunoyoarfljqihtzdyifpkixoeyzlckhwdnntgehyqbwlkhafywzcnwuxhblhe
jdlpmtsefkmubvauwleowvbbmwnqmzayqrbiwlxmfxwfpjqprztezeyxesxsekhpocpsmutgadrsrzncgaabzjyjvpjjek
bsqfhmwmfioutweieylnsnmzbnigdrwvkwwqngjhvesfkfhyeukstvcuynbscxaruihpavndvcpiqgufowyguzggdontic
jmjibbifeaazsdcvpoozwjjcxtkattjglgnjvqhphpu");
lld.B(6,'j',1);lld.A(5,'i','f');lld.B(14,'g',1);lld.B(11,'b',1);lld.B(9,'o',1);lld.A(9,'i','p'
);lld.B(6,'m',1);lld.B(6,'o',1);lld.B(11,'m',2);lld.A(23,'q','r');lld.A(8,'e','p');lld.B(14,'a
',1);lld.B(11,'o',2);lld.B(15,'s',2);lld.A(29,'i','g');lld.A(25,'m','i');lld.A(13,'y','o');lld
.B(19,'r',2);lld.A(28,'o','a');lld.B(7,'x',2);lld.A(11,'z','m');lld.A(21,'t','p');lld.B(22,'k'
,2);lld.B(15,'i',2);lld.A(3,'d','n');lld.B(11,'s',2);lld.B(15,'c',1);lld.A(26,'h','j');lld.B(1
7,'k',2);lld.A(8,'w','e');
System.out.println(lld);
```

HINT: Your answer will start with `x_f_i_q_v_` and end with `p_h_j_p_u`

CHALLENGE 4

`C();`

- This method will reduce the size of linked list by half by doing mean pooling with size of 2 and stride of 2. Size of 2 means we take the mean of 2 characters. Stride of 2 means we take 2 steps at a time.
- Definition of mean:
  - Mean of a, a ⇒ a
  - Mean of a, b ⇒ b # Take the ceiling
  - Mean of a, c ⇒ b
- If the length of sequence is odd, the last character stays the same.
  - Eg: [‘a’, ‘a’, ‘a’, ‘b’, ‘c’, ‘z’, ‘e’, ‘c’] ⇒ C() ⇒ [‘a’, ‘b’, ‘o’, ‘d’] # Case 1: even
  - Eg: [‘a’, ‘a’, ‘a’, ‘b’, ‘c’, ‘z’, ‘e’, ‘c’, ‘z’] ⇒ C() ⇒ [‘a’, ‘b’, ‘o’, ‘d’, ‘z’] # Case 2: odd
- Run the following code and check your answer.

CODE:

```
lld = new LinkedListDecoder();
lld.addString("tfjbgrmhgdgxbquxhqjvgmvhxfptdsfalpqjjrlluzaqhrpgujalxksbdnarxwmbcsugrqxuuwvixlh
kivfjuczgkfqxirhcqztupljtdivdwhnieupilaconldyyjekeeasjbchvysqnhjsapenrghgaesujhjobfnomqwjklyaq
tifyutnzgzkazqporblgscuoemxlzspqlehcpudadzbdpzngwblwvmkclzgkmsurslnsuupwyndqipkqsyffyntdmfyhab
nliegdvzwikhagoxkannudrhhssrubdxlybutqviuetxgenagelchebrlpcumyjhsdctuodfmqimapnbftswygjecokyrq
aofhzhkbmkosuqwwycjegqgtztmzbjzlstoxahaxcchsiskwhfdrkbqeyuyenkspfbgyynqobzsalrrcpyyptbfkwedjqr
bsuixiveozkulgbfzrsiglpmxyjwoqdmzmufqelupfgongusmgtqylahdpdaoupgmxhmaxcbztnwuylyvivsknveebjxbg
mdxrbreredpqsjivxwamidkttljynfvoxchtwldlbkttgqieiirnrdpuwannrgrwcusachvvmsvjhlknrgxtpztgwrmovz
wolrvnytcsgfilhzbuykxkqwhmfatvmyvksjzmziowbaillpstlkdlgaofgppgztxobsnzyvzjxlzymhcgnvoqrdddrpdp
vtawravkxtxbqolhypprojvmcydpsncxfdijwlkpxfshaudvgwtifouxnsuckhgryjheomzqutubthfksbevumiyokjako
mjqhoyxcnmpkeapyrktqynwigzjszzqckstumwmjoxrmorrpbaeokmufaqixyjqaucpkvocbprbtkoerbwtlhufnxvdkrz
elegtqnumapvmkeegiewsfoaxmuksijepxjbdawibciqvawgymodagvwvfygsknaxjwknxcgdraxvaefmiqhukwhpxnshx
fnvbsvilmvdfcxcnvbgxdbeqdhvimwoqrijtmnzgjwjoumdvxhqjyedtfdyeyfowvrjlsikvinyubmejritrwkqnpzidct
qynecyemaibsrpwxeaixdjxcmwitlicbtevfiqfawfwsnkpgetbxobohdhrjotxabpfxsidlfmdtmvwtuzqmfgqjchkize
zwvyfcizqeuhuazfcqfatkrmnqocceptlyoxfznmaiipvqmfvtfxxnleyotycrfqqatwfwdecwdopfjyngfdlhuljtqrtn
losduiulacbtrsjypevmtvrgijvnshkpwwyzmdiugaarfopjczokgxuzhzsajvvjeeoszhlhmnaovzizkoehbkrotpnelr
expojwiekkqmjgjhfzolgkyvhvwzzbbqwofhcomchzyxwlypfaybdwzqpwlwhcjgssnkybumlzwaxbjfscjnzypoticgwz
jszrccpbpxrdccwghcdnpvbrfncpxuskqeznlbqrgrybjdtosczlhuvbnikqildcfndfecv");
lld.A(59,'k','g');lld.A(52,'e','s');lld.A(54,'c','j');lld.C();lld.C();lld.A(9,'i','z');lld.C()
;lld.A(7,'g','f');lld.A(1,'w','b');lld.A(7,'x','b');lld.A(3,'t','c');lld.A(1,'h','r');lld.A(6,
'm','c');lld.C();lld.A(3,'g','n');lld.A(3,'j','v');lld.A(2,'g','t');lld.A(2,'y','b');lld.C();l
ld.C();
System.out.println(lld);
```

HINT: Your answer will start with `o_p_n_p_o_` and end with `o_p_o_o_m`

END OF QUESTION PAPER