

LINK

Deze opgave gaat over het manipuleren van pointers.

We gebruiken in deze opgave dezelfde struct (**element**) als we hebben gebruikt in het college week 2 bij gelinkte lijst. In het voorbeeld in de les (week2) werd het aangeboden woord steeds in een struct opgenomen die aan het einde van de lijst werd geplaatst.

Hier gaan we twee alternatieven implementeren.

1. Plaats (de struct met) het nieuwe woord aan het begin van de lijst.
Het nieuwe woord wordt het 1^e in de lijst. De rest van de oorspronkelijke lijst wordt hierachter geplaatst.
2. Plaats (de struct met) het nieuwe woord op de alfabetische positie in de lijst.
Voorbeeld : stel de lijst bevat: **root** → '**dit**' → '**is**' → **null**
en het woord: '**een**' wordt aangeboden.
Dan is het resultaat: **root** → '**dit**' → '**een**' → '**is**' → **null**.
Zie tip.

Codeer instructies :

Je kunt het voorbeeld uit de les gebruiken als uitgangspunt.

Bij het manipuleren van pointers is de volgorde altijd van belang.

Wijzig de inhoud van een pointer pas als de info op de plaats waar de pointer naar wijst:

1. niet meer van belang is of
2. is veilig gesteld.

De ToonLijst functie geeft een string return die geen spaties mag bevatten.

Let op: belangrijke info staat in het commentaar in de header file.

Gegeven is de volgende code :

link.cpp:

```
#include <iostream>
#include "link.h"

// deze 1e mag je overnemen uit het voorbeeld van de les.
void VoegToeAanEindeLijst(element ** lijst, std::string tekst ){ ... }

void VoegToeAanBeginLijst(element ** lijst, std::string tekst ){ ... }
void VoegToeAlfabetisch(element ** lijst, std::string tekst ){ ... }
std::string ToonLijst( element * lijst ){ ... }

int main()
{
    ...
    return 0;
}
```

LINK

Conditie:

Zie het commentaar in de header.

Merk op:

1. De inhoud van de '**main**' routine is niet van belang voor de test.
Je mag hier test-code opnemen voor de gemaakte functies, geheel naar eigen inzicht.
2. De 'signatuur' van de functies mag niet worden aangepast. Wijzigingen geven fouten in de test.

Tips:

Voor het 'vergelijken van string' bestaat de functie `compare()`. Zie: [referentie](#)

Upload instructies:

Upload de volgende bestanden:

1. link.cpp