Skilaverkefni IV (Endurkvæmni/Recursion)

1. Palindrome (20%)

Skrifið eftirfarandi fall / *Write the following function*:

```
bool palindrome(const char a[], int first, int last)
```

Þetta fall skilar true ef character fylkið a er „palindrome“ á bilinu first til last, þ.e. strengur sem hægt að lesa á sama hátt frá vinstri til hægri eða frá hægri til vinstri / *This function returns true if the character array a is a palindrome in the range first to last, i.e. a string which can read in the same manner from left to right as right to left.*

Athugið að þið megið **EKKI** nota neinar lykkjur í útfærslunni á þessu falli / *Note that you are **NOT** allowed to use any loops inside this function.*

Aðalforrit er gefið sem sýnir dæmi um inntak og úttak. Jafnframt fylgir skrá fyrir skil, `palindome.h` en þið eigið að útfæra fallið í `palindrome.cpp`. / *A main program is given which shows example input/output. An interface file* `palindome.h` *is also given, but you need to implement your function in the file* `palindrome.cpp`.

2. JumpIt (30%)

Skrifið eftirfarandi fall / *Write the following function*:

```
int jumpIt(const int board[], int startIndex, int endIndex)
```

Lýsingu á þessu falli er að finna í verkefni 6 í kafla 14 (bls. 859) í kennslubókinni / *The description for this function can be found in project 6 in chapter 14 (page 859) in the textbook.*

Athugið að þið megið **EKKI** nota neinar lykkjur í útfærslunni á þessu falli / *Note that you are **NOT** allowed to use any loops inside this function.*

Aðalforrit er gefið sem sýnir dæmi um inntak og úttak. Jafnframt fylgir skrá fyrir skil, `jumpit.h` en þið eigið að útfæra fallið í `jumpit.cpp`. / *A main*

program is given which shows example input/output. An interface file `jumpit.h` *is also given, but you need to implement your function in the file* `jumpit.cpp`.

3. Umraðanir/Permutations (50%)

Lýsing á íslensku

Athugið að þetta verkefni getur reynst erfitt og krefst örugglega aflúsunar!

Þið eigið að leysa dæmi 8 á bls. 860 í kennslubókinni. Lesið lýsinguna í bókinni vel og verið viss um að þið skiljið lausnaraðferðina

Skilin, `permutations.h`, fyrir `Permutations` klasann og aðalforritið, `main.cpp` er gefið. Hluti af útfærslunni, `permutations.cpp`, er líka gefinn en þið eigið að forrita það sem á vantar í þeirri skrá (merkt með „You have to implement this function“).

Mengið af umröðunum er í okkar tilfelli geymt sem tengdur listi af kviklegum fylkjum (sjá `permutations.h`). Þið eigið, eins og bókin ræðir, að skilja síðasta (n-ta) stakið eftir úr menginu $\{a_1, a_2, ..., a_n\}$ og finna á endurkvæman hátt allar umraðanir með því að nota mengi af n-1 stökum. Síðan bætið þið síðasta stakinu við í sérhverri stöðu í öllum umröðunum.

English description

This part may turn out to be difficult and you probably need to debug your program!

Solve exercise 11 on page 839 in the textbook. Read the description carefully and make sure that you understand the method.

The interface, `permutations.h`, for the `Permutations` class and the main program, `main.cpp`, is given. Part of the implementation, `permutations.cpp`, is also given, but you need to implement the missing part (marked as „You have to implement this function“).

A set of permutations is, in our case, stored as a linked list of dynamic arrays (see `permutations.h`). You need, as discussed in the book, to leave out the last (n-th) element of the set $\{a_1, a_2, ..., a_n\}$ and find, in a recursive

manner, all the permutations by using a set of n-1 elements. Then, you add the last element to each position in every permutation.

Dæmi um virkni / An example run:

```
Enter a postive integer: 2
Permutations of the set of numbers from 1 to 2:
1: {1,2}
2: {2,1}

Again?(y/n): y
Enter a postive integer: 3
Permutations of the set of numbers from 1 to 3:
1: {2,1,3}
2: {2,3,1}
3: {3,2,1}
4: {1,2,3}
5: {1,3,2}
6: {3,1,2}

Again?(y/n): y
Enter a postive integer: 4
Permutations of the set of numbers from 1 to 4:
1: {3,1,2,4}
2: {3,1,4,2}
3: {3,4,1,2}
4: {4,3,1,2}
5: {1,3,2,4}
6: {1,3,4,2}
7: {1,4,3,2}
8: {4,1,3,2}
9: {1,2,3,4}
10: {1,2,4,3}
11: {1,4,2,3}
12: {4,1,2,3}
13: {3,2,1,4}
14: {3,2,4,1}
15: {3,4,2,1}
16: {4,3,2,1}
17: {2,3,1,4}
18: {2,3,4,1}
19: {2,4,3,1}
20: {4,2,3,1}
21: {2,1,3,4}
22: {2,1,4,3}
```

```
23: {2,4,1,3}
24: {4,2,1,3}

Again?(y/n): n
```

Athugið að hér birtist listinn yfir umraðanir ekki á sama máta og sýnt er í bókinni (berið t.d. saman lista yfir umraðanir á menginu {1,2,3} hér að ofan við bókina). Ástæðan er sú að lausnin, sem úttakið að ofan byggir á, setur nýjan hnút (nýja umröðun) fremst í listann en ekki aftast / *Note that here the list of permutations is shown in a different order compared to how it is shown in the book (for example, compare this output and the output in the book for the list of permutations of the set {1,2,3}). The reason is that the solution, on which the output shown above is based, puts a new node (a new permutation) at the front of the list but not at the end of the list.*