

## Suffix array – Specifikáció

### Feladateleírás:

Készítsen objektumot ún. suffix array megvalósításához! A suffix array a stringekben történő keresést gyorsítja. Lényegében egy olyan tömb, amiben rendezetten tároljuk az összes olyan stringet, ami előállítható egy adott stringből a string elején levő karakterek elhagyásával.

Tegyük fel, hogy keressük "je" minta előfordulásait az "objektumje" stringben. A suffix stringben való bináris kereséssel azt kapjuk, hogy a suffix array 2. és 3. indexei határolják a keresett "je" mintát. Így innen könnyen kiolvasható, hogy a "je" minta az eredeti "objektumje" stringben a 8. és 2. pozíción található meg. A példában csak két pozíciót találtunk, de elképzelhető, hogy egy minta kettőnél többször illeszkedik, ezért célszerű iterátorral visszaadni a talált pozíciókat.

Valósítsa meg az összes értelmes műveletet operátor átdefiniálással (overload), de nem kell ragaszkodni az összes operátor átdefiniálásához!

Demonstrálja a működést külön modulként fordított tesztprogrammal! A megoldáshoz ne használjon STL tárolót!

### Feladatspecifikáció:

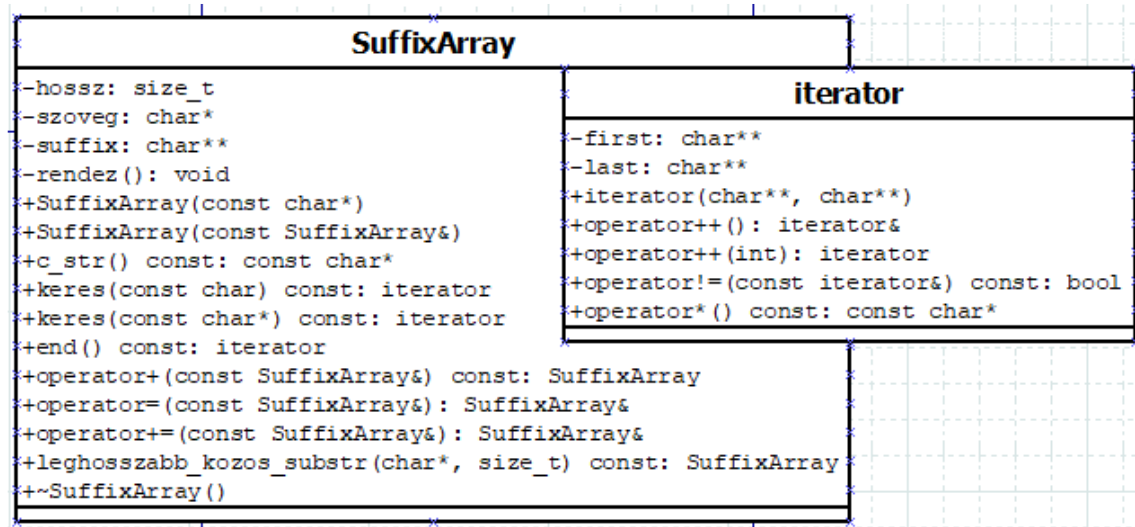
A feladat egy suffix array elkészítése. Az array bármekkora karaktersorozat kezelésére lesz képes. A bemenet lehet egy fájl, ha a program egy fájl útvonal paraméterrel hívódik meg, ellenkező esetben pedig a konzolból olvasódik be a suffix array tartalma. A benne keresett suffix-ot csak a konzolból lehet megadni.

A suffix array-en elvégezhető műveletek:

- keresés: megkeresi a keresett karaktersorozattal kezdődő összes előfordulást az array-ben és visszaadja őket egy iterátorban felsorolva.
- összefűzés (overload +): összefűzi egy másik karaktersorozattal.
- leghosszabb közös prefix (overload &): visszaadja egy másik suffix array-el alkotott leghosszabb közös prefixet.

## Suffix array – Terv

UML diagram:



Adattagok és függvények részletes leírása:

- **szoveg:**
  - tartalmazza a megadott eredeti karaktersorozatot.
- **suffix:**
  - stringeket tartalmazó tömb
  - a szoveg változó összes karakterétől kezdődő és a végéig tartó karaktersorozatot tárolja növekvő sorrendben
- **rendez():**
  - a suffix tömb tartalmát rendezi
  - a rendezéshez sort függvényt használ
  - az összehasonlítást a `string <` operátora végzi
- **SuffixArray(konstruktor):**
  - létrehozza a SuffixArray-t
  - inicializálja a szoveg változót a kapott paraméterek alapján
  - feltölti a suffix tömböt a szoveg változó részkaraktersorozatával
  - meghívja a rendez függvényt
- **SuffixArray(másoló):**
  - lemásolja a szoveg, suffix változó tartalmát
- **c\_str():**
  - visszaadja a tárolt szövegre mutató pintert

- keres(const char):
  - megkeresi az összes olyan suffixot ami az adott karakterrel kezdődik bináris keresés segítségével
  - visszaadja a találatokat iterátorként
- keres(const char\*):
  - megkeresi az összes olyan suffixot ami az adott karaktersorozattal kezdődik bináris keresés segítségével
  - visszaadja a találatokat iterátorként
- keres(const string&):
  - megkeresi az összes olyan suffixot ami az adott karaktersorozattal kezdődik bináris keresés segítségével
  - visszaadja a találatokat iterátorként
- end():
  - visszaad egy iterátort, ami minden kereső iterátornak a végeként szolgál
- operator+():
  - visszaad egy új SuffixArray-t a két szöveg változó egymás után kötve
- operator=():
  - felveszi a kapott SuffixArray értékeit
- operator+=():
  - hozzáfűzi az eredeti szöveg-hez a kapott SuffixArray szöveg-ét
  - meghívja a rendez függvényt
- operator&:
  - megkeresi a két SuffixArray leghosszabb azonos karaktersorozatát
  - meghívja a leghosszabb\_kozos\_substr függvényt
- leghosszabb\_kozos\_substr(string\*, size\_t):
  - megkeresi az első leghosszabb rész karaktersorozatát, ami szerepel a SuffixArray-ban
  - x hosszú részsorozatot keres, ha nem talál akkor rekurzívan keres egy (x-1) hosszú részsorozatot
  - ha nem talál semmilyen egyezést üres stringet ad vissza
- ~SuffixArray:
  - felszabadítja a használt memóriát
- iterator:
  - first\*:
    - az első egyezésre mutató pointer

- a suffix tömbbeli elemre mutat
- last\*:
  - az utolsó egyezésre mutató pointer
  - a suffix tömbbeli elemre mutat
- iterator(string\*, string\*)
  - létrehozza az iterátort
- operator++():
  - tovább lépteti a first mutatót (pre)
- operator++(int):
  - tovább lépteti a first mutatót (post)
- operator==(const iterator&):
  - a két first változó egyezését vizsgálja
- operator!=(const iterator&):
  - a két first változó nem egyezését vizsgálja
- operator\*():
  - visszaad egy konstans mutatót a first változó által mutatott stringre
  - ha first egy nullpointer akkor hibát dob
- operator->():
  - visszaad egy konstans mutatót a first változó által mutatott stringre
  - ha first egy nullpointer akkor hibát dob

## Suffix array – Dokumentáció

A programozói dokumentációt a programkódban leírt doxygen formátumú kommentálás alkotja.

Felhasználói dokumentáció:

A program két részből áll.

Első rész:

A program a bemeneten két bármilyen hosszú szöveget vár el új sorral elválasztva a kettőt.

A program kiírja az összes olyan karaktersorozatot, ami az első szövegben a második szöveggel kezdődik.

Második rész:

A program a bemeneten két bármilyen hosszú szöveget vár el új sorral elválasztva a kettőt.

A program kiírja a leghosszabb karaktersorozatot, ami mindkét szövegben szerepel.

A tesztek a programkódban nem olvasnak be, hanem azt vizsgálják, hogy a megadott bemenetre az elvárt kimenetet produkálják-e.