

Sorok beolvasása állományból és megszámozása

```
#include<iostream>
#include<fstream>
#include<string.h>
using namespace std;
int main()
{
    ifstream be("szoveg.txt");
    char s[300];
    int k=0;
    while (be.getline(s,300)!=0) //amig van mit beolvasni
    {
        k++; // novelem a sorok szamat
        cout<<k<<": "<<s<<"\n"; //kiírom
    }
}
```

Szavak beolvasása állományból: csak a fehér karaktereket tekinti elválasztó jeleknek

```
#include<iostream>
#include<fstream>
#include<string.h>
using namespace std;
int main()
{
    ifstream be("szoveg.txt");
    char s[300];
    int k=0;
    while (be>>s) //amíg van újabb szó
    {
        k++; // növelem a szavak számát
        cout<<k<<": "<<s<<"\n"; //kiírom
    }
}
```

Magánhangzók duplázása vagy cseréje más jelekre

```
#include<iostream>
#include<fstream>
#include<string.h>
using namespace std;
int main()
{   ifstream be("szoveg.txt");
    char s[300], t[300];
    int i, k=0;
    while (be.getline(s,300)!=0)
    {   i=0;
        while(s[i]!=0)
        {   if(strchr("aeiou", s[i])!=0) // ha magánhangzót talál
            {   strcpy(t,s+i+1); // lemásolja az i-től jobbra levő részt
                s[i+1]=s[i]; // megduplázza a magánhangzót
                strcpy(s+i+2,t); // az új jel után bemásolja a félretett részt
                i=i+2; // az új résztől folytatja a keresést
            }
            else i++; // továbblép
        }
        k++; // növeli a sorok számát
        cout<<k<<": "<<s<<"\n";
    }
}
```

Replace: egy mondatban egy szöveg összes előfordulásának cseréje más szövegre

```
#include<iostream>
#include<string.h>
using namespace std;
int main()
{
    char s[300]=" három öt ketto három öt három het három kilenc";
    char t[300], *p;
    char regi[]="három"; // ezt a szöveget keresi
    char uj[]="trei"; // erre cseréli majd le
    cout<<s<<"\n";
    int n=strlen(regi); // keresett rész hossza
    int m=strlen(uj); // cserélt rész hossza
    p=strstr(s, regi); // megkeresi az első előfordulást
    while(p) // amíg van találat
    {
        strcpy(t, p+n); // lemásolja a régi utáni darabot
        strcpy(p,uj); // átírja a régit az újra
        strcpy(p+m, t); // az új után beírja a lemásolt részt
        p=strstr(p+m,regi); // keresi a következő előfordulást
    }
    cout<<s<<"\n"; // kiírja az átlalakított szöveget
}
```

Szavak összegyűjtése és rendezése, gyakoriság megszámlálása

```
#include<iostream>
#include<fstream>
#include<string.h>
using namespace std;
int main()
{ ifstream be("szoveg.txt");
  char x[100][15]; // 100 darab szó, egy szó 14 betüből állhat
  int db[100]={0}; // előfordulások tömbje
  char elv[]=" .?! ,; :-"; // szavakat elválasztó jelek
  char s[15], sor[300], *p;
  int n=0, i, j;
  while(be.getline(sor,300))
  { p=strtok(sor,elv); // megtalálja az első szót
    while(p) // amíg van találat
    { strcpy(x[n],p); // az utolsó pozícióba bemásolja az új szót
      db[n]=1; // most szerepelt elször
      i=0; // elindulok az elejétől
      while(strcmp(x[i],p)!=0) // amíg meg nem találja
      { i++; // továbblép
        if(i<n) db[i]++; // ha korábban már volt, a gyakoriság nő
        else n++; // új szót találtunk
      }
      p=strtok(NULL,elv); // keresi a következő szót
    }
  }
  // szavak rendezése
  for(i=0;i<n-1;i++)
  for(j=i+1;j<n;j++)
  if( strcmp(x[i],x[j])>0) // ha az ABC sorrend nem jó
  { strcpy(s,x[i]); // felcseréli az x[i] és x[j] szavakat
    strcpy(x[i],x[j]);
    strcpy(x[j],s);
    int y=db[i]; // felcseréli a gyakoriságokat is
    db[i]=db[j];
    db[j]=y;
  }
  // kiírja a rendezett sorozatot
  for(i=0;i<n;i++)
  cout<<x[i]<<" "<<db[i]<<"\n";
}
```