

Elemi alkalmazások fejlesztése I/10. Absztrakt file típus

Steingart Ferenc
`stengi@inf.elte.hu`

2002. április 29.

Feladat

Egy szöveges fileból írjuk ki a képernyőre (soronként egyet) a pontosan 5 karakterből álló szavakat. A szavak a file-ban egy vagy több szóközzel illetve sor-emelés karakterrel vannak elválasztva.

Megoldás

Absztrakt file típus bevezetése:

Az eredeti szövegfile, amelyből karaktereket tudunk olvasni (x):

\n		F	ü	r	g	e			r	ó	k	a		l
----	--	---	---	---	---	---	--	--	---	---	---	---	--	---

á	b	a	k	,	\n		s	u	r	r	a	n	ó	
---	---	---	---	---	----	--	---	---	---	---	---	---	---	--

k	i	s		á	r	n	y	a	k
---	---	---	--	---	---	---	---	---	---

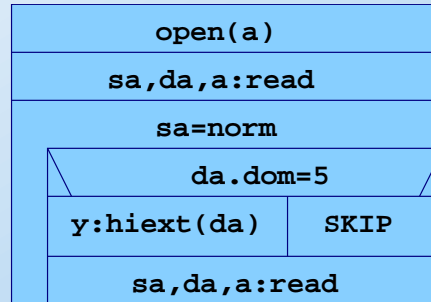
Az absztrakt file, amiből szavakat tudunk olvasni (a):

Fürge	róka	lábak,
-------	------	--------

surránó	kis	árnyak
---------	-----	--------

Absztrakt megoldóprogram

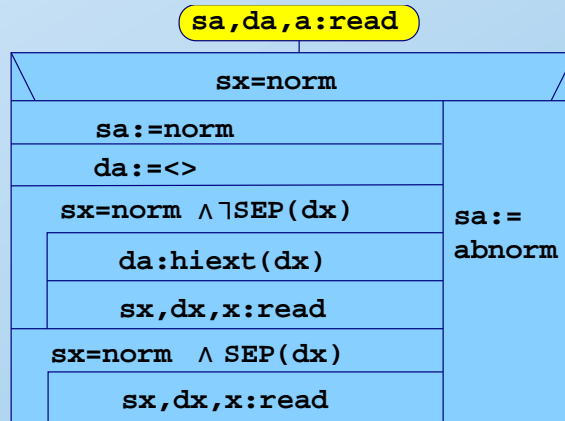
Egyváltozós, egyértékű elemenkénti feldolgozás az absztrakt a file-ra:



Az olvasó művelet

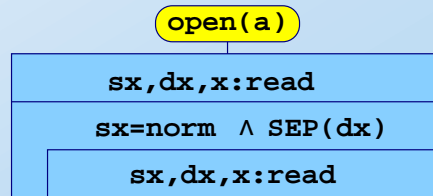
Az olvasó művelet feltételezi az alábbi "típusinvariáns" meglétét:

$sx=abnorm$
 vagy
 dx a következő beolvasandó szó első karaktere



Az `open` művelet

A "típusinvariáns"-t először az `open` művelettel teljesítjük:



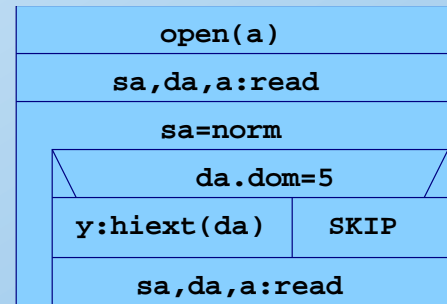
A főprogram futása

\n		F	ü	r	g	e			r	ó	k	a		l
----	--	---	---	---	---	---	--	--	---	---	---	---	--	---

á	b	a	k	,	\n		s	u	r	r	a	n	ó	
---	---	---	---	---	----	--	---	---	---	---	---	---	---	--

k	i	s		á	r	n	y	a	k
---	---	---	--	---	---	---	---	---	---

sx	dx	sa	da
—	—	—	—



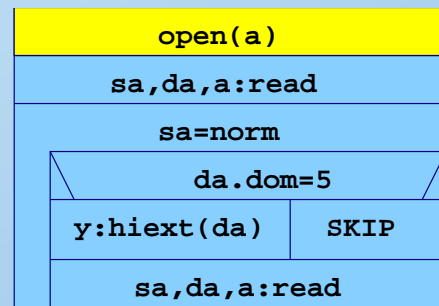
A főprogram futása

ü	r	g	e			r	ó	k	a		l
---	---	---	---	--	--	---	---	---	---	--	---

á	b	a	k	,	\n		s	u	r	r	a	n	ó	
---	---	---	---	---	----	--	---	---	---	---	---	---	---	--

k	i	s		á	r	n	y	a	k
---	---	---	--	---	---	---	---	---	---

sx	dx	sa	da
norm	'F'	—	—



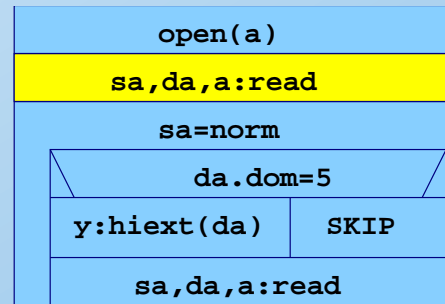
A főprogram futása

ó	k	a		l
---	---	---	--	---

á	b	a	k	,	\n		s	u	r	r	a	n	ó	
---	---	---	---	---	----	--	---	---	---	---	---	---	---	--

k	i	s		á	r	n	y	a	k
---	---	---	--	---	---	---	---	---	---

sx	dx	sa	da
norm	'r'	norm	"Fürge"



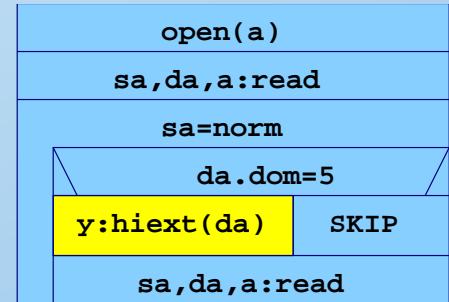
A főprogram futása

ó	k	a		l
---	---	---	--	---

á	b	a	k	,	\n		s	u	r	r	a	n	ó	
---	---	---	---	---	----	--	---	---	---	---	---	---	---	--

k	i	s		á	r	n	y	a	k
---	---	---	--	---	---	---	---	---	---

sx	dx	sa	da
norm	'r'	norm	"Fürge"

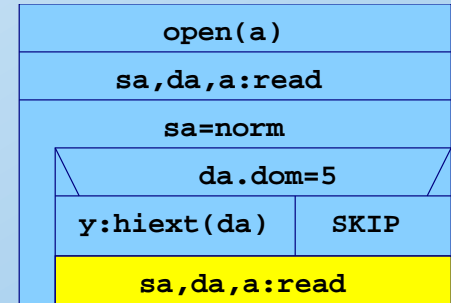


A főprogram futása

á	b	a	k	,	\n		s	u	r	r	a	n	ó	
---	---	---	---	---	----	--	---	---	---	---	---	---	---	--

k	i	s		á	r	n	y	a	k
---	---	---	--	---	---	---	---	---	---

sx	dx	sa	da
norm	'l'	norm	"róka"

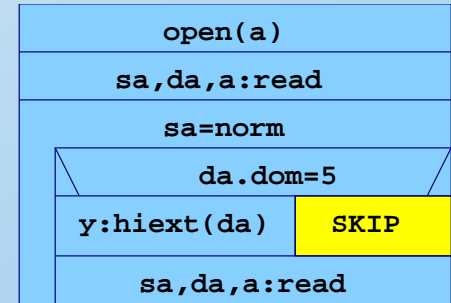


A főprogram futása

á	b	a	k	,	\n		s	u	r	r	a	n	ó	
---	---	---	---	---	----	--	---	---	---	---	---	---	---	--

k	i	s		á	r	n	y	a	k
---	---	---	--	---	---	---	---	---	---

sx	dx	sa	da
norm	'l'	norm	"róka"

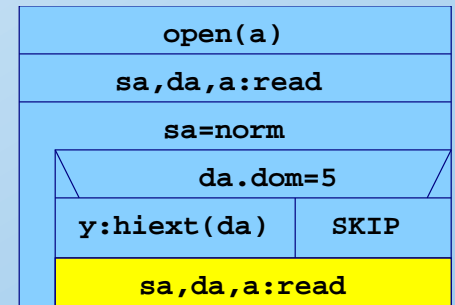


A főprogram futása

u	r	r	a	n	ó	
---	---	---	---	---	---	--

k	i	s		á	r	n	y	a	k
---	---	---	--	---	---	---	---	---	---

sx	dx	sa	da
norm	's'	norm	"lábak,"

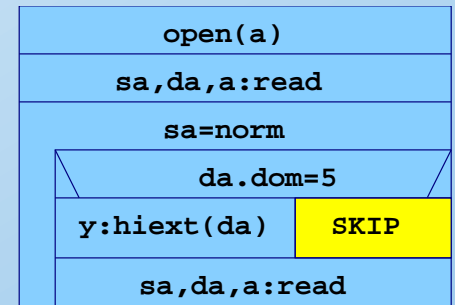


A főprogram futása

u	r	r	a	n	ó	
---	---	---	---	---	---	--

k	i	s		á	r	n	y	a	k
---	---	---	--	---	---	---	---	---	---

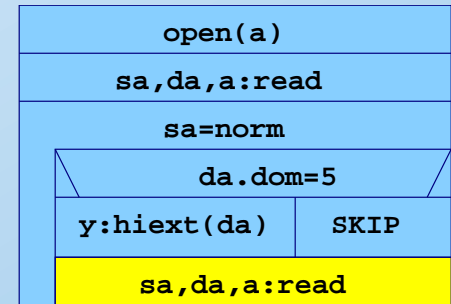
sx	dx	sa	da
norm	's'	norm	"lábak,"



A főprogram futása

i	s		á	r	n	y	a	k
---	---	--	---	---	---	---	---	---

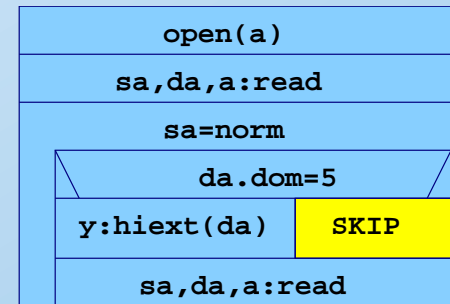
sx	dx	sa	da
norm	'k'	norm	"surranó"



A főprogram futása

i	s		á	r	n	y	a	k
---	---	--	---	---	---	---	---	---

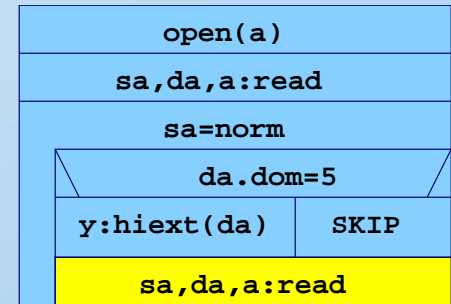
sx	dx	sa	da
norm	'k'	norm	"surranó"



A főprogram futása

r	n	y	a	k
---	---	---	---	---

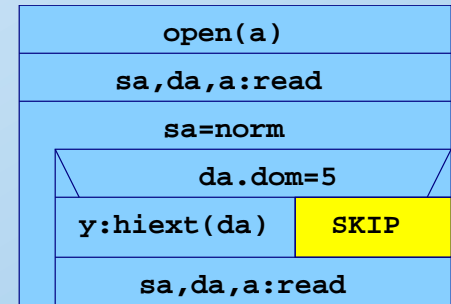
sx	dx	sa	da
norm	'á'	norm	"kis"



A főprogram futása

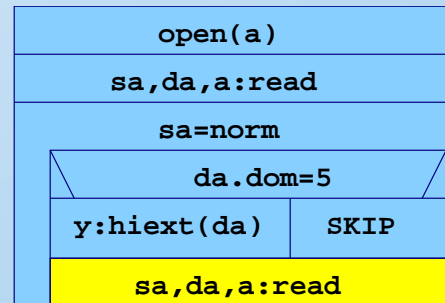
r	n	y	a	k
---	---	---	---	---

sx	dx	sa	da
norm	'á'	norm	"kis"



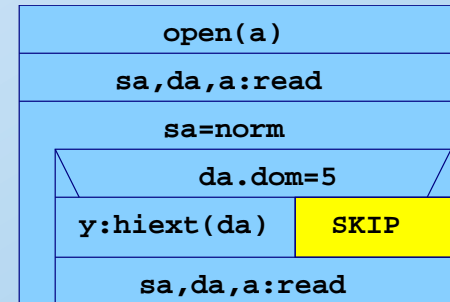
A főprogram futása

sx	dx	sa	da
abnorm	—	norm	"árnyak"



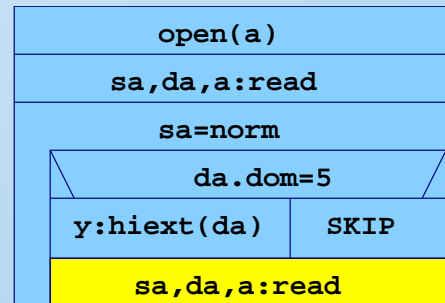
A főprogram futása

sx	dx	sa	da
abnorm	—	norm	"árnyak"



A főprogram futása

sx	dx	sa	da
abnorm	—	abnorm	—



Megoldás C++-ban

A kódolás során elvégzendő feladatok:

- az absztrakt file típus definiálása osztállyal
- a főprogram (`main` függvény) kódolása
- a típusműveletek implementációja

A file típus

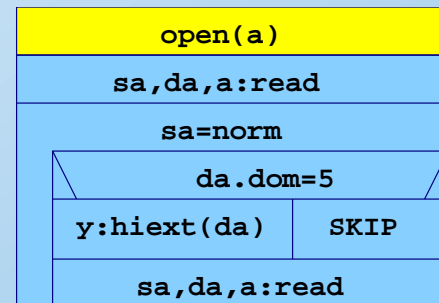
```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

class A{
    ifstream x;
    bool      sx;
    char      dx;
public:
    A(const string& fn);
    void open();
    void read(bool& sa, string& da);
};
```

A főprogram

```
int main(int argc, char *argv[])
{
    bool sai;
    string da;
    A a("x.txt");

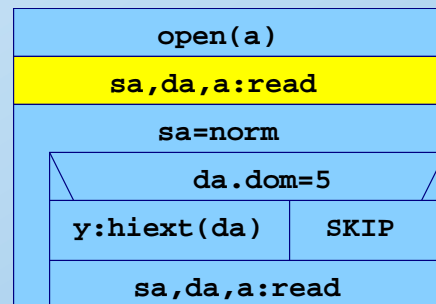
    a.open();
    a.read(sa, da);
    while(sa){
        if(da.length()==5){
            cout << da << endl;
        }
        a.read(sa, da);
    }
    return 0;
}
```



A főprogram

```
int main(int argc, char *argv[])
{
    bool sa;
    string da;
    A a("x.txt");

    a.open();
    a.read(sa, da);
    while(sa){
        if(da.length()==5){
            cout << da << endl;
        }
        a.read(sa, da);
    }
    return 0;
}
```



A főprogram

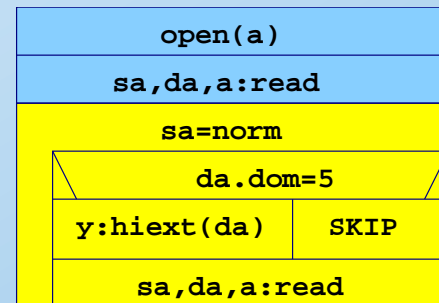
```

int main(int argc, char *argv[])
{
    bool sa;
    string da;
    A a("x.txt");

    a.open();
    a.read(sa, da);

    while(sa){
        if(da.length()==5){
            cout << da << endl;
        }
        a.read(sa, da);
    }
    return 0;
}

```



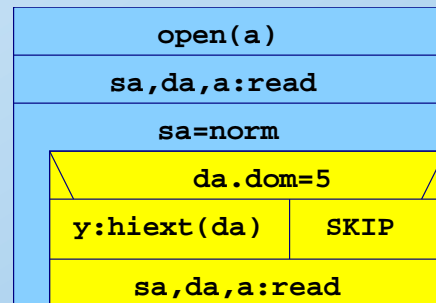
A főprogram

```

int main(int argc, char *argv[])
{
    bool sa;
    string da;
    A a("x.txt");

    a.open();
    a.read(sa, da);
    while(sa){
        if(da.length()==5){
            cout << da << endl;
        }
        a.read(sa, da);
    }
    return 0;
}

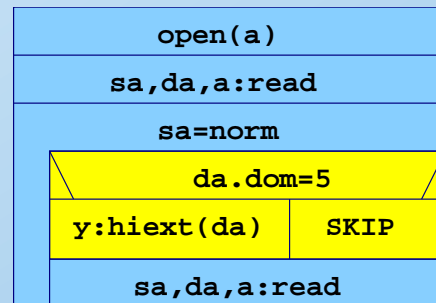
```



A főprogram

```
int main(int argc, char *argv[])
{
    bool sa;
    string da;
    A a("x.txt");

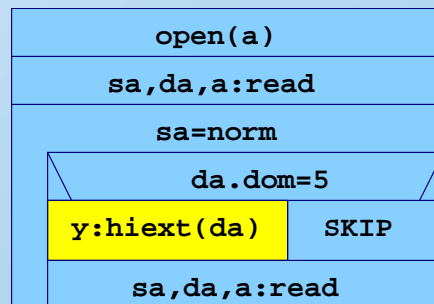
    a.open();
    a.read(sa, da);
    while(sa){
        if(da.length()==5){
            cout << da << endl;
        }
        a.read(sa, da);
    }
    return 0;
}
```



A főprogram

```
int main(int argc, char *argv[])
{
    bool sa;
    string da;
    A a("x.txt");

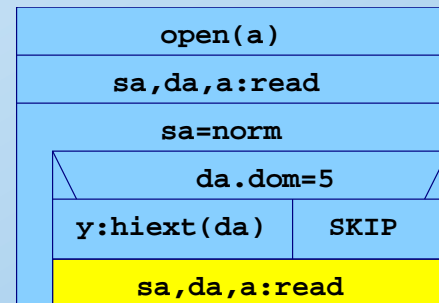
    a.open();
    a.read(sa, da);
    while(sa){
        if(da.length()==5){
            cout << da << endl;
        }
        a.read(sa, da);
    }
    return 0;
}
```



A főprogram

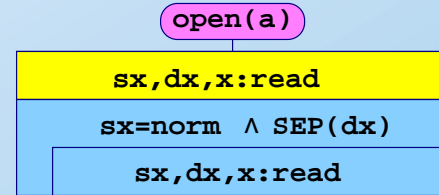
```
int main(int argc, char *argv[])
{
    bool sa;
    string da;
    A a("x.txt");

    a.open();
    a.read(sa, da);
    while(sa){
        if(da.length()==5){
            cout << da << endl;
        }
        a.read(sa, da);
    }
    return 0;
}
```



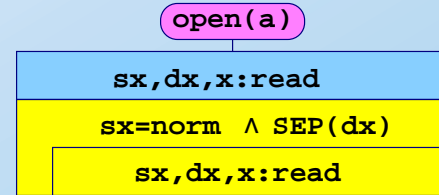
Az open művelet

```
void A::open()
{
    x.get(dx);
    sx = !x.eof();
    while(sx && (dx==' ' || dx=='\n')){
        x.get(dx);
        sx = !x.eof();
    }
}
```



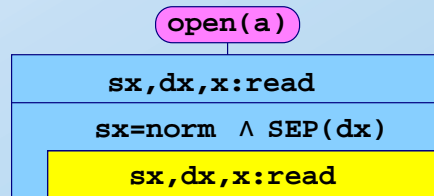
Az open művelet

```
void A::open()
{
    x.get(dx);
    sx = !x.eof();
    while(sx && (dx==' ' || dx=='\n')){
        x.get(dx);
        sx = !x.eof();
    }
}
```



Az open művelet

```
void A::open()
{
    x.get(dx);
    sx = !x.eof();
    while(sx && (dx==' ' || dx=='\n')){
        x.get(dx);
        sx = !x.eof();
    }
}
```

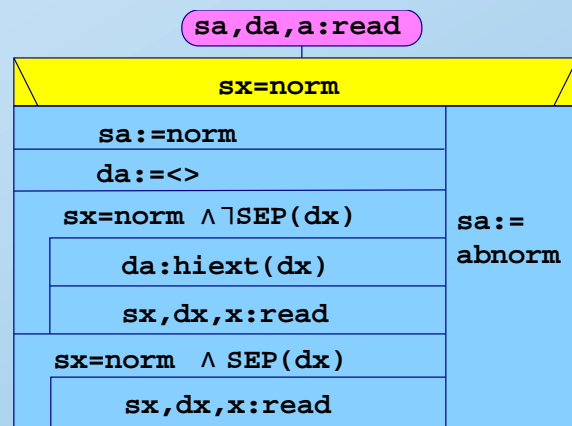


A read művelet

```

void A::read(bool& sa, string& da)
{
    if(sx){
        sa = true;
        da = "";
        while(sx && dx!=' ' && dx!='\n'){
            da+=dx;
            x.get(dx);
            sx = !x.eof();
        }
        while(sx && (dx==' ' || dx=='\n')){
            x.get(dx);
            sx = !x.eof();
        }
    }else{
        sa=false;
    }
}

```

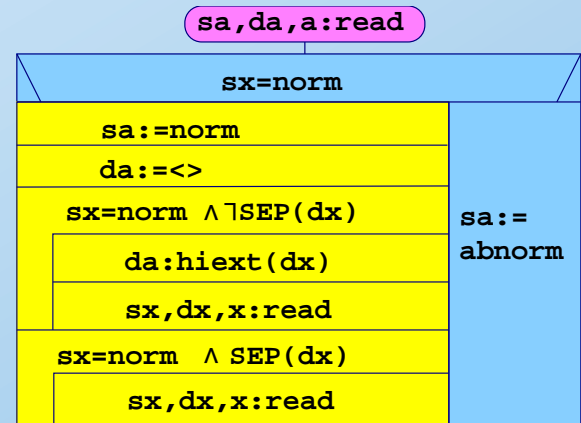


A read művelet

```

void A::read(bool& sa, string& da)
{
    if(sx){
        sa = true;
        da = "";
        while(sx && dx!=' ' && dx!='\n'){
            da+=dx;
            x.get(dx);
            sx = !x.eof();
        }
        while(sx && (dx==' ' || dx=='\n')){
            x.get(dx);
            sx = !x.eof();
        }
    }else{
        sa=false;
    }
}

```

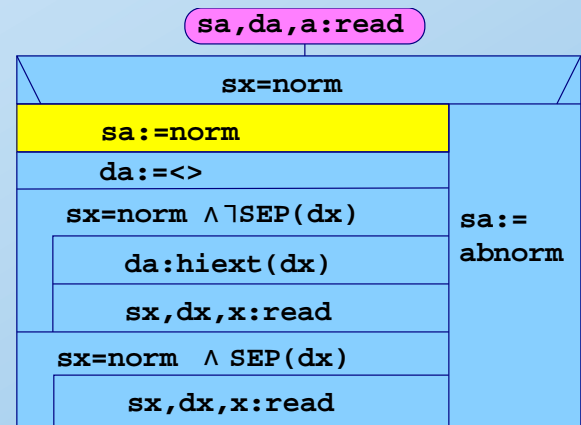


A read művelet

```

void A::read(bool& sa, string& da)
{
    if(sx){
        sa = true;
        da = "";
        while(sx && dx!=' ' && dx!='\n'){
            da+=dx;
            x.get(dx);
            sx = !x.eof();
        }
        while(sx && (dx==' ' || dx=='\n')){
            x.get(dx);
            sx = !x.eof();
        }
    }else{
        sa=false;
    }
}

```

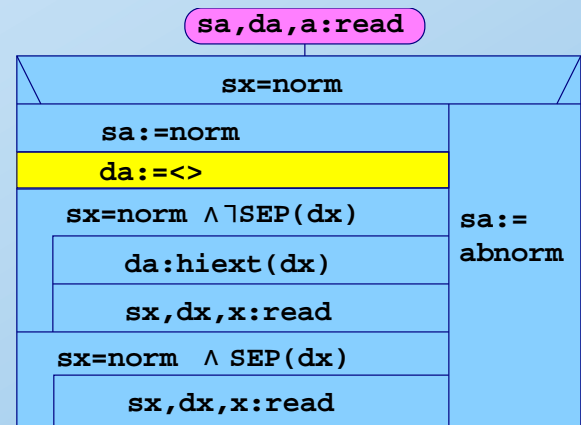


A read művelet

```

void A::read(bool& sa, string& da)
{
    if(sx){
        sa = true;
        da = "";
        while(sx && dx!=' ' && dx!='\n'){
            da+=dx;
            x.get(dx);
            sx = !x.eof();
        }
        while(sx && (dx==' ' || dx=='\n')){
            x.get(dx);
            sx = !x.eof();
        }
    }else{
        sa=false;
    }
}

```

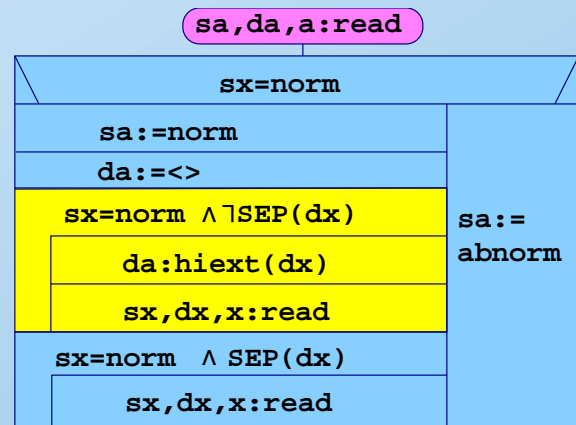


A read művelet

```

void A::read(bool& sa, string& da)
{
    if(sx){
        sa = true;
        da = "";
        while(sx && dx!=' ' && dx!='\n'){
            da+=dx;
            x.get(dx);
            sx = !x.eof();
        }
        while(sx && (dx==' ' || dx=='\n')){
            x.get(dx);
            sx = !x.eof();
        }
    }else{
        sa=false;
    }
}

```

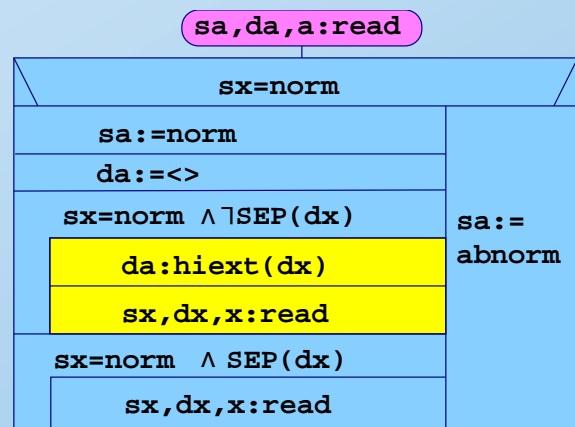


A read művelet

```

void A::read(bool& sa, string& da)
{
    if(sx){
        sa = true;
        da = "";
        while(sx && dx!=' ' && dx!='\n'){
            da+=dx;
            x.get(dx);
            sx = !x.eof();
        }
        while(sx && (dx==' ' || dx=='\n')){
            x.get(dx);
            sx = !x.eof();
        }
    }else{
        sa=false;
    }
}

```

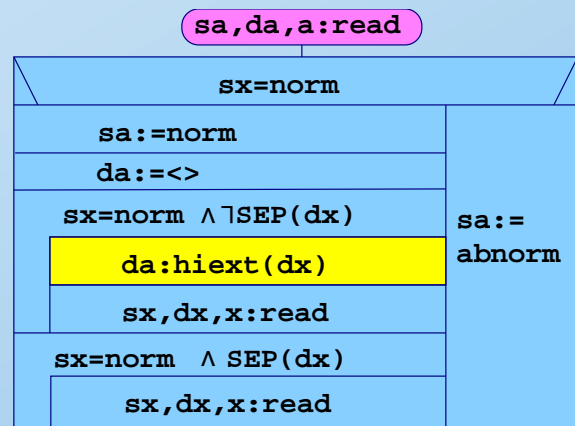


A read művelet

```

void A::read(bool& sa, string& da)
{
    if(sx){
        sa = true;
        da = "";
        while(sx && dx!=' ' && dx!='\n'){
            da+=dx;
            x.get(dx);
            sx = !x.eof();
        }
        while(sx && (dx==' ' || dx=='\n')){
            x.get(dx);
            sx = !x.eof();
        }
    }else{
        sa=false;
    }
}

```

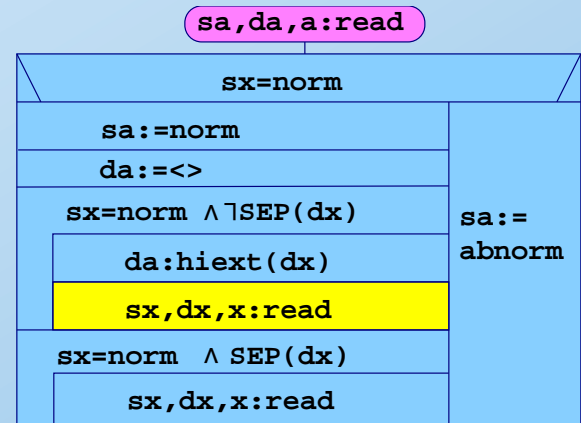


A read művelet

```

void A::read(bool& sa, string& da)
{
    if(sx){
        sa = true;
        da = "";
        while(sx && dx!=' ' && dx!='\n'){
            da+=dx;
            x.get(dx);
            sx = !x.eof();
        }
        while(sx && (dx==' ' || dx=='\n')){
            x.get(dx);
            sx = !x.eof();
        }
    }else{
        sa=false;
    }
}

```

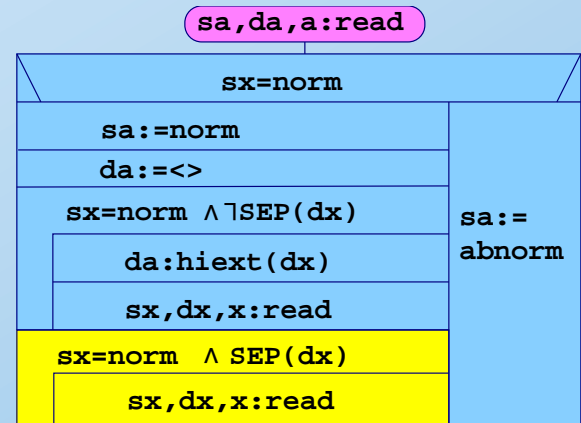


A read művelet

```

void A::read(bool& sa, string& da)
{
    if(sx){
        sa = true;
        da = "";
        while(sx && dx!=' ' && dx!='\n'){
            da+=dx;
            x.get(dx);
            sx = !x.eof();
        }
        while(sx && (dx==' ' || dx=='\n')){
            x.get(dx);
            sx = !x.eof();
        }
    }else{
        sa=false;
    }
}

```

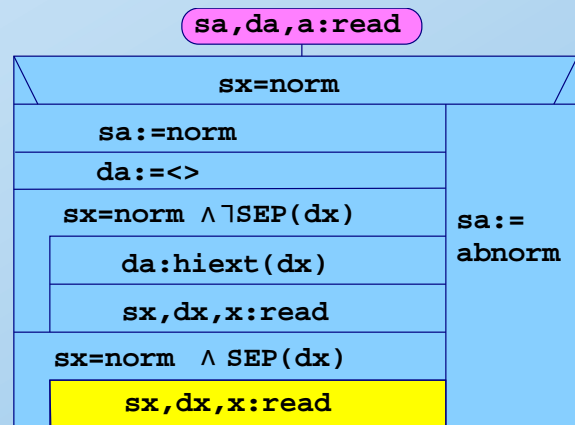


A read művelet

```

void A::read(bool& sa, string& da)
{
    if(sx){
        sa = true;
        da = "";
        while(sx && dx!=' ' && dx!='\n'){
            da+=dx;
            x.get(dx);
            sx = !x.eof();
        }
        while(sx && (dx==' ' || dx=='\n')){
            x.get(dx);
            sx = !x.eof();
        }
    }else{
        sa=false;
    }
}

```

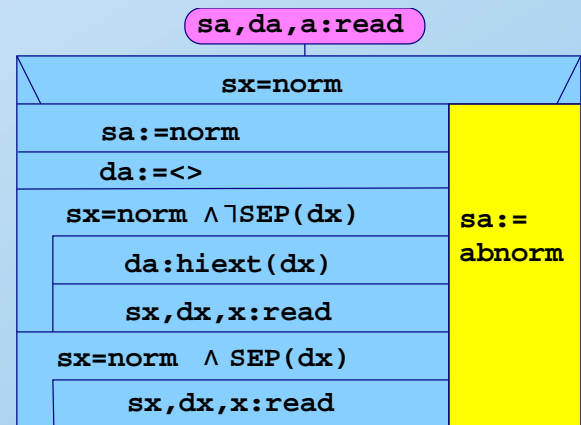


A read művelet

```

void A::read(bool& sa, string& da)
{
    if(sx){
        sa = true;
        da = "";
        while(sx && dx!=' ' && dx!='\n'){
            da+=dx;
            x.get(dx);
            sx = !x.eof();
        }
        while(sx && (dx==' ' || dx=='\n')){
            x.get(dx);
            sx = !x.eof();
        }
    }else{
        sa=false;
    }
}

```



4. beadandó feladatok

1. Adott egy szövegfile. Számoljuk meg, hogy a szöveg hány szóból áll, ha a 12 jelnél hosszabb szavakat két szónak tekintjük! (A szavakat tetszőleges számú szóköz és sorvégjel választhatja el.)
2. Adott egy szövegfile. Számoljuk meg, hogy a szövegben hány nagybetűvel kezdődő szó van! (A szavakat tetszőleges számú szóköz és sorvégjel választhatja el.)
3. Adott egy szövegfile. Állapítsuk meg, hogy hány olyan szó van a szövegben, ami tartalmaz "R" betűt! (A szavakat tetszőleges számú szóköz és sorvégjel választhatja el.)

A file nevét a program parancssorában kell megadni. A feladatot absztrakt file típus bevezetésével kell megoldani.

VÉGE