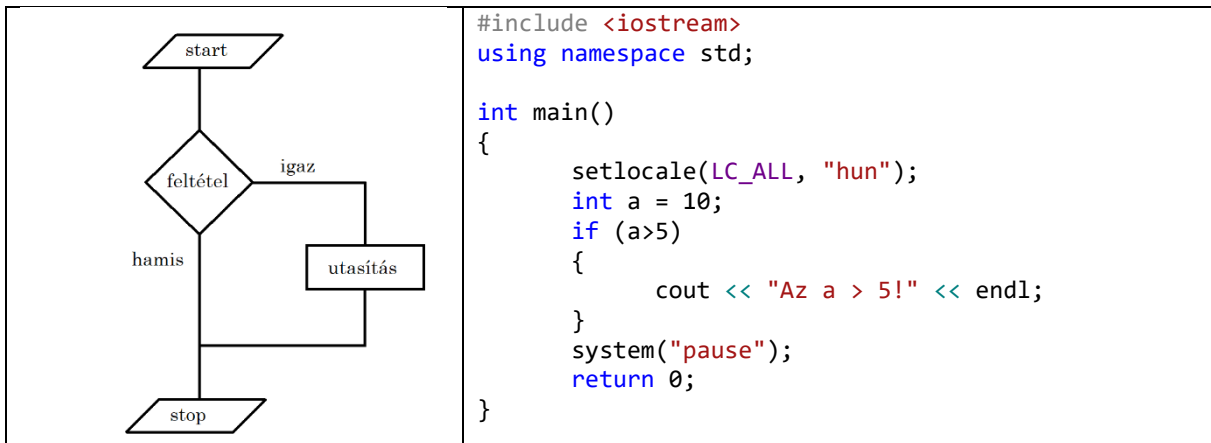


7. A C++ nyelv vezérlési szerkezetei

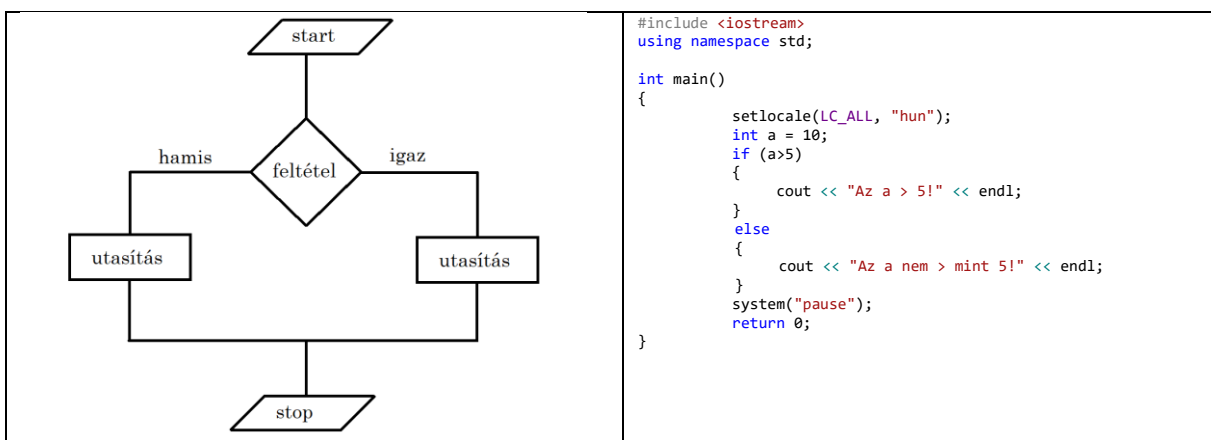
7.1 if

Az 1. vezérlési szerkezet amit tanulmányozni fogunk az `if()` -lesz. Ezt a vezérlési szerkezetet elágazásnak is szokás hívni. Értsük meg először a folyamat ábrát.



A program végrehajtása amikor megérkezik az `if` vezérlési szerkezethez, a belépési feltétel kiértékelődik és annak igazság tartalma szerint tovább halad a végrehajtás az igaz vagy a hamis ág felé. Az `a > 5` szöveg csak akkor íródik ki ha az `a` változó tényleg nagyobb mint 5.

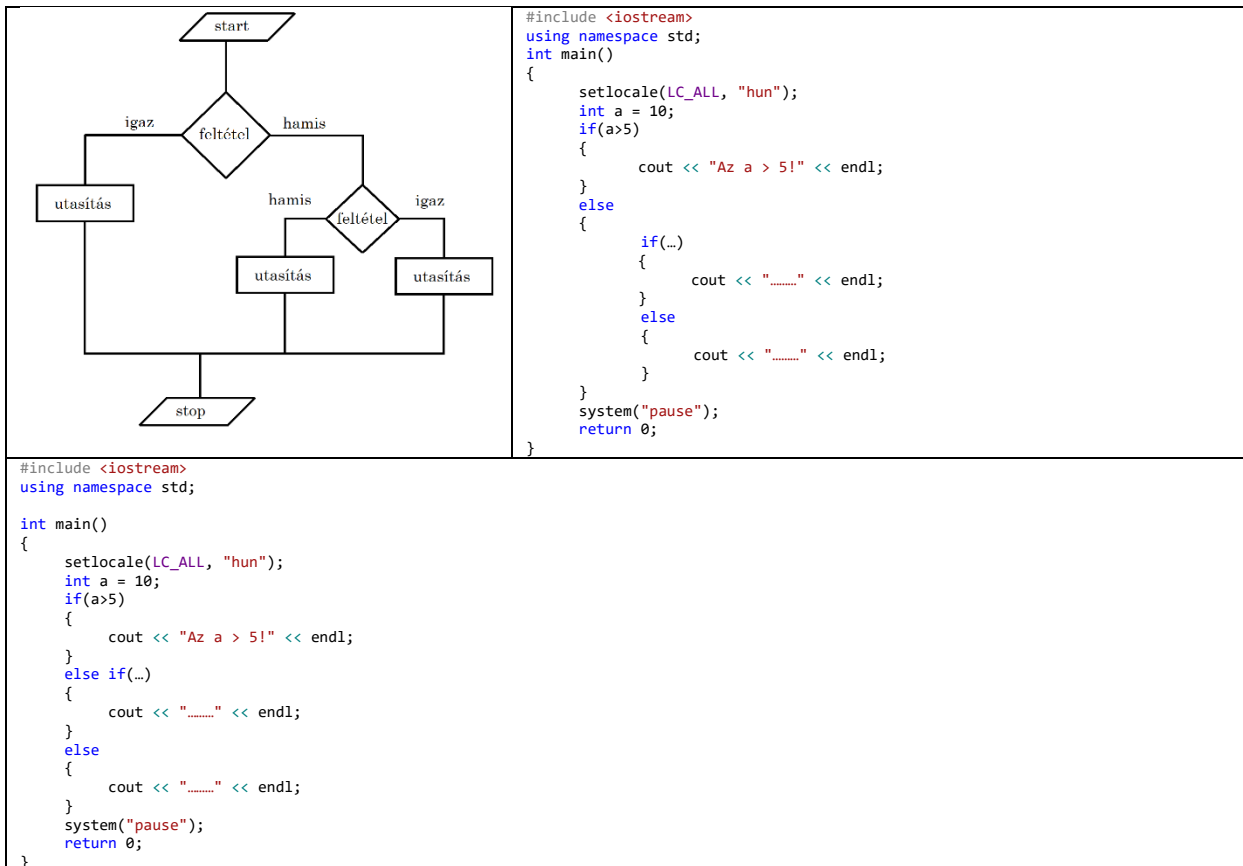
7.2 if-else



7.3 egymásba ágyazott elágazások

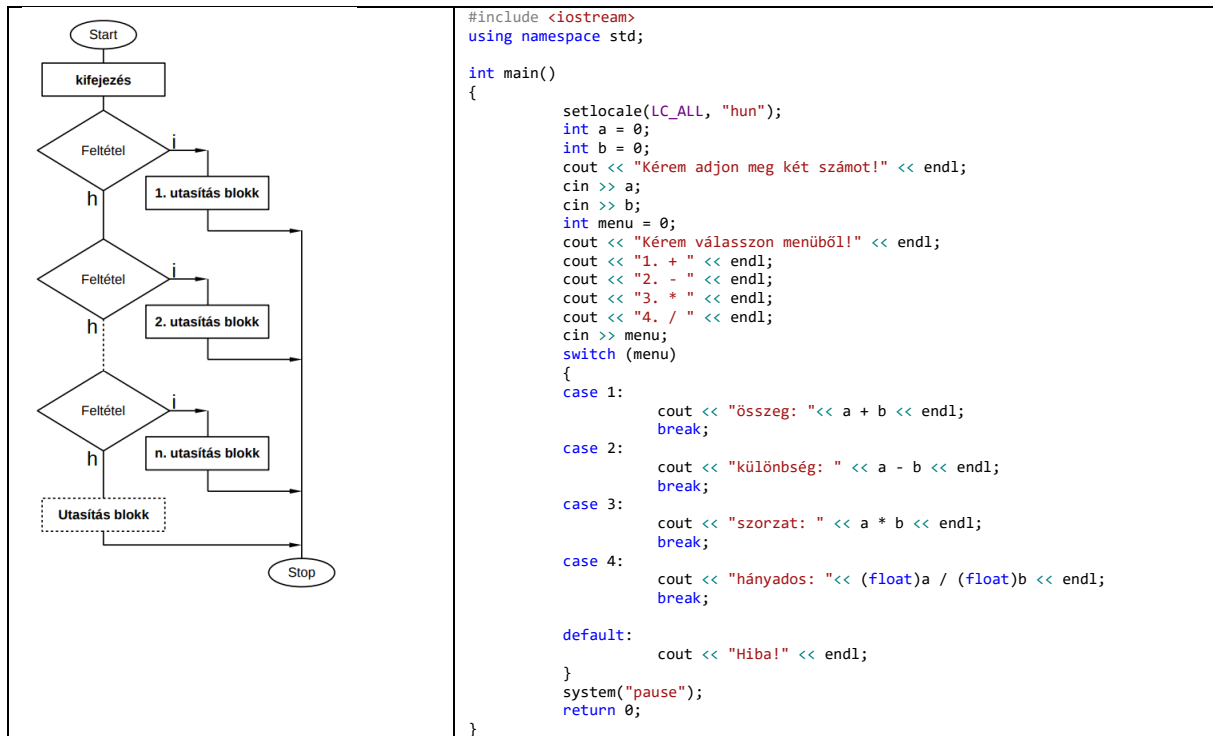
Egy elágazás akár igaz, akár a hamis ágába egyaránt további elágazások illeszthetők be.

if-else-if



7.4 switch-case

A többirányú elágazás másnéven összetett elágazás több végrehajtási ággal rendelkezik, és egyetlen kifejezés kerül kiértékelésre, melynek eredménye alapján kerül az egyes ágakra a vezérlés. Ha végrehajtódott az adott ág, akkor a **break** utasítás miatt a vezérlési szerkezet végére kerül a végrehajtás.



A cin >> utasítás segítségével (consol in) bekérhetünk egy számot a felhasználtól.

```
int menu = 0;
cin >> menu;
```

A default ágba akkor kerül a végrehajtás ha egyik esetnek, „case” -nek megfelelő ág sem jön létre a belépési feltételben.

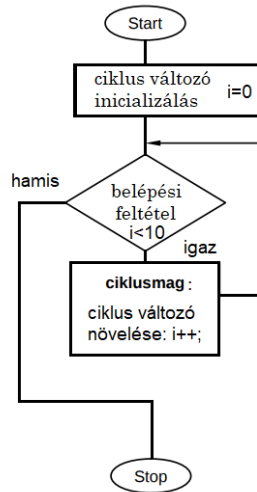
<pre> #include <iostream> using namespace std; int main() { setlocale(LC_ALL, "hun"); int a,b = 0; cout << "Kérem adjon meg két számot!" << endl; cin >> a; cin >> b; int menu = 0; cout << "Kérem válasszon menüből!" << endl; cout << "1. + " << endl; cout << "2. - " << endl; cout << "3. * " << endl; cout << "4. / " << endl; cin >> menu; switch (menu) { case 1: cout << "összeg: " << a + b << endl; case 2: cout << "különbség: " << a - b << endl; break; case 3: cout << "szorzat: " << a * b << endl; break; case 4: cout << "hányados: " << (float)a / (float)b << endl; break; default: cout << "Hiba!" << endl; } system("pause"); return 0; } </pre>	<p>Az itt látható kódban a case 1 ágban nincs break utasítás. Ez nem túl célszerű mert ha a case 1 kerül végrehajtásra akkor a case 2 is végrehajtódik pedig azt nem választottuk ki.</p>
---	--

7.5 ciklus

elől tesztelő ciklus: while, for

hátral tesztelő ciklus: do - while

Az előtesztelési ciklus először megvizsgálja, hogy a feltétel teljesül-e. Ha igen, akkor végrehajtja a ciklusmagot, majd a folyamat az újból kezdődik. Ha nem, akkor a program a ciklus utáni ponton folytatódik tovább, azaz a **ciklusmag** kimarad.



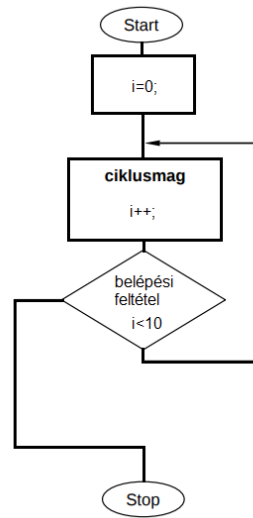
```

#include <iostream>
using namespace std;
int main()
{
    setlocale(LC_ALL, "hun");
    int size = 10;
    int i = 0;
    while (i < size)
    {
        cout << "számok: " << i << endl;
        i++;
    }
    system("pause");
    return 0;
}
  
```

```

#include <iostream>
using namespace std;
int main()
{
    setlocale(LC_ALL, "hun");
    int size=10;
    for (int i = 0; i < size; i++)
    {
        cout << "számok: " << i << endl;
    }
    system("pause");
    return 0;
}
  
```

A hátulatesztelési ciklus először végrehajtja a ciklusmagot utána megvizsgálja, hogy a feltétel teljesül-e.



```

#include <iostream>
using namespace std;
int main()
{
    setlocale(LC_ALL, "hun");
    int size = 10;
    int i = 0;
    do
    {
        cout << "számok: " << i << endl;
        i++;
    }
    while (i < size);
    system("pause");
    return 0;
}
  
```