**ლექცია 11. მრავალგანზომილიებიანი მასივი. მასივის ფუნქციები. ფაილების ჩართვა**

**მრავალგანზომილებიანი მასივი**

მასივის ელემენტი სკალარული ცვლადის გარდა შეიძლება იყოს ასევე თავად მასივი. ორგან­ზომილებიანი მასივი შეიძლება წარმოვიდგინოთ როგორც მატრიცა ან ცხრილი, რომელსაც სიგანე და სიმაღლე გააჩნია. მოვიყვანოთ მაგალითი ბუბას ავტონაწილების მაღაზიის საქონელის ორგან­ზომილებიან მასივად წარმოდგენისა, რომლის ყოველი სტრიქონი არის ცალკეული საქონელის წარმოდგენა, ხოლო ამ სტრიქონის ყოველი სვეტი შესაბამისი საქონელის ატრიბუტი:

|  |  |  |
| --- | --- | --- |
| **კოდი**  საქონელილი | **აღწერა** | **ფასი** |
| TIR | საბურავები | 100 |
| OIL | ზეთი | 10 |
| SPK | აალების სანთლები | 4 |

ნახატი 1. ორგანზომილებიანი მასივი საქონელის შესახებ

საქონლის ატრიბუტი

ნახატ 1-ში მოცემული მონაცემების ჩაწერა ორგანზომილებიან მასივში შესაძლებელია, შემდეგი კოდის სახით:

ლისტინგი 1 ორგანზომილებიანი მასივის შექმნა

$products=array(array('TIR','საბურავები',100),

array('OIL','ზეთი',10),

array('SPK','აალების სანთლები','4));

ლისტინგი 1-დან ჩანს, რომ მასივი $products შეიცავს სამ მასივს. მასივის ელემენტის წვდომის­თვის საჭიროა მივუთითოთ მასივის სახელი და ელემენტის ინდექსი, ზოგადად n-განზომილებიან მასივს ექნება n ინდექსი. ჩვენს მაგალითში გვაქვს ორი ინდექსი, მაგრამ არ უნდა დაგვავიწყდეს, რომ ინდექსების ნუმერაცია ნულიდან იწყება. ლისტინგი 1-ში ჩუმათობის პრინცით მასივი ინდექსირებულია - ანუ რიცხვითი და ავტონატურად იწყებქ ინდექსაცია ნულიდან. ამ მასივის ელემენტების მნიშვნელობების გამოსატანად, შეგვიძლია უშუალოდ მივმართოთ თითეულ ელემენტს, შემდეგი თანამიმდევრობით:

ლისტინგი 2. მასივის ელემენტების ცალ-ცაკე გამოტანა

echo'|'.$products[0][0].'|'.$products[0][1].'|'.$products[0][2].'|<br>';

echo'|'.$products[1][0].'|'.$products[1][1].'|'.$products[1][2].'|<br>';

echo'|'.$products[2][0].'|'.$products[2][1].'|'.$products[2][2].'|<br>';

იგივე შედეგის მისაღებად შეიძლება გამოვიყენოთ ორი ჩადგმული ციკლი

ლისტინგი 3. მასივის ელემენტების ციკლში გამოტანა

for($row=0;$row<3;$row++){

for($column=0;$column<3;$column++){

echo'|'.$products[$row][$column];

}

echo'|,br>';

}

ლისტინგი 2-სა და 3-ის შედეგი ერთიდა იგივეა:

TIR|საბურავები|100|  
|OIL|ზეთი|10|  
|SPK|აალების სანთლები|4|  
  
|TIR|საბურავები|100|  
|OIL|ზეთი|10|  
|SPK|აალების სანთლები|4|

გავაერთიანოთ ლისტინგები 1,2,3 და ჩავწეროთ

ლისტინგი 4 ორგანზომილებიანი მასივის შექმნა და მისი ელემენტების გამოტანა

<!DOCTYPE html>

<html>

<body>

<?php

$products=array(array('TIR','საბურავები',100),

array('OIL','ზეთი',10),

array('SPK','აალების სანთლები',4));

echo'<br>';

echo'|'.$products[0][0].'|'.$products[0][1].'|'.$products[0][2].'|<br>';

echo'|'.$products[1][0].'|'.$products[1][1].'|'.$products[1][2].'|<br>';

echo'|'.$products[2][0].'|'.$products[2][1].'|'.$products[2][2].'|<br>';

echo'<br>';

for ($row=0;$row < 3;$row++){

for ($column=0;$column < 3;$column++){

echo'|'.$products[$row][$column];

}

echo'|<br/>';

}

?>

</body>

</html>

არაა აუცილებელი გავჩერდეთ ორგანზომილებიან მასივზე, ჩვენს მიერ მოყვანილ მაგალითში მისი ელემენტები შეიძლება შეიცავდნენ ასევე მასივებს. ამდენად შეგვიძლია მივიღოთ სამგანზო­მილებიანი მასივი, რომელიც ხასიათდება სიგანით, სიმაღლით და სიღრმით. თუ კი ორგანზომი­ლებიანი მასივის წარმოდგენა მისახერხებელია ცხრილის სახით, მაშინ სამგანზომილებიანი მასი­ვი, არის ასეთი ცხრილების ერთობლიობა შრეების სახით. სამგანზო­მილებიანი მასივის ყოველი ელემენტის წვდომისასს უნდა მივუთითოდ მასივის სახელი, ფენა, სტრიქონი და სვეტი. თუ კი ბუბას ექნება სურვილი, რომ მისი საქონელი დალაგებული იყოს კატეგორიების მიხედვით, მაშინ ისინი შეიძლება შევინახოთ სამგანზომილებიან მასივში.

|  |  |  |
| --- | --- | --- |
| **სათადარიგო ნაწილები სატვირთო ა/მ** | | |
| **კოდი** | **აღწერა** | **ფასი** |

|  |  |  |
| --- | --- | --- |
| **სათადარიგო ნაწილები ფურგუნი ა/მ** | | |
| **კოდი** | **აღწერა** | **ფასი** |

|  |  |  |
| --- | --- | --- |
| **სათადარიგო ნაწილები მსუბუქი ა/მ** | | |
| **კოდი**  საქონელილი | **აღწერა** | **ფასი** |
| TIR | საბურავები | 100 |
| OIL | ზეთი | 10 |
| SPK | აალების სანთლები | 4 |

ნახატი 2. სამგანზომილებიანი მასივით საქონელი დახარისხებულია

საქონლის ატრიბუტი

კატეგორიებად

მოვიყვანოთ კოდი, რომელიც ამ მასივს განსაზღვრავს:

ლისტინგი 5 სამგანზომილებიანი მასივის შექმნა

<?php

$categories=array(array(array('CAR-TIR','საბურავები',100),

array('CAR-OIL','ზეთი',10),

array('CAR-SPK','აალების სანთლები',4)

)

array(array('VAN-TIR','საბურავები',100),

array('VAN-OIL','ზეთი',10),

array('VAN-SPK','აალების სანთლები',4)

)

array(array('TRK-TIR','საბურავები',100),

array('TRK-OIL','ზეთი',10),

array('TRK-SPK','აალების სანთლები',4)

);

?>

თუ კი გვსურს ამ მასივის ელემენტების გამოტანა, გამოვიყენოთ ჩადგმული for ციკლები, მით უფრო, რომ საქმე გვაქვს ინდექსირებულ მასივებთან:

ლისტინგ 6 სამგანზომილებიანი მასივის გამოტანა

<?php

for ($shre=0;$shre < 3;$shre++){

echo "ფენა $shre<br>";

for ($row=0;$row < 3;$row++){

for ($column=0;$column < 3;$column++){

echo'|'.$categorie[$shre][$row][$column];

}

echo'|<br/>';

}

}

?>

გავაერთიანოთ ლისტინგები 5 და 6, გვექნება სამგანზოილებიანი მასივის შექმნსც და გამოტანაც

ლისტინგი 7 სამგანზომილებიანი მასივის შექმნა და გამოტანა

<!DOCTYPE html>

<html>

<body>

<?php

$categories=array(array(array('CAR-TIR','საბურავები',100),

array('CAR-OIL','ზეთი',10),

array('CAR-SPK','აალების სანთლები',4)

)

array(array('VAN-TIR','საბურავები',100),

array('VAN-OIL','ზეთი',10),

array('VAN-SPK','აალების სანთლები',4)

)

array(array('TRK-TIR','საბურავები',100),

array('TRK-OIL','ზეთი',10),

array('TRK-SPK','აალების სანთლები',4)

);

echo'<br>';

for ($shre=0;$shre < 3;$shre++){

echo "ფენა $shre<br>";

for ($row=0;$row < 3;$row++){

for ($column=0;$column < 3;$column++){

echo'|'.$categorie[$shre][$row][$column];

}

echo'|<br/>';

}

}

?>

</body>

</html>

**მასივების დალაგება**

## მასივის ელემენტები შესაძლებელია დალაგებული იყოს ანბანის ან რიცხვების მხრივ, კლება­დობით ან ზრდადობით. ანბანის (სტრინგით) დალაგება ხდება სიმბოლოების ASCII -კოდების საშუალებით (ASCII - American Standard Code for Information Interchange).

ცხრილი 1. გაფართოებული ASCII -კოდები

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **.0** | **.1** | **.2** | **.3** | **.4** | **.5** | **.6** | **.7** | **.8** | **.9** | **.A** | **.B** | **.C** | **.D** | **.E** | **.F** |
| **0.** | NUL | SOM | EOA | EOM | EQT | WRU | RU | BELL | BKSP | HT | LF | VT | FF | CR | SO | SI |
| **1.** | DC0 | DC1 | DC2 | DC3 | DC4 | ERR | SYNC | LEM | S0 | S1 | S2 | S3 | S4 | S5 | S6 | S7 |
| **2.** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **3.** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **4.** | BLANK | ! | " | # | $ | % | & | ' | ( | ) | \* | + | , | - | . | / |
| **5.** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| **6.** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **7.** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **8.** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **9.** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **A.** | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| **B.** | P | Q | R | S | T | U | V | W | X | Y | Z | [ | \ | ] | ↑ | ← |
| **C.** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **D.** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **E.** |  | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| **F.** | p | q | r | s | t | u | v | w | x | y | z |  | \_ |  | ESC | DEL |

როგორც ვხედავთ, პატარა ასო ყველა დიდ ასოზე მცირეა. ASCII -კოდებში კლებადობა ხდება ჰორიზონრალურად: მარცხნიდან-მარჯვნივ და ვერტიკალურად: ზემოდან ქვევით. მაგალითად Z მეტია a-ზე. ციფრებს შორის შედარება ტრადიციულია, რაც ასახულია ASCII -კოდებია ცხრილშიც.

განვიხილოთ მასივის სორტირების შემდეგი PHP ფუნქციები:

* sort () - მასივების სორტირება ზრდადობით
* rsort () - მასივების სორტირება კლებადობით
* asort () - ასოციური მასივების სორტირება ზრდადობით მნიშვნელობების მიმართ
* ksort () - ასოციური მასივების სორტირება ზრდადობით გასაღების მიმართ
* arsort () - ასოციური მასივების სორტირება კლებადობით მნიშვნელობების მიმართ
* krsort () - ასოციური მასივების სორტირება კლებადობით გასაღების მიმართ

**sort () - მასივების სორტირება ზრდადობით**

სინტაქსისი

sort($array [, $sort\_flags])

ხდება $array მასივის ელემენტების ზრდადობით დალაგება. $sort\_flags - არააუცილებელი პარამეტრით მოიცემა სორტირების პარამეტრი:

* SORT\_REGULAR - ჩვეულებრივი დალაგება, ანუ ადარებს ელემენტებს როგორც ისინი არიან;
* SORT\_NUMERIC - ელემენტების, როგორც რიცხვების შედარება
* SORT\_STRING - ელემენტების, როგორც სიმბოლოების შედარება
* SORT\_LOCALE\_STRING - ელემენტების, როგორც სიმბოლოების შედარება, ლოკალის მიმდინარე მდგომარეობაზე დაფუძნებული

ლისტინგი 8 მასივის დალაგება ზრდადობით

<!DOCTYPE html>

<html>

<body>

<?php

$saqoneli = array("საბურავები", "ზეთი", "აალების სანთლები");

echo '<h3>საწყისი მასივი</h3>';

for($x= 0; $x < count($saqoneli); $x++) {

echo $saqoneli[$x];

echo "<br>";

}

sort($saqoneli);

echo '<h3>დალაგებული მასივი</h3>';

for($x= 0; $x < count($saqoneli); $x++) {

echo $saqoneli[$x];

echo "<br>";

}

?>

</body>

</html>

ლისტინგი 8-ის გაშვების შემდეგ, ვღებულობთ ზრდადობით დალაგებულ მასივს

**საწყისი მასივი**

საბურავები  
ზეთი  
აალების სანთლები

**ზრდადობით დალაგებული მასივი**

აალების სანთლები  
ზეთი  
საბურავები

**rsort () - მასივების სორტირება კლებადობით**

სინტაქსისი

rsort($array [, $sort\_flags])

ხდება $array მასივის ელემენტების კლებადობით დალაგება. $sort\_flags - არააუცილებელი პარამეტრით მოიცემა სორტირების პარამეტრი, ისევე როგორც sort():

ლისტინგი 9 მასივის დალაგება კლებადობით

<!DOCTYPE html>

<html>

<body>

<?php

$saqoneli = array("საბურავები", "ზეთი", "აალების სანთლები");

echo '<h3>საწყისი მასივი</h3>';

for($x= 0; $x < count($saqoneli); $x++) {

echo $saqoneli[$x];

echo "<br>";

}

rsort($saqoneli);

echo '<h3>კლებადობით დალაგებული მასივი</h3>';

for($x= 0; $x < count($saqoneli); $x++) {

echo $saqoneli[$x];

echo "<br>";

}

?>

</body>

</html>

ლისტინგ 9-ს გაშვებისას ვღებულობთ

**საწყისი მასივი**

საბურავები  
ზეთი  
აალების სანთლები

**კლებადობით დალაგებული მასივი**

საბურავები  
ზეთი  
აალების სანთლები

**asort () - ასოციური მასივების სორტირება ზრდადობით მნიშვნელობების მიმართ**

სინტაქსისი asort($arr [, $sort\_flags])

დავალაგოთ ასოცირებული მასივი მნიშვნელობების ზრდადობით შემდეგ კონკრეტულ მაგალითში,

ლისტინგი 10 ასოცირებული მასივის დალაგება მნიშვნელობების ზრდადობით

<!DOCTYPE html>

<html>

<body>

<?php

$saqoneli=array ('საბურავები'=>7, 'ზეთი'=>10, 'აალების სანთლები'=>6);

echo '<h3>საწყისი მასივი</h3>';

foreach($saqoneli as $gasaRebi=>$mniSvneloba) {

echo $gasaRebi." ".$mniSvneloba;

echo "<br>";

}

asort ($saqoneli);

echo '<h3>მნიშვნელობების ზრდადობით დალაგებული ასოცირებული მასივი</h3>';

foreach($saqoneli as $gasaRebi=>$mniSvneloba) {

echo $gasaRebi." ".$mniSvneloba;

echo "<br>";

}

?>

</body>

</html>

ლისტინგი 10-ის გაშვებისას ვღებულობთ

**საწყისი მასივი**

საბურავები 7  
ზეთი 10  
აალების სანთლები 6

**მნიშვნელობების ზრდადობით დალაგებული ასოცირებული მასივი**

აალების სანთლები 6  
საბურავები 7  
ზეთი 10

**arsort () - ასოციური მასივების სორტირება მნიშვნელობების კლებადობით**

სინტაქსისი arsort($arr [, $sort\_flags])

დავალაგოთ ასოცირებული მასივი მნიშვნელობების კლებადობით შემდეგ კონკრეტულ მაგალითში

ლისტინგი 11. ასოცირებული მასივის დალაგება მნიშვნელობების კლებადობით

<!DOCTYPE html>

<html>

<body>

<?php

$saqoneli=array ('საბურავები'=>7, 'ზეთი'=>10, 'აალების სანთლები'=>6);

echo '<h3>საწყისი მასივი</h3>';

foreach($saqoneli as $gasaRebi=>$mniSvneloba) {

echo $gasaRebi." ".$mniSvneloba;

echo "<br>";

}

arsort ($saqoneli);

echo '<h3>მნიშვნელობების კლებადობით დალაგებული ასოცირებული მასივი</h3>';

foreach($saqoneli as $gasaRebi=>$mniSvneloba) {

echo $gasaRebi." ".$mniSvneloba;

echo "<br>";

}

?>

</body>

</html>

ლისტინგი 11-ის გაშვებით ვღებულობთ მნიშვნელობების კლებადობით დალაგებულ ასოცირებული მასივს.

**საწყისი მასივი**

საბურავები 7  
ზეთი 10  
აალების სანთლები 6

**მნიშვნელობების კლებადობით დალაგებული ასოცირებული მასივი**

ზეთი 10  
საბურავები 7  
აალების სანთლები 6

**ksort () - ასოციური მასივების სორტირება ზრდადობით გასაღების მიმართ**

სინტაქსისი ksort($arr [, $sort\_flags])

დავალაგოთ ასოცირებული მასივი გასაღების ზრდადობის მიხედვით შემდეგ კონკრეტულ მაგალითში

ლისტინგი 12 ასოცირებული მასივის დალაგება გასაღების ზრდადობით მიხედვით

<!DOCTYPE html>

<html>

<body>

<?php

$saqoneli=array ('საბურავები'=>8, 'ზეთი'=>10, 'აალების სანთლები'=>6);

echo '<h3>საწყისი მასივი</h3>';

foreach($saqoneli as $gasaRebi=>$mniSvneloba) {

echo $gasaRebi." ".$mniSvneloba;

echo "<br>";

}

ksort ($saqoneli);

echo '<h3>გასაღებების ზრდადობით დალაგებული ასოცირებული მასივი</h3>';

foreach($saqoneli as $gasaRebi=>$mniSvneloba) {

echo $gasaRebi." ".$mniSvneloba;

echo "<br>";

}

?>

</body>

</html>

ლისტინდ 12\_ის გაშვებით ვღებულობთ გასაღების ზრდადობის მიხედვით დალაგებულ ასოცირებულ მასივს

**საწყისი მასივი**

საბურავები 8  
ზეთი 10  
აალების სანთლები 6

**გასაღებების ზრდადობით დალაგებული ასოცირებული მასივი**

აალების სანთლები 6  
ზეთი 10  
საბურავები 8

**krsort () - ასოციური მასივების სორტირება კლებადობით გასაღების მიმართ**

სინტაქსისი krsort($arr [, $sort\_flags])

დავალაგოთ ასოცირებული მასივი გასაღების კლებადობის მიხედვით შემდეგ კონკრეტულ მაგალითში

ლისტინგი 13 ასოცირებული მასივის დალაგება გასაღების კლებადობის მიხედვით

<!DOCTYPE html>

<html>

<body>

<?php

$saqoneli=array ('საბურავები'=>8, 'ზეთი'=>10, 'აალების სანთლები'=>6);

echo '<h3>საწყისი მასივი</h3>';

foreach($saqoneli as $gasaRebi=>$mniSvneloba) {

echo $gasaRebi." ".$mniSvneloba;

echo "<br>";

}

krsort ($saqoneli);

echo '<h3>გასაღების კლებადობით დალაგებული ასოცირებული მასივი</h3>';

foreach($saqoneli as $gasaRebi=>$mniSvneloba) {

echo $gasaRebi." ".$mniSvneloba;

echo "<br>";

}

?>

</body>

</html>

ლისტინგი 13-ის გაშვების შემდეგ მივიღებთ ასოცირებული მასივს დალაგებულს გასაღების კლებადობის მიხედვით

**საწყისი მასივი**

საბურავები 8  
ზეთი 10  
აალების სანთლები 6

**გასაღების კლებადობით დალაგებული ასოცირებული მასივი**

საბურავები 8  
ზეთი 10  
აალების სანთლები 6

**natsort() ფუნქცია განსაზღვრავს ბუნებრივ სორტირებას**

სინტაქსისი natsort($arr)

”ბუნებრივი სორტირება” ნიშნავს იმას, რომ მასივი დალაგდება ისე, როგორც მათ ადამიანი დაალაგებდა, მაგალითად მოცემულია მასივი

*საწყისი*

fili1.txt

fili10.txt

fili2.txt

fili12.txt

fili20.txt

*“მანქანური დალაგება”*

fili1.txt

fili10.txt

fili12.txt

fili2.txt

fili20.txt

*“ადამიანური დალაგება”*

fili1.txt

fili2.txt

fili10.txt

fili12.txt

fili20.txt

დავალაგოთ მასივი ნატურალურად შემდეგ კონკრეტულ მაგალითში

ლისტინგი 14 მასივის ნატურალური დალაგება

<!DOCTYPE html>

<html>

<body>

<?php

$arr\_pirveli = $arr\_meore =

array( "fili12.txt",

"fili10.txt",

"fili2.txt",

"fili1.txt");

sort($arr\_pirveli);

echo "ჩვეულებრივი დალაგება<br>";

echo "<pre>";

print\_r($arr\_pirveli);

echo "</pre>";

natsort($arr\_meore);

echo "<br>ბუნებრივი - ადამიანური დალაგება<br>";

echo "<pre>";

print\_r($arr\_meore);

echo "</pre>";

?>

</body>

</html>

ლისტინგ 14 -ის გაშვების შემდეგ მივიღებთ ნატურალურად დალაგებულ მასივს

**ჩვეულებრივი დალაგება**

Array

(

[0] => fili1.txt

[1] => fili10.txt

[2] => fili12.txt

[3] => fili2.txt

)

**ბუნებრივი - ადამიანური დალაგება**

Array

(

[3] => fili1.txt

[2] => fili2.txt

[1] => fili10.txt

[0] => fili12.txt

)

**ფაილების ჩართვა**

PHP იძლევა საშუალებას სკრიპტში ჩავრთოთ სხვა სკრიპტი ან ფაილი. ამისთვის არსებობს ოთხი საშუალება, რომელიც მოცემულია ცხრილში

ცხრილი 2

|  |  |
| --- | --- |
| კონსტრუქცია | აღწერა |
| include("index.php") | ჩართავს ფაილს index.php, თუ კი ეს ფაილი არ არსებობს, გამოიტანს ამის შესახებ შეტყობინებას და გააგრძელებს მუშაობას |
| include\_once ("index.php") | ჩართავს მხოლოდ ერთხელ ფაილს index.php , იმისდა მიუხედავათ, თუ რამდენჯერაც არ შეგვხდება ჩართვის კონსტრუქცია. თუ კი ეს ფაილი არ არსებობს, გამოიტანს ამის შესახებ შეტყობინებას და გააგრძელებს მუშაობას |
| require("index.php") | ჩართავს ფაილს index.php, თუ კი ეს ფაილი არ არსებობს, გამოიტანს ამის შესახებ შეტყობინებას და შეწყვეტს მუშაობას |
| require\_once ("index.php") | ჩართავს მხოლოდ ერთხელ ფაილს index.php , იმისდა მიუხედავათ, თუ რამდენჯერაც არ შეგვხდება ჩართვის კონსტრუქცია. თუ კი ეს ფაილი არ არსებობს, გამოიტანს ამის შესახებ შეტყობინებას და შეწყვეტს მუშაობას |

ლისტინგი 15 include -ს გამოყენება ჩასართავად

<!DOCTYPE html>

<html>

<body>

<?php

include("file.php");

echo "გამარჯობა, $user <br>";

echo "თქვენი შეკვეთა დამუშავდა $today -ში";

?>

</body>

</html>

ჩართული ფაილის ლისტინგი

<?php

$user = "მომხმარებელო";

$today = date("d.m.y");

?>

ლისტინგ 15 -ის გაშვებისას მივიღებთ

გამარჯობა, მომხმარებელო   
თქვენი შეკვეთა დამუშავდა 26.11.14 -ში