



NUMRAT ARMSTRONG

PLUSPERFECT NUMBERS / NARCISSISTIC NUMBERS

NUMRAT ARMSTRONG

Ne serine e kategorive te veçanta te numrave, pasi jemi njohur me numrat prim, numrat perfekt, serine e Fibonacit, rradhen e kane numrat Armstrong. Siç e theksuam ne rastin e numrave perfekt, duke qene qe cilesia qe numrat plotesojne eshte e forte, sasia e numrave perfekt eshte mjaft e limituar. Bie ne sy qe sa me e forte te jete cilesia apo kushti qe numrat e nje kategorie plotesojne, aq me te paket jane ata, keshtu ndodh me numrat Armstrong.

Me perkufizim nje numer Armstrong (ndryshe quhen edhe plusperfekt ose narcissistic) eshte numri natyror i tille qe, shuma e fuqive te shifrave te tij, me indeks sa numri i shifrave, na jep vete numrin:

$$\text{Keshtu psh. } 153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$$

Ne segmentin numerik [1..500] kemi vetem keta numra armstrong:

1, 153, 370, 371, 407.

Po keshtu numri 8208:

$$8^4 + 2^4 + 0^4 + 8^4 = 4096 + 16 + 0 + 4096 = 8208$$

Me poshte eshte programi ne C++, qe kontrollon nje numer tre shifror natyror nese eshte numer Armstrong ose jo:

// NUMRAT ARMSTRONG

```
#include <iostream>

using namespace std;

int main()
{
    int Num, num, mbetja, sum = 0;
    cout << "Jep nje numer natyror: ";
    cin >> Num;
    num = Num;
    while(num != 0)
    {
        mbetja = num % 10;
        sum += mbetja * mbetja * mbetja;
```

```

        num /= 10;
    }
    if(sum == Num)
        cout << Num << " eshte numer Armstrong.";
    else
        cout << Num << " nuk eshte numer Armstrong.";
    return 0;
}

```

Ne vijim gjeni programin qe kontrollon dhe afishon te gjithë numrat Armstrong ne segmentin num1 – num2:

// NUMRAT ARMSTRONG NE NJE SEGMENT treshifror

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int num1, num2, i, num, digit, sum;
```

```
    cout << "Jepni numrin e pare: ";
```

```
    cin >> num1;
```

```
    cout << "Jepni numrin e dyte: ";
```

```
    cin >> num2;
```

```
    cout << "Numrat Armstrong ndermjet " << num1 << " dhe " << num2 << " jane: " << endl;
```

```
    for(i = num1; i <= num2; i++)
```

```
    {
```

```
        sum = 0;
```

```
        num = i;
```

```
        for(; num > 0; num /= 10)
```

```
{  
    digit = num % 10;  
    sum += digit * digit * digit;  
}  
if(sum == i)  
{  
    cout << i << endl;  
}  
}  
return 0;  
}
```