

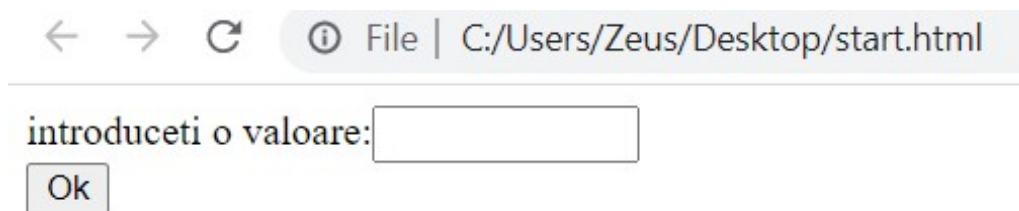
## Introducere

Pentru a ne obișnui mai repede cu limbajul Php, un exercițiu ușor poate fi transcrierea unor algoritmi simplii din C++ în php cu observația că, asigurarea datelor în vederea prelucrării acestora pe un server se poate face prin intermediul formularelor html. Iată în continuare conținutul unui astfel de fișier html (start.html) care va transmite paginii rezolva.php valoarea numerică introdusă în pagina start. În mod analog, putem adăuga fișierului start mai multe “căsuțe” în funcție de cerințele unei probleme sau de ce are nevoie un utilizator să prelucreze.

### Codul paginii start.html

```
<form action="rezolva.php" method="post">
introduceti o valoare:
<input type="number" name="n" min="0" max="99999999"/>
<br/>
<input type="submit" value="Ok"/>
</form>
```

Versiunile noi de html permit așa cum se poate observă mai sus definirea unui tip de dată simplu pe care utilizatorul va fi obligat să o introducă. Din acest motiv munca programatorului este ușurată în sensul că acesta nu mai trebuie să verifice dacă datele introduse sunt corecte din punctul de vedere al tipului de dată așteptat în vederea prelucrării.



The screenshot shows a web browser window with the address bar displaying 'File | C:/Users/Zeus/Desktop/start.html'. Below the address bar, the text 'introduceti o valoare:' is followed by an empty text input field. Below the input field is a button labeled 'Ok'.

În continuare vom efectua câteva prelucrări simple asupra unei singure valori pentru a ne obișnui cu sintaxa și regulile php, analizând în paralel codul unei secvențe c++ care face același lucru.

Se va observa din start că, în php nu sunt obligat să definesc înaintea utilizării tipuri pentru variabilele folosite însă, trebuie avut în vedere în cadrul prelucrării operațiile specific asupra diverselor tipuri de date și funcțiile utilizate în vederea afișării unor rezultate corecte.

## Prelucrarea cifrelor unui număr

Php	C++
Numărări – <b>câte</b> cifre are n?	
<pre>&lt;?php \$n=\$_POST['n'];//preluare din formular \$k=0; if(\$n==0)\$k=1; while(\$n&gt;0) {     \$uc=\$n%10;     \$k++;     \$n=intval(\$n/10);//impartire intreaga } echo \$k;//afisarea ?&gt;</pre>	<pre>int n, k, uc; cin &gt;&gt; n;//citire tastatura k=0; if ( n==0 ) {     k=1; } while ( n!=0 ) {     uc=n%10;     n=n/10;     k=k+1; } cout &lt;&lt; k;//afisarea</pre>

Observăm din start câteva aspecte:

- numele unei variabile în php obligatoriu începe cu semnul \$
- dacă în C++ preiau valoarea unei variabile de la consolă, în php o preiau dintr-un formular, utilizând metoda post sau metoda get, depinde cum a fost specificat acest lucru în formularul html
- nu există diferențe între principalele structuri (decizională, repetitive)
- php nu trateaza valorile numerice ca C++ în senul că, nedeclarând variabilele, împărțirea va avea ca rezultat un numar real, din acest motiv se pot utiliza o serie de funcții numerice pentru a asigura un rezultat corespunzător, așteptat.

Există o serie de tutoriale publicate pe internet care vă pot ajuta în utilizarea corectă a unor funcții.

Iată câteva exemple:

- [https://marplo.net/php-mysql/functii\\_matematice.html](https://marplo.net/php-mysql/functii_matematice.html)
- <https://www.w3schools.com/php/default.asp>

Php	C++
<b>Numărări – câte cifre pare are n?</b>	
<pre>&lt;?php \$n=\$_POST['n']; \$k=0; if(\$n==0)\$k=1; while(\$n&gt;0) {     \$uc=\$n%10;     \$n=intval(\$n/10);     if(\$uc%2==0)\$k++; } echo \$k; ?&gt;</pre>	<pre>int n, k, uc; cin &gt;&gt; n; k=0; if ( n==0 ) {     k=1; } while ( n!=0 ) {     uc=n%10;     n=n/10;     if ( uc%2==0 ) {         k=k+1;     } } cout &lt;&lt; k;</pre>
<b>Numărări – are n cifre pare (da/nu)?</b>	
<pre>&lt;?php \$n=\$_POST['n']; \$k=0; if(\$n==0)\$k=1; while(\$n&gt;0&amp;&amp;\$k==0) {     \$uc=\$n%10;     \$n=intval(\$n/10);     if(\$uc%2==0)\$k++; } if(\$k==1) echo "da"; else echo "nu"; ?&gt;</pre>	<pre>int n, k, uc; cin &gt;&gt; n; k=0; if ( n==0 ) {     k=1; } while ( n!=0 &amp;&amp;\$k==0 ) {     uc=n%10;     n=n/10;     if ( uc%2==0 ) {         k=k+1;     } } if ( k==0 ) {     cout &lt;&lt; "nu"; } else {     cout &lt;&lt; "da"; }</pre>

Sume – <b>suma</b> cifrelor lui n	
<pre>&lt;?php \$n=\$_POST['n']; \$s=0; while(\$n&gt;0) {     \$s=\$s+\$n%10;     \$n=intval(\$n/10); } echo \$s; ?&gt;</pre> <p>Obs. Vom elimina în continuare extragerea ultimei cifre în variabila uc</p>	<pre>int n, s, uc; cin &gt;&gt; n; s=0; while ( n!=0 ) {     uc=n%10;     n=n/10;     s=s+uc; } cout &lt;&lt; s;</pre>
Sume – suma cifrelor <b>impare</b> ale lui n	
<pre>&lt;?php \$n=\$_POST['n']; \$s=0; while(\$n&gt;0) {     if(\$n%2==1)\$s=\$s+\$n%10;     \$n=intval(\$n/10); } echo \$s; ?&gt;</pre>	<pre>int n, s, uc; cin &gt;&gt; n; s=0; while ( n!=0 ) {     uc=n%10;     n=n/10;     if ( uc%2==1 ) {         s=s+uc;     } } cout &lt;&lt; s;</pre>
Produs – <b>produsul</b> cifrelor lui n	
<pre>&lt;?php \$n=\$_POST['n']; \$p=1; if(\$n==0)\$p=0; while(\$n&gt;0) {     \$p=\$p*(\$n%10);     \$n=intval(\$n/10); } echo \$p; ?&gt;</pre>	<pre>int n, p, uc; cin &gt;&gt; n; p=1; if ( n==0 ) p=0; while ( n!=0 ) {     uc=n%10;     n=n/10;     p=p*uc; } cout &lt;&lt; p;</pre>

Determinare minim/maxim - cifra <b>minimă</b>	
<pre> &lt;?php \$n=\$_POST['n']; \$mini=\$n%10; \$n=intval(\$n/10); while(\$n&gt;0) {     if(\$n%10&lt;\$mini)         \$mini=\$n%10;     \$n=intval(\$n/10); } echo \$mini; ?&gt; </pre>	<pre> int n, min, uc; cin &gt;&gt; n; min=n%10; while ( n!=0 ) {     uc=n%10;     n=n/10;     if ( uc&lt;min ) {         min=uc;     } } cout &lt;&lt; min; </pre>
Cifra <b>maximă</b> și de <b>câte</b> ori apare	
<pre> &lt;?php \$n=\$_POST['n']; \$maxi=\$n%10; \$n=intval(\$n/10); \$ap=1; while(\$n&gt;0) {     if(\$n%10&gt;=\$maxi)         if(\$n%10&gt;\$maxi)             {                 \$maxi=\$n%10;                 \$ap=1;             }         else \$ap++;     \$n=intval(\$n/10); } echo \$maxi.' '.\$ap; ?&gt; </pre>	<pre> int n, max, ap, uc; cin &gt;&gt; n; max=n%10; ap=1; n=n/10; while ( n!=0 ) {     uc=n%10;     n=n/10;     if ( uc&gt;=max ) {         if ( uc&gt;max ) {             max=uc;             ap=2;         } else {             ap=ap+1;         }     } } cout &lt;&lt; max &lt;&lt;" " &lt;&lt; ap; </pre>

<b>Testul de palindrom și construcția inversului lui n</b>	
<pre> &lt;?php \$n=\$_POST['n']; \$inv=0; \$aux=\$n; while(\$n&gt;0) {     \$inv=\$inv*10+\$n%10;     \$n=intval(\$n/10); } if(\$inv==\$aux) echo "palindrom"; else echo "nu e palindrom"; ?&gt; </pre>	<pre> int n, aux, inv, uc; cin &gt;&gt; n; aux=n; inv=0; while ( n!=0 ) {     uc=n%10;     n=n/10;     inv=inv*10+uc; } if ( inv==aux ) {     cout &lt;&lt; "palindrom"; } else {     cout &lt;&lt; "nu e palindrom"; } </pre>
<b>Eliminări de cifre din n – eliminarea cifrelor pare, construcții</b>	
<pre> //în doi pași &lt;?php \$n=\$_POST['n']; \$inv=0; \$aux=\$n; while(\$n&gt;0) {     if(\$n%2==1)         \$inv=\$inv*10+\$n%10;     \$n=intval(\$n/10); } while(\$inv&gt;0) {     \$n=\$n*10+\$inv%10;     \$inv=intval(\$inv/10); } echo \$n; ?&gt; </pre>	<pre> //în doi pași int n, inv, uc; cin &gt;&gt; n; inv=0; while ( n!=0 ) {     uc=n%10;     n=n/10;     if ( uc%2==1 ) {         inv=inv*10+uc;     } } while ( inv!=0 ) {     uc=inv%10;     inv=inv/10;     n=n*10+uc; } cout&lt;&lt;n; </pre>

<b>Eliminări de cifre din n – eliminarea cifrelor pare, construcții</b>	
<pre>//direct, un pas &lt;?php \$n=\$_POST['n']; \$nr=0; \$p=1; while(\$n&gt;0) {     if(\$n%2==1)     {         \$nr=\$p*(\$n%10)+\$nr;         \$p=\$p*10;     }     \$n=intval(\$n/10); } echo \$nr; ?&gt;</pre>	<pre>//direct, un pas int n, nr, p, uc; cin &gt;&gt; n; nr=0; p=1; while ( n!=0 ) {     uc=n%10;     n=n/10;     if ( uc%2==1 ) {         nr=p*uc+nr;         p=p*10;     } } cout &lt;&lt; nr;</pre>
<b>Dublări de cifre ale lui n – dublarea cifrelor de 5</b>	
<pre>&lt;?php \$n=\$_POST['n']; \$nr=0; \$p=1; while(\$n&gt;0) {     \$nr=\$p*(\$n%10)+\$nr;     \$p=\$p*10;     if(\$n%10==5)     {         \$nr=\$p*(\$n%10)+\$nr;         \$p=\$p*10;     }     \$n=intval(\$n/10); } echo \$nr; ?&gt;</pre>	<pre>int n, nr, p, uc; cin &gt;&gt; n; nr=0; p=1; while ( n!=0 ) {     uc=n%10;     n=n/10;     nr=p*uc+nr;     p=p*10;     if ( uc%10==5 ) {         nr=p*uc+nr;         p=p*10;     } } cout &lt;&lt; nr;</pre>

## Divizibilitate

Algoritmul general	
<pre>&lt;?php \$n=\$_POST['n']; //initializari for(\$d=1;\$d&lt;=\$n;\$d++)     if(\$n%\$d==0)         //prelucrez \$d //afisari ?&gt;</pre>	<pre>int n, d; cin &gt;&gt; n; //initializari; for ( d=1;d&lt;=n;d=d+1 ) {     if ( n%d==0 ) {         //prelucrez d     } } //afisari</pre>
Numărări – câți divizori are n?	
<pre>&lt;?php \$n=\$_POST['n']; \$k=0; for(\$d=1;\$d&lt;=\$n;\$d++)     if(\$n%\$d==0)         \$k++; echo \$k; ?&gt;</pre>	<pre>int n, k, d; cin &gt;&gt; n; k=0; for ( d=1;d&lt;=n;d=d+1 ) {     if ( n%d==0 ) {         k=k+1;     } } cout &lt;&lt; k;</pre>
Numărări – câți divizori <b>pari</b> are n?	
<pre>&lt;?php \$n=\$_POST['n']; \$k=0; for(\$d=1;\$d&lt;=\$n;\$d=\$d+2)     if(\$n%\$d==0)         \$k++; echo \$k; ?&gt;</pre>	<pre>int n, k, d; cin &gt;&gt; n; k=0; for ( d=1;d&lt;=n;d=d+1 ) {     if ( n%d==0 &amp;&amp; d%2==0 ) {         k=k+1;     } } cout &lt;&lt; k;</pre>



Testul de <b>prim</b> – prin <b>numărare</b>	
<pre>&lt;?php \$n=\$_POST['n']; \$k=0; for(\$d=1;\$d&lt;=\$n;\$d++) if(\$n%\$d==0)     \$k++; if(\$k==2) echo "prim"; else echo "nu e prim"; ?&gt;</pre>	<pre>int n, k, d; cin &gt;&gt; n; k=0; for ( d=1;d&lt;=n;d=d+1 ) {     if ( n%d==0 &amp;&amp; d%2==0 ) {         k=k+1;     } } if ( k==2 ) {     cout &lt;&lt; "prim"; } else {     cout &lt;&lt; "nu e prim"; }</pre>
Suma <b>tuturor</b> divizorilor lui n - simplu	
<pre>&lt;?php \$n=\$_POST['n']; \$s=0; for(\$d=1;\$d&lt;=\$n;\$d++) if(\$n%\$d==0)     \$s=\$s+\$d; echo \$s; ?&gt;</pre>	<pre>int n, s, d; cin &gt;&gt; n; s=0; for ( d=1;d&lt;=n;d=d+1 ) {     if ( n%d==0 ) {         s=s+d;     } } cout &lt;&lt; s;</pre>
Suma <b>tuturor</b> divizorilor lui n - <b>optimizat</b>	
<pre>&lt;?php \$n=\$_POST['n']; \$s=0; for(\$d=1;\$d*\$d&lt;=\$n;\$d++)     if(\$n%\$d==0)     {         \$s=\$s+\$d;         if(\$d!=intval(\$n/\$d))             \$s=\$s+intval(\$n/\$d);     } echo \$s; ?&gt;</pre> <p>Obs. S-a inlocuit aici structura while cu structura for! La fel se poate scrie și în C++!</p>	<pre>int n, s, d; cin &gt;&gt; n; s=0; d=1; while ( d*d&lt;=n ) {     s=s+d;     if ( d!=n/d ) {         s=s+n/d;     }     d++; } cout &lt;&lt; s;</pre>

cmmdc	
<pre>&lt;?php \$a=\$_POST['a']; \$b=\$_POST['b']; while(\$a!=\$b) if(\$a&gt;\$b)     \$a=\$a-\$b; else \$b=\$b-\$a; echo \$a; ?&gt;</pre> <p>Atenție – trebuie să primim aici doi parametri!</p>	<pre>int a, b; cin &gt;&gt; a &gt;&gt; b; while ( a!=b ) {     if ( a&lt;b ) {         a=a-b;     } else {         b=b-a;     } } cout &lt;&lt; a;</pre>
cmmmc(a,b)=a*b/cmmdc(a,b)	
<pre>&lt;?php \$a=\$_POST['a']; \$b=\$_POST['b']; \$aa=\$a;\$bb=\$b; while(\$a!=\$b)     if(\$a&gt;\$b)         \$a=\$a-\$b;     else \$b=\$b-\$a; \$cmmmc=\$aa*\$bb/\$a; echo \$cmmdc; ?&gt;</pre>	<pre>int a, b, aa, bb, r, cmmmc; cin &gt;&gt; a &gt;&gt; b; aa=a; bb=b; while ( b!=0 ) {     r=a % b;     a=b;     b=r; } cmmmc=aa*bb/a;</pre>

## **Bibliografie**

1. [https://marplo.net/php-mysql/functii\\_matematice.html](https://marplo.net/php-mysql/functii_matematice.html)
2. <https://www.w3schools.com/CPP/default.asp>
3. <https://www.w3schools.com/php/default.asp>
4. <https://www.w3schools.com/html/>