

```

> restart;
> #=====
==
> # Author: Dmitry K Demskoi
# Email: ddemskoy@csu.edu.au
# The following procedure produces a LaTeX code which
illustrates the process of dividing A by B using long division.
# Usage: LongDivision(A,B);
# Here A and B are positive integers and A>B. For example the
following command LongDivision(1460912312,273)
# should produce the following output
#=====
==

$$\setlength\tabcolsep{0.5mm}
\begin{tabular}{cccccccccccc}
&&&&&&5 & 3 & 5 & 1 & 3 & 2 & 7\\\cline{4-14}
2 & 7 & 3&$$&1 & 4 & 6 & 0 & 9 & 1 & 2 & 3 & 1 & 2\\
&&&\textbf{--}&1 & 3 & 6 & 5\\\cline{5-8}
&&&&9 & 5 & 9\\
&&&&\textbf{--}&8 & 1 & 9\\\cline{7-9}
&&&&1 & 4 & 0 & 1\\
&&&&\textbf{--}&1 & 3 & 6 & 5\\\cline{7-10}
&&&&&3 & 6 & 2\\
&&&&&\textbf{--}&2 & 7 & 3\\\cline{9-11}
&&&&&&8 & 9 & 3\\
&&&&&&\textbf{--}&8 & 1 & 9\\\cline{10-12}
&&&&&&&7 & 4 & 1\\
&&&&&&&\textbf{--}&5 & 4 & 6\\\cline{11-13}
&&&&&&&&1 & 9 & 5 & 2\\
&&&&&&&&\textbf{--}&1 & 9 & 1 & 1\\\cline{11-14}
&&&&&&&&&4 & 1\\
\end{tabular}}$$

> #=====
==
# This code when compiled in LaTeX will look something like

```

$$\begin{array}{r}
 \phantom{273} \phantom{)} \phantom{1460912312} \phantom{-} \phantom{1365} \phantom{959} \phantom{-} \phantom{819} \phantom{1401} \phantom{-} \phantom{1365} \phantom{362} \phantom{-} \phantom{273} \phantom{893} \phantom{-} \phantom{819} \phantom{741} \phantom{-} \phantom{546} \phantom{1952} \phantom{-} \phantom{1911} \phantom{41} \\
 273 \phantom{)} \phantom{1460912312} \phantom{-} \phantom{1365} \phantom{959} \phantom{-} \phantom{819} \phantom{1401} \phantom{-} \phantom{1365} \phantom{362} \phantom{-} \phantom{273} \phantom{893} \phantom{-} \phantom{819} \phantom{741} \phantom{-} \phantom{546} \phantom{1952} \phantom{-} \phantom{1911} \phantom{41} \\
 \hline
 \phantom{273} \phantom{)} \phantom{1460912312} \phantom{-} 1365 \\
 \hline
 \phantom{273} \phantom{)} \phantom{1460912312} \phantom{-} \phantom{1365} 959 \\
 \phantom{273} \phantom{)} \phantom{1460912312} \phantom{-} \phantom{1365} \phantom{959} - 819 \\
 \hline
 \phantom{273} \phantom{)} \phantom{1460912312} \phantom{-} \phantom{1365} \phantom{959} \phantom{-} 819 1401 \\
 \phantom{273} \phantom{)} \phantom{1460912312} \phantom{-} \phantom{1365} \phantom{959} \phantom{-} \phantom{819} \phantom{1401} - 1365 \\
 \hline
 \phantom{273} \phantom{)} \phantom{1460912312} \phantom{-} \phantom{1365} \phantom{959} \phantom{-} \phantom{819} \phantom{1401} \phantom{-} 1365 362 \\
 \phantom{273} \phantom{)} \phantom{1460912312} \phantom{-} \phantom{1365} \phantom{959} \phantom{-} \phantom{819} \phantom{1401} \phantom{-} \phantom{1365} \phantom{362} - 273 \\
 \hline
 \phantom{273} \phantom{)} \phantom{1460912312} \phantom{-} \phantom{1365} \phantom{959} \phantom{-} \phantom{819} \phantom{1401} \phantom{-} \phantom{1365} \phantom{362} \phantom{-} 273 893 \\
 \phantom{273} \phantom{)} \phantom{1460912312} \phantom{-} \phantom{1365} \phantom{959} \phantom{-} \phantom{819} \phantom{1401} \phantom{-} \phantom{1365} \phantom{362} \phantom{-} \phantom{273} \phantom{893} - 819 \\
 \hline
 \phantom{273} \phantom{)} \phantom{1460912312} \phantom{-} \phantom{1365} \phantom{959} \phantom{-} \phantom{819} \phantom{1401} \phantom{-} \phantom{1365} \phantom{362} \phantom{-} \phantom{273} \phantom{893} \phantom{-} 819 741 \\
 \phantom{273} \phantom{)} \phantom{1460912312} \phantom{-} \phantom{1365} \phantom{959} \phantom{-} \phantom{819} \phantom{1401} \phantom{-} \phantom{1365} \phantom{362} \phantom{-} \phantom{273} \phantom{893} \phantom{-} \phantom{819} \phantom{741} - 546 \\
 \hline
 \phantom{273} \phantom{)} \phantom{1460912312} \phantom{-} \phantom{1365} \phantom{959} \phantom{-} \phantom{819} \phantom{1401} \phantom{-} \phantom{1365} \phantom{362} \phantom{-} \phantom{273} \phantom{893} \phantom{-} \phantom{819} \phantom{741} \phantom{-} 546 1952 \\
 \phantom{273} \phantom{)} \phantom{1460912312} \phantom{-} \phantom{1365} \phantom{959} \phantom{-} \phantom{819} \phantom{1401} \phantom{-} \phantom{1365} \phantom{362} \phantom{-} \phantom{273} \phantom{893} \phantom{-} \phantom{819} \phantom{741} \phantom{-} \phantom{546} \phantom{1952} - 1911 \\
 \hline
 \phantom{273} \phantom{)} \phantom{1460912312} \phantom{-} \phantom{1365} \phantom{959} \phantom{-} \phantom{819} \phantom{1401} \phantom{-} \phantom{1365} \phantom{362} \phantom{-} \phantom{273} \phantom{893} \phantom{-} \phantom{819} \phantom{741} \phantom{-} \phantom{546} \phantom{1952} \phantom{-} 1911 41
 \end{array}$$

>

```

> #=====
==
> LongDivision:=proc(a,b)
  local i,l,la,lb,dvdn,dvdns,dvsr,dvsrs,s,shft,ashft,basespace,
  crct,crct2,flag,Is,substrng,BreakNumber,NumberToArrayString,
  unionlists;
  Is:=proc(a,b,c) if a then RETURN(b) else RETURN(c);fi; end;
  substrng:=proc() parse(substring(convert(args[1],string),args[2])
  );end;

```

```

BreakNumber := proc (a) local i, b; if a=0 then RETURN([0]);fi;b
:= []; for i to length(a) do b := unionlists(b,substrng(a,i .. i)
) end do end;
NumberToArrayString := proc (a) local i, b, s; s := ""; b :=
BreakNumber(a); for i to nops(b)-1 do s := cat(s,b[i]," & ") end
do; RETURN(cat(s,b[nops(b)])) end proc;
unionlists := proc () RETURN([seq(op(args[i]),i = 1 .. nargs)])
end proc;
if a<=0 or b<=0 or a<b then ERROR(`Invalid argument!`);fi;
la:=length(a);
lb:=length(b);
l:=la+lb;
dvdn:=substrng(a,1..1);
#dvsrs:=[];
if dvdn>b then
    flag:=true;dvdns:=[dvdn];dvsrs:=[floor(dvdn/b)];
    else dvdns:=[];dvsrs:=[];flag:=false;
fi;
for i from 2 to la+1 do
    if dvdn<b then
        dvdn:=parse(cat(dvdn,substrng(a,i..i)));
        dvsr:=0;
        else dvsr:=floor(dvdn/b);
            dvdn:=parse(cat(dvdn-dvsr*b,substrng(a,i..i)));
        fi;
        dvdns:=unionlists(dvdns,dvdn);
        dvsrs:=unionlists(dvsrs,dvsr);
    od;
dvsrs:=unionlists(dvsrs,0);
while dvsrs[2]=0 do
    dvsrs:=[0,seq(dvsrs[i],i=3..nops(dvsrs))];
    dvdns:=dvdns[2..nops(dvdns)];
od;
s:=cat("{\\setlength\\tabcolsep{0.5mm}
\\begin{tabular}{",seq("c",i=1..l+2),"}\\n");
s:=cat(s,seq("&",i=1..la+lb-nops(dvsrs)+3),NumberToArrayString
(parse(cat(op(dvsrs[Is(flag,2,1)..nops(dvsrs)-1])))),"\\\\\\cline
{",lb+1,"-",l+1,"}\\n");
s:=cat(s,NumberToArrayString(b),"&\\$)&",NumberToArrayString(a),
"\\\\\\n");
basespace:=lb+1+length(dvdns[1])-length(b*dvsrs[2]);
s:=cat(s,seq("&",i=1..basespace-1),"\\textbf{--}&",

```

```

NumberToArrayString(b*dvsrs[2]),"\\\\\\cline{" ,basespace+1,"-",
basespace+length(dvdns[1]),"}\n");
ashft:=lb+1;
crct:=0;crct2:=0;
for i from 2 to nops(dvdns) do
  if dvdns[i-1]-b*dvsrs[i]=0 then crct:=1;
  else crct:=0;
  fi;
  shft:=length(dvdns[i-1])-length(dvdns[i-1]-b*dvsrs[i]);
  ashft:=ashft+shft;
  s:=cat(s,seq("&",j=1..ashft-crct),Is(crct=0,"","0 &"),Is(i=
nops(dvdns) and dvdns[i]=0,"",NumberToArrayString(dvdns[i])),
"\\\\\\n");
  if i<nops(dvdns) then
    if dvsrs[i+1]=0 then crct2:=1;
    else crct2:=0;
    fi;
    s:=cat(s,seq("&",j=1..ashft-crct2-1+length(dvdns[i])
-length(b*dvsrs[i+1])), "\\textbf{--}&",NumberToArrayString(b*
dvsrs[i+1]),"\\\\\\cline{" ,ashft+1,"-",min(ashft+length(dvdns[i]
),l+1),"}\n");
    fi;
  od;
RETURN(sprintf("$$s$$",cat(s,"\\end{tabular}")));
end;

```

*LongDivision* := **proc**(*a*, *b*)

(1)

```

local i, l, la, lb, dvdn, dvdns, dvsr, dvsrs, s, shft, ashft, basespace, crct, crct2, flag, Is,
substrng, BreakNumber, NumberToArrayString, unionlists;
Is := proc(a, b, c) if a then RETURN(b) else RETURN(c) end if end proc;
substrng := proc( ) parse(substring(convert(args[1], string), args[2])) end proc;
BreakNumber := proc(a)
  local i, b;
  if a = 0 then RETURN([0]) end if;
  b := [ ];
  for i to length(a) do
    b := unionlists(b, substrng(a, i..i))
  end do
end proc;
NumberToArrayString := proc(a)
  local i, b, s;

```

```

s := "";
b := BreakNumber(a);
for i to nops(b) - 1 do s := cat(s, b[i], " & ") end do;
RETURN(cat(s, b[nops(b)]))
end proc;
unionlists := proc( ) RETURN([seq(op(args[i]), i = 1 ..nargs)]) end proc;
if a <= 0 or b <= 0 or a < b then ERROR(Invalid argument!) end if;
la := length(a);
lb := length(b);
l := la + lb;
dvdn := substrng(a, 1 ..1);
if b < dvdn then
    flag := true; dvdns := [dvdn]; dvsrs := [floor(dvdn/b)]
else
    dvdns := [ ]; dvsrs := [ ]; flag := false
end if;
for i from 2 to la + 1 do
    if dvdn < b then
        dvdn := parse(cat(dvdn, substrng(a, i ..i))); dvsr := 0
    else
        dvsr := floor(dvdn/b); dvdn := parse(cat(dvdn - dvsr*b, substrng(a, i ..i)))
    end if;
    dvdns := unionlists(dvdns, dvdn);
    dvsrs := unionlists(dvsrs, dvsr)
end do;
dvsrs := unionlists(dvsrs, 0);
while dvsrs[2] = 0 do
    dvsrs := [0, seq(dvsrs[i], i = 3 ..nops(dvsrs))]; dvdns := dvdns[2 ..nops(dvdns)]
end do;
s := cat("{\setlength\tabcolsep{0.5mm}\n\begin{tabular}{", seq("c", i = 1 ..l + 2), "}\n");
s := cat(s, seq("&", i = 1 ..la + lb - nops(dvsrs) + 3),
NumberToArrayString(parse(cat(op(dvsrs[Is(flag, 2, 1) ..nops(dvsrs) - 1])))),
"\\cline{", lb + 1, "-", l + 1, "}\n");
s := cat(s, NumberToArrayString(b), "&$$&", NumberToArrayString(a), "\\n");
basespace := lb + 1 + length(dvdns[1]) - length(b * dvsrs[2]);

```



