

[7.11] Run TI Basic programs before archiving for better execution speed

Archiving TI Basic programs saves RAM, but you will get better execution speed if you run the program once before archiving it. As an example, I'll use TipDS' program to calculate the digits of pi:

```
PI(digits)
Func
©
©Programmed copyright by Tip DS
©This program is free for use, provided
©no modification is made. There may be no
©charge for this program, unless
©permission is given in righting by
©Tip DS. For more information, right
©to tipds@yahoo.com
©
Local  expan1,expan2,frac,answer,tmp1
x+Σ((-1)^t*x^(2*t+1)/(2*t+1),t,1,iPart(digits/(1.4))-1)→expan1
expan1|x=y→expan2
4*(4*expan1-expan2)|x=1/5 and y=1/239→frac
frac-3→frac
"3."→answer
iPart(10^digits*frac)→tmp1
answer&string(tmp1)→answer
EndFunc
```

Note that this is an *old* version of this program, and is only for demonstration purposes!

If you download this program with GraphLink, and immediately archive it, then it has a size of 520 bytes and an execution time of about 1.32 seconds per call. If you run the program once before archiving it, it has a size of 602 bytes, and an execution time of about 1.23 seconds per call, for an improvement of about 7%.

Bigger, more complex programs will show more improvement in execution speed. The reason is that if you archive the program before you run it, the operating system has to recompile the program each time it is run. If you run it once before archiving it, then the OS saves the compiled version.

On the other hand, if code size is more important to you than execution speed, you might want to archive the programs *before* running them. Note that the *PI()* program size increases by 82 bytes, or about 16%, if the program is run once before archiving.

If you are distributing a software package with many programs and functions, you might consider writing a routine that would automatically execute and archive all the programs for the user.

(Credit to Lars Frederiksen)