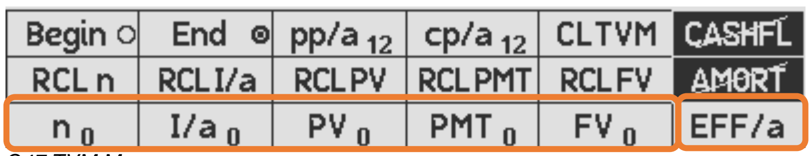
Application note 012 TVM,

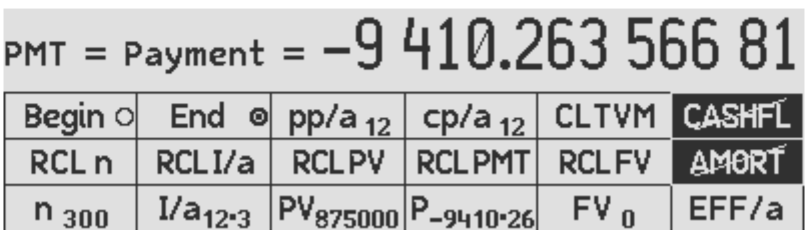
The intention of this AppNote is to demonstrate the way to use the TVM menu, using an available web tutorial.

Note the C47 TVM key terminology is slightly more detailed but essentially the same as on the HP10bII and other financial HP calculators:

Photo credit of HP10BII+, hp.com



C47 TVM Menu



C47 TVM Menu with typical values populated

5 Standard TVM keys:

n ⇒ This key refers to the number of (payment) periods
I/a ⇒ This key refers to the interest rate (10% would be 10 not 0.10). a.k.a. discount rate or rate of return.
PV ⇒ This key refers to the Present Value
PMT ⇒ This key refers to the Annuity Payment
FV ⇒ This key refers to the Future Value

Special options:

pp/a Payment periods per year, generally 1 or 12.
cp/a Compounding periods per year, generally 1 or 12.
Begin ○ End ⊙ Interest on the beginning of the period, or end of the period.

Section 1:

From the great tutorial by Dr. Kevin Bracker; Dr. Fang Lin; and jpursley at Pressbooks or Wayback Machine:

All references below coloured in yellow are directly taken from linked document, under the appropriate Creative Commons license NC4.0 by in this referenced document.

Example One:

6 [n] 7 [I/a] 10 000 [PV] 0 [PMT] Here, on C47 you have to specify the details of compounding. Do 1 [pp/a] and [cp/a] will be defaulted to the same. Ensure End is set: [Begin ○ End ◉]. [FV] is -15 007.30	You are investing $10,000 today and want to know how much you will have after 6 years if you earn a 7% rate of return over the 6-year time frame. Since you are starting with $10,000, that is your present value. You have 6 years, so the number of time periods is 6. The 7% rate of return means you have a 7% interest rate. In this example we are not using an annuity, so we are going to set the Annuity Payment to zero. 6 N 7 I/Y 10,000 PV 0 PMT FV -15,007.30

Example Two:

5 [n] 9 [I/a] 0 [PMT] 6 000 [FV] Here, on C47 you have to specify the details of compounding. Do 1 [pp/a] and [cp/a] will be defaulted to the same. Ensure End is set: [Begin ○ End ◉]. [PV] is -3 899.59	You are going to receive $6000 in 5 years. Assuming a 9% discount rate, what is this worth to you today? 5 N 9 I/Y 0 PMT 6000 FV PV -3899.59

Note: light grey represents settings the same as the previous problem, i.e. no need to re-enter those.

Example Three:

25 [n] 6 [I/a] 400 000 [PMT] 0 [FV] Here, on C47 you have to specify the details of compounding. Do 1 [pp/a] and [cp/a] will be defaulted to the same. Ensure End is set: [Begin ○ End ⊙]. [PV] is -5 113 342.46	Assume you have just won a $10 Million Lottery Jackpot. However, instead of paying you the $10 Million up front, you have the choice of receiving $5 Million today or $400,000 per year at the end of each year for the next 25 years. Assuming a 6% discount rate, which would you prefer? In order to answer this, you need to find the PV of the $400,000 per year for 25 years. This is done as follows: 25 N 6 I/Y 400,000 PMT 0 FV PV 5,113,342.46 (Note that we dropped the negative sign) Our understanding is that the lure is 0.4M x 25 = 10M, vs. the single payment of half of that, but right now. To get back to the original question, receiving the $10M in total over 25 years is clearly better than receiving $5M now!

Note: light grey represents settings the same as the previous problem, i.e. no need to re-enter those.

Example Three A:

All the same as Ex 3, set to BEGIN: [Begin ⊙ End ○] [PV] is -5 420 143.01	To be more realistic, as the first 400k will be paid immediately, set to Begin, and see the 'deal' being worth even more making it the right decision to take the deal. PV 5,420,143.01 (Note that we dropped the negative sign)

Example Three B & C: calculating using future end values

C47 Solution B: 25 [n] 6 [I/a] 5 000 000 [PV] 0 [PMT] 1 [pp/a] [Begin] [FV] is -21 459 353.60	Although the original solution THREE A (in yellow above) is the most elegant, a more "hand-waiving" intuitive way (as opposed to the quickest way) to illustrate the compounding comparison better, is equally easily done on the C47: We simply compare the tangible future values of both deals instead of the 'financial' way of evaluating the current worth of a future investment: Clearly the same answer, that $23.3M is more convincing than $21.5M:	
C47 Solution C: 25 [n] 6 [I/a] 0 [PV] 400 000 [PMT] 1 [pp/a] [Begin] [FV] is -23 262 553.09	B: We first check what $5M would do for us in 25 years: 25 N 6 I/Y 5 000 000 PV 0 PMT FV $21,459,353.60 (dropped the negative sign)	C: Then what the "deal" would do for us in the same 25 years: 25 N 6 I/Y 0 PV 400 000 PMT FV $23,262,553.09 (dropped the negative sign)

Example Four:

20 **n** 10 **I/a** 0 **PV** 1 000 000 **FV** Here, on C47 you have to specify the details of compounding. Do 1 **pp/a** and **cp/a** will be defaulted to the same. Ensure End is set: **Begin ○ End ◉**. **PMT** is -17 459.62	You want to become a millionaire and plan to do so through a savings/investment plan. Assuming you want to reach your goal in 20 years and anticipate earning a 10% rate of return, how much must you save at the end of each year in order to reach your goal? 20 N 10 I/Y 0 PV 1,000,000 FV PMT $17,459.62 This means you will need to save $17,459.62 per year in order to achieve your goal.

Note: light grey represents settings the same as the previous problem, i.e. no need to re-enter those.

Example Five:

C47 has the setting Payment Periods Per Year, so type 26 **pp/a** and it is assumed the Compounding Periods Per Year **cp/a** is the same. 30 ENTER 26 X **n** 10 **I/a** (still the same as Ex 4) 0 **PV** (still the same as Ex 4) 1 000 000 **FV** (still the same as Ex 4) Ensure End is set: **Begin ○ End ◉** (still the same as Ex 4). **PMT** is -202.75	Now assume that you want to accumulate $1 million in 30 years, but instead of saving each year, you are going to save every two weeks (we will earn a 10% annual rate of return). There are 26 2-week periods in each year, so now you have to adjust your calculator to work with 26 periods per year. You can do this as follows: 26 SHIFT P/YR 780 N 10 I/Y 0 PV 1,000,000 FV PMT $202.75

Note: light grey represents settings the same as the previous problem, i.e. no need to re-enter those.

Example Six:

C47 has the setting Payment Periods Per Year, so type 1 **pp/a** and it is assumed the Compounding Periods Per Year **cp/a** is the same. (still the same as Ex 4, 5) 35 **n** -3 000 **PMT** 0 **PV** (still the same form Ex 4, 5) 1 000 000 **FV** (still the same form Ex 4, 5) Ensure End is set: **Begin ○ End ◉** (still the same as Ex 4, 5). **I/a** is 10.89 %	Let's keep working with the goal of becoming a millionaire. However, instead of calculating how much you must save, we'll assume you can save $3000 per year and want to find the rate of return you will need to earn to reach your goal. This time we will give ourselves 35 years of saving $3000 per year. 35 N 0 PV -3000 PMT 1,000,000 FV I/Y 10.89%

Note: light grey represents settings the same as the previous problem, i.e. no need to re-enter those.

Example Seven:

C47 has the setting Payment Periods Per Year, so type 1 **pp/a** and it is assumed the Compounding Periods Per Year **cp/a** is the same. (still the same as Ex 4, 5, 6) 35 **n** (still the same as Ex 6) -3 000 **PMT** (still the same as Ex 6) -40 000 **PV** 1 000 000 **FV** (still the same as Ex 4, 5, 6) Ensure End is set: **Begin ○ End ◉** (still the same as Ex 4, 5, 6). **I/a** is 7.63 %	Here is one last variation on our millionaire example. This time, instead of starting with nothing, let's assume that we already have $40,000 and plan to save an additional $3000 per year over the next 35 years. Now, what rate of return must we earn in order to accumulate $1,000,000 at the end of the 35th year? 35 N -40,000 PV -3000 PMT 1,000,000 FV I/Y 7.63%

Note: light grey represents settings the same as the previous problem, i.e. no need to re-enter those.

Examples Eight through Ten are “Cash Flow Problems” cannot yet be done by C47:

Example Eleven:

C47 does have EFF% from the latest version 00.109.02.00.

7.8 [I/a] to enter NOM% or Nominal interest value 4 [cp/a] for 4 compounding periods per year. [EFF/a] is 8.0311 % (Or press 4 [pp/a] for 4 Payment Periods Per Year, which does not matter, but [cp/a] will always follow.)	4 SHIFT P/YR (this sets your periods per year to 4 for quarterly compounding. Be careful here as this means all your time value of money calculations will use 4 periods per year until you change your P/YR again...just like if you changed the P/YR for a 5-key problem.) 7.8 SHIFT NOM% (this enters the 7.8% nominal rate) SHIFT EFF% (this solves for the effective annual rate to generate your final answer of 8.03%)
7.6 [I/a] to enter NOM% or Nominal interest value 365 [cp/a] for 365 Periods Per Year. [EFF/a] is 7.8954%	For the 7.6% compounded daily it is: 365 SHIFT P/YR 7.6 SHIFT NOM% SHIFT EFF% This will give you your answer of 7.90%

Section 2: From Swissmicros Forum

From the post of DM319 https://forum.swissmicros.com/viewtopic.php?f=2&t=3987, relating to comparative tests for TVM on multiple calculators. Here, the problem solutions are handled for C47 interest only as part of the application note for illustrating the compilation of RPN TVN programs on C47. Do follow the linked reference for updated results on all tested calculators:

#	Ref	N	I%YR	PV	PMT	FV	P/YR	End
1	DM	38 x 12	5.25%	270 000	?????	0	12	yes
1b	DM	38 x 12	?????	270 000	-14 584/12	0	12	yes

#	Ref	N	I%YR	PV	PMT	FV	P/YR	End
2	SlideRule	360	15% → 12%	100 000	?????	0	12	yes

https://www.hpmuseum.org/forum/thread-20707.html. Calculate PMT, given n = 360, I%YR = 15%, PV = 100000, FV = 0, then calculate PMT again with I%YR = 12%. Subtract the two results, and put that back into PMT, then change n = 36 and I%YR = 15% again, and calculate PV.

#	Ref	N	I%YR	PV	PMT	FV	P/YR	End
3	Kahan 1983	60x60x24x365	10%	0	-0.01	?????	=N	yes

https://www.hpmuseum.org/forum/thread-1012.html

#	Ref	N	I%YR	PV	PMT	FV	P/YR	End
4	DM	480	~0 → ?????	100 000	?→ PMT	0	12	yes

Calculate PMT first given I%YR = 0, then re-input this back into PMT and calculate I%YR. On the HP-12c this is best done by pressing x<>y twice before putting back into PMT. On C47 this is not required as the last solved value remains in the I/a register.

#	Ref	N	I%YR	PV	PMT	FV	P/YR	End
5	Dieter	10	?????	50	-30	400	1	yes
6	Dieter	10	?????	50	-30	80	1	yes

https://www.hpmuseum.org/cgi-sys/cgiwrap/hpmuseum/archv021.cgi?read=234439. Note that both these problems have more than one solution.

#	Ref	N	I%YR	PV	PMT	FV	P/YR	End
7	A Chan	10	?????	-100	10	1e-10	12	yes

https://www.hpmuseum.org/forum/thread-18359-post-161549.html#pid161549

#	Ref	N	I%YR	PV	PMT	FV	P/YR	End
8	Miguel	32	?????	-999 999	0	1e6	1	yes

https://www.hpmuseum.org/cgi-sys/cgiwrap/hpmuseum/archv017.cgi?read=120592

#	Ref	N	I%YR	PV	PMT	FV	P/YR	End
9	DM	?????	25	100 000	-2 083.333334	0	12	yes
10	DM	?????	25	100 000	-2 040.816327	0	12	no

Table copied from quoted link, modified typographically, 2024-05-30.
URL Links in table above, as per original table, on quoted link.
Added text in brown.

These problems were solved using RPN programs, illustrating the use thereof on C47: All programs can be downloaded from https://47calc.com, directly linked here:

Dm319 Problem 1:

```
LBL 'P1'
  38
  12
  *
  STO 'NPPER'
  5.25
  STO 'I%/a'
  270 000
  STO 'PV'
  0
  STO 'PMT'
  0
  STO 'FV'
  12
  STO 'PPER/a'
  STO 'CPER/a'
  ENDP
  PMT
END
```

Result in 0.6 sec, USB powered C47 on DM42:

PMT = −1368.14835535057757212882816319868

Dm319 Problem 1b:

```
LBL 'P1b'
  38
  12
  *
  STO 'NPPER'
  0
  STO 'I%/a'
  270 000
  STO 'PV'
  −14 584
  12
  /
  STO 'PMT'
  0
  STO 'FV'
  12
  STO 'PPER/a'
  STO 'CPER/a'
  ENDP
  I/a
END
```

Result in 1.1 sec, USB powered C47 on DM42:

I/a = 4.373218372310086835354687826497751 %

Dm319 Problem 2:

```
LBL 'P2'
  360
  STO 'NPPER'
  15
  STO 'I%/a'
  100 000
  STO 'PV'
  0
  STO 'FV'
  12
  STO 'PPER/a'
  STO 'CPER/a'
  ENDP
  0
  STO 'PMT'
  PMT
  12
  STO 'I%/a'
  DROPX
  PMT
  −
  STO 'PMT'
  36
  STO 'NPPER'
  15
  STO 'I%/a'
  PV
END
```

Result in 2.2 sec, USB powered C47 on DM42:

PV = 6803.0921619855971136255557644647 24

Dm319 Problem 3:

```
LBL 'P3'
  60
  X^2
  24
  *
  365
  *
  STO 'NPPER'
  STO 'PPER/a'
  STO 'CPER/a'
  10
  STO 'I%/a'
  0
  STO 'PV'
  −0.01
  STO 'PMT'
  ENDP
  FV
END
```

Result in 3.2 sec, USB powered C47 on DM42:

FV = 331667.0066907768917803419084360507

Dm319 Problem 4:

```
  LBL 'P4'
    480
    STO 'NPPER'
    0
    STO 'I%/a'
    100 000
    STO 'PV'
    0
    STO 'PMT'
    STO 'FV'
    12
    STO 'PPER/a'
    STO 'CPER/a'
    ENDP
    PMT
    I/a
  END
```

Result in 4.3 sec, USB powered C47 on DM42:
I/a = −3.75935122288958926199999999999999E−19 %

Dm319 Problem 5: First solution

Default result, without biasing the solver.

```
LBL 'P5'
  10
  STO 'NPPER'
  0
  STO 'I%/a'
  50
  STO 'PV'
  -30
  STO 'PMT'
  400
  STO 'FV'
  1
  STO 'PPER/a'
  STO 'CPER/a'
  ENDP
  I/a
END

Result in 4.8 sec, USB powered C47 on DM42:
I/a = 14.43587132807995697420470129887705 %
```

Dm319 Problem 5: Second solution

The key to the second solution is to point the solver to closer to the expected second result, i.e. store 50 in I%/a to make it search starting at 50. See brown:

```
LBL 'P5b'
  10
  STO 'NPPER'
  0
  STO 'I%/a'
  50
  STO 'PV'
  -30
  STO 'PMT'
  400
  STO 'FV'
  1
  STO 'PPER/a'
  STO 'CPER/a'
  ENDP
  50
  STO 'I%/a'
  DROPX
  I/a
END

Result in 5.3 sec, USB powered C47 on DM42:
I/a = 53.17221326838472431046024137437116 %
```

Dm319 Problem 6: First solution

Default result, without biasing the solver.

```
LBL 'P6'
  10
  STO 'NPPER'
  0
  STO 'I%/a'
  50
  STO 'PV'
  -30
  STO 'PMT'
  80
  STO 'FV'
  1
  STO 'PPER/a'
  STO 'CPER/a'
  ENDP
  I/a
END

Result in 5.8 sec, USB powered C47 on DM42:
I/a = -36.89336987417774420723991401135426 %
```

Dm319 Problem 6: Second solution

The key to the second solution is to point the solver to closer to the expected second result, i.e. store 50 in I%/a to make it search starting at 50. See brown:

```
LBL 'P6b'
  10
  STO 'NPPER'
  0
  STO 'I%/a'
  50
  STO 'PV'
  -30
  STO 'PMT'
  80
  STO 'FV'
  1
  STO 'PPER/a'
  STO 'CPER/a'
  ENDP
  50
  STO 'I%/a'
  DROPX
  I/a
END

Result in 6.5 sec, USB powered C47 on DM42:
I/a = 58.46195526622098709870320744645958 %
```

Dm319 Problem 7:

```
LBL 'P7'
  10
  STO 'NPPER'
  0
  STO 'I%/a'
  -100
  STO 'PV'
  10
  STO 'PMT'
  1.*10^-10
  STO 'FV'
  12
  STO 'PPER/a'
  STO 'CPER/a'
  ENDP
  I/a
END
```

Result in 7.2 sec, USB powered C47 on DM42:
I/a = 2.18181818181580062818816748324244E-10 %

Dm319 Problem 8:

```
LBL 'P8'
  32
  STO 'NPPER'
  0
  STO 'I%/a'
  -999 999
  STO 'PV'
  0
  STO 'PMT'
  1.*10^6
  STO 'FV'
  1
  STO 'PPER/a'
  STO 'CPER/a'
  ENDP
  I/a
END
```

Result in 7.8 sec, USB powered C47 on DM42:
I/a = 0.000003125001611329216004244745444268481 %

Dm319 Problem 9:

```
LBL 'P9'
  0
  STO 'NPPER'
  25
  STO 'I%/a'
  100 000
  STO 'PV'
  -2 083.333 334
  STO 'PMT'
  0
  STO 'FV'
  12
  STO 'PPER/a'
  STO 'CPER/a'
  ENDP
  n
END
```

Result in 11.9 sec, USB powered C47 on DM42:
n = 1060.303390000510007127328529884813

Dm319 Problem 10:

```
LBL 'P10'
  0
  STO 'NPPER'
  25
  STO 'I%/a'
  100 000
  STO 'PV'
  -2 040.816 327
  STO 'PMT'
  0
  STO 'FV'
  12
  STO 'PPER/a'
  STO 'CPER/a'
  BeginP
  n
END
```

Result in 14.9 sec, USB powered C47 on DM42:
n = 1076.31954436770664643510613104444

Sequencing program 'Pall':

Run all problem files and reporting results and time durations of processing:

```
0001:  LBL 'Pall'

0002:  XEQ 'P1'
0003:    'TVM P1:'
0004:  XEQ 'TT'

0005:  XEQ 'P1b'
0006:    'TVM P1b:'
0007:  XEQ 'TT'

0008:  XEQ 'P2'
0009:    'TVM P2:'
0010:  XEQ 'TT'

0011:  XEQ 'P3'
0012:    'TVM P3:'
0013:  XEQ 'TT'

0014:  XEQ 'P4'
0015:    'TVM P4:'
0016:  XEQ 'TT'

0017:  XEQ 'P5'
0018:    'TVM P5:'
0019:  XEQ 'TT'

0020:  XEQ 'P5b'
0021:    'TVM P5B(50):'
0022:  XEQ 'TT'

0023:  XEQ 'P6'
0024:    'TVM P6:'
0025:  XEQ 'TT'

0026:  XEQ 'P6b'
0027:    'TVM P6B(50):'
0028:  XEQ 'TT'

0029:  XEQ 'P7'
0030:    'TVM P7:'
0031:  XEQ 'TT'

0032:  XEQ 'P8'
0033:    'TVM P8:'
0034:  XEQ 'TT'

0035:  XEQ 'P9'
0036:    'TVM P9:'
0037:  XEQ 'TT'

0038:  XEQ 'P10'
0039:    'TVM P10:'
0040:  XEQ 'TT'

0041:  END
```

Subroutine 'TT':

Print results to text files:

```
0001:  LBL 'TT'

0002:    LASTT           //store last op-time
0003:    STO 00
0004:    DROPX

0005:    Printxy         //print results to file
0006:    RCL 00
0007:    10
0008:    /
0009:    ' Sec'
0010:    +
0011:    Printx          //print op-time to file

0012:  END
```

Author Jaco Mostert; Proofread by Duncan Murray