

COMPLEMENT



AND ADD

DECEMBER 1979-JANUARY 1980

INTERNATIONAL NO.6

Bonjour a la SC/MP useres everywherement, cette newsletter est from maintenant on est etre ecrire en Francais.

NO? Oh all right! Hard to believe I once passed a French O level. The reason for this foolishness is that Complement and Add has been read in France, Belgium, Denmark (I hope) and South Africa.

Not by many people, I will grant you, but still it has travelled further than me.

Lots and lots has been happenning recently, so here is a quick round up, (as opposed to truncate), firstly on meetings.

In the Canterbury area, John Leach organised a meeting for those in his area. I could not attend, but was informed that only a handfull of SC/MP fanatics turned up.

In London, as mentioned last issue, though unfortunately too late for some, Martin Stamp hosted an informal meeting.

Five of us were there, Me, Paul Kufman, Bob Bowdler, Martin Stamp (surprise!) and Nick Toop from S of C fame. (Now at Acorn)

Bob had his system up and running with a PE VDU. Hopefully I can retrieve some of Bob's software for this mag, though some of it runs on a special monitor for the VDU running in his 16K ram. For those without such luxuries, he had a ripoff of 'Simon' called Charlie, which (for those who haven't played with the WH Smith selection of electronic games) gives you an ever increasing stream of numbers which you must repeat. (pardon)

The program blows a raspberry if you get the numbers wrong!

Nick Toop was very helpfull over explaining certain aspects of the MK14 and VDU. One of the things that came out of the meeting was that the 8070 series may never hit the popular market, nat Semi aren't aiming for the same market as the 8060.

For those who by some fluke of fate, or the P.C., have not seen issue 5 yet, Clive Isbell is still after you lot in East Anglia. See the revised list of members for his address, it would be nice to start up a sub-group over there. Anyone wanting to do the same is of course welcome, let me know in advance when you can hold meetings, in time for the newsletters.

Forget what I said about Breadboard tickets, I never did get round to it - its too late now anyway! If you go there, I will be wearing a homemade MK14 badge! I'll probably go on the saturday around 11.00. Don't think S. of C. will be there, but Acorn will be.

Sorry about the delays in writing, I'm often waiting for the return of something you may have asked for, such as the Basic Assembler listing. But please do keep on writing

Staples will no longer appear, as they are only a nuisance to those who (for some strange reason) want to photocopy bits.

Also, apologies to anyone recieving more than 1 newsletter, and to those who have not recieved a particular one.

An example of delays that can occur are for instance R. Gilchrist moving off to the States without letting me know, and someone moving without

giving me a two month notice. Hope that doesn't sound too official!

Bits and pieces available

I'm selling these things for stamps, cos it saves me buying them, and also saves the trouble of paying in silly amounts to the bank.
I only charge the cost price anyway which is just for the photocopying (6p each).

SC/MP Assembler in basic (v. odd dialect)	20p of stamps.
Each issue of C&A (4 onwards)	70p of stamps.
Issue 1	35p "
Issue 2	45p "
Issue 3	50p "
Maze program	45p "
How the above works!!	40p "
Mastermind prog (machine guesses)	40p "

To borrow any item for a month, 10p of stamps per item.

Oh yes, a couple more,

Addressing modes a 4 page wonder for 35p of stamps.

same but different - 7 pages 55p "

SC/MP label assembler, as in no.5 40p "

Life program for PE VDU, for S OF C soon 30p "

As it is nearing Christmas, here are some Christmas tunes from Brian Johnson intended for use with the music box program.

So just put in the bytes listed instead of the standard 'God save our Queen'

LAND OF HOPE AND GLORY

MARK THE HERALD ANGELS SING								THE FIRST NOEL								ONCE IN ROYAL DAVIDS CITY							
66	6B	8B	4A	6B	6F	48	46	84	46	48	49	87	87	46	47	89	E4						
6F	6D	72	72	92	70	AB	4D	4F	90	8F	8D	E2	AC	AC	4B	4C	8E						
6F	6D	CF	66	6B	8B	AB	4D	4F	70	6F	6D	C9	8B	4B	4D	4E	90						
4A	6B	6F	6F	6D	72	6B	6D	6F	70	6B	69	B5	8E	93	73	52	50						
6D	8D	6B	6A	68	66	A8	48	46	84	46	48	AE	00	00	00	00	00						
20	72	72	72	6	70	49	AB	4D	4F	70	6F	64	68	89	49	49	48						
6F	6F	6D	72	72	72	6D	AB	4D	4F	70	6F	49	4B	6B	69	69	6D						
6B	70	6F	6F	6D	74	6D	6B	6D	6F	70	6B	90	4D	4D	4B	49	48						
74	74	72	70	6F	DO	69	A8	48	46	84	46	A9	64	68	89	49	49						
6D	2F	30	92	4B	6B	48	49	AB	50	4F	AD	48	49	4B	6B	69	69						
6D	CF	94	74	74	72	6D	CB	70	6F	6D	6B	6D	90	4D	4D	4B	49						
70	6F	DO	6D	2F	30	6D	6F	70	6B	69	A8	48	A9	72	72	90	49						
92	4B	6B	6D	EB	00	00						6E	6E	AD	72	72	90						
												4D	4D	4B	49	48	A9						

WHILE SHEPHERDS WATCHED

6B	AF	6F	6D	6B	70
70	6F	6D	6F	72	72
71	D2	6F	B4	52	70
6F	6D	6B	6A	6F	6D
6B	6B	6A	EB	00	00

Bad joke no. 1, via Mike Grundy. He suggested I have a stall at Breadboard called an 'Exchange Program Counter'.

My joke is : What is green& silver and lies in two pieces in the bin?
Answer: An MK14 with all the ICs in back to front! Oh sorry.

I've put a page of hasty notes on adding another 1K of ram, I'm not really a hardware fanatic, so if I can expand my system anyone can! The Editor of Recreational Computing has asked if anyone in the group is using a MK14 for education in anyway - so please write if so. By the way, I have now built a VDU like Paul, it seems to work very well. From a hardware point of view its limitations seem to be just to be an incorrect vertical position - the top is just about visible on my trustee portable, whilst the lowest position is plenty inches off the bottom. Last week it was used at a local Christmas Fete, as a Fruit machine. It was so popular with the kids that I was asked to do another 'gig' (joke) at another fete the following week. The original program worked in 1K, but now it does clever things like displaying 'Machine fruit' ... or 'Frut Machine' and doing a correction! In graphics mode, the only achievement is an image of the USS Enterprise! On a serious note, the VDU can make a good device for showing program output - no multiplexing!

Some Stop press news! I received a letter from Lintronics with a catalogue of their goods. Obviously I can't put it all here - I'm sure they will send a catalogue for a SAE. They are selling ready made MK14's at £58, and a keyboard to fit at £10.00. a quid off the normal price.

The other point of interest is a £16 power supply (5V @ 2A) which doesn't sound too bad. They can also supply other power ranges.
Address: Lintronics, 37A Chiltern Av, Bushy, Herts.

Address: Elintronics, 57A, Chiltern AV, Bushey, Herts.

I will now give some details about writing games programs that learn! There is no point in me listing the program here as it was intended for a 1½K ram area! But hopefully the ideas will help you get started on something worthwhile!

Writing Programs For Board Games.

I have unfortunately lost the tape dump of the complete program but I can recall roughly how the program behaved.

You must first define the rules of the game, you will always have a board (n by n grid) which should be numbered so that moving from one square to another is a constant addition or subtraction.

Also there will be two sets of pieces, one for each player, so the number of these must be known.

W The first thing to think of is whether to store game information as
1) A picture of the whole board, i.e. each location indicates what piece
is at the board position

is at the board position.
2) A picture of each piece and its position number on the board - this is the quickest way of dealing with most games.

Elaborating on this, we can say:

To move a piece, just delete the entry and add a new one for the revised position.

If a piece is removed, then put '20' into the current entry.

As an example, the game I designed was called Quadpawn, based on Byte Magazine's Hexpawn. The board is a 4 by 4 affair with 4 black pawns at one end, 4 white pawns at the other. The pawns move like in chess, i.e. forward 1 square, but diagonally forward when taking an opponent.

The object is to either get to the other end (any one piece) or to stop the opponent from being able to move.

So by numbering the board as shown, you only need 8 bytes to hold the status of the game, as below.

black	white
01020304	13141516

and more such as 04 → CA
would be 0405060708090A

12	13	14	15	16	17
					united states
	D	E	F	I O	
6	7	3	9	A	
	2	3	4	2	
	2	3	4	2	5A basic direction

The best way of validating a possible move, and the way I programmed Quadpawn was to use a subroutine which took two input parameters and produced a code which gave info as to whether the move would go to an empty square, one occupied by you, or a position occupied by the opponent.

The major point about Quadpawn is that the game learnt by its mistakes! Whenever it was your turn, you could reply with 'GO' to indicate that the machine had lost. The machine would then store the status of both players into a table of 'bad moves'. The table would then be used on every move by the machine, so that it would not repeat a mistake.

If there were no possible moves available to the machine, it would resign, storing the previous position onto its table! All these things are easy to program, if you take a bit at a time. Most of the storage was used up in storing bad moves - about 700 bytes in the end. Eventually the machine could not seem to lose, I must resurrect the program sometime, just to see if I can beat it somehow!

Some Stop Press Stuff

Letter received from DJD of Practical Electronics asking for any interesting material, so have sent him some stuff! Computer Centre are offering the MK14 at £38 if anyone wants another one. Write to: Computercentre, 34-36, St. Helens Rd, Westcliff on Sea, Essex. Leaflet enclosed in one circulation.

The next issue is by the way, a birthday, the first newsletter being sent out in February of this year! Page 5 of the newsletter is a rough layout of some idea for an extra 1K of memory. I am not the hardware sort, so don't expect too much!

Time for another program, from Graham Turrell in Suffolk. The program calculates, then displays, the day of the week of any year between 1752 and 2099

The data should be entered as follows:

First two digits of year:	0F8E
Last two digits of year:	0F8F
Month 1 to 12	0F90
Day of month	0F91

All these must be calculated in hexad decimal.

E.g. 25th December 79 is 13,4F,0C,19.

The program was compiled on the SC/MP assembler he is developing.

0F12	C40F	1	LDI	15
0F14	35	2	KPAH	1
0F15	C497	3	LDI	-105
0F17	31	4	KPAL	1
0F18	C40F	5	LDI	15
0F1A	36	6	KPAH	2
0F1B	C493	7	LDI	-102
0F1D	32	8	KPAL	2
0F1E	C070	9	LD	YEAR
0F20	C376	10	ST	YEAR2
0F22	C400	11	LDI	0

0F24	C871	12	ST	COUNT
0F26	C070	13	LD	YEAR2
0F23	02	14	CCL	
0F29	F4FC	15	ADI	-4
0F2B	C86B	16	ST	YEAR2
0F2D	9402	17	JP	19
0F2F	9004	18	JMP	21
0F31	A864	19	ILD	COUNT
0F33	90EF	20	JMP	12
0F35	C060	21	LD	COUNT
0F37	02	22	CCL	
0F38	F056	23	ADD	YEAR
0F3A	02	24	CCL	
0F3B	F055	25	ADD	DAY
0F3D	C854	26	ST	TOT
0F3F	C04F	27	LD	YEAR
0F41	02	28	CCL	
0F42	F4FC	29	ADI	-4
0F44	C862	30	ST	YEAR2
0F46	94F9	31	JP	23
0F48	02	32	CCL	
0F49	F404	33	ADI	4
0F4B	C847	34	ST	REMAINDER
0F4D	9C0B	35	JNZ	42
0F4F	C041	36	LD	MONTH
0F51	02	37	CCL	
0F52	F4FD	38	ADI	-3
0F54	9404	39	JP	42
0F56	C4FF	40	LDI	-1
0F58	9002	41	JMP	43
0F5A	C400	42	LDI	0
0F5C	C837	43	ST	VAL
0F5E	C031	44	LD	NINTH
0F60	C802	45	ST	PER
0F62	C101	46	LD	+1(1)
0F64	02	47	CCL	
0F65	F02E	48	ADD	VAL
0F67	02	49	CCL	
0F68	F029	50	ADD	TOT
0F6A	C827	51	ST	TOT
0F6C	C021	52	LD	CENT
0F6E	C802	53	ST	UNK2
0F70	C201	54	LD	+1(2)
0F72	02	55	CCL	
0F73	F01E	56	ADD	TOT
0F75	C81C	57	ST	TOT
0F77	02	58	CCL	
0F78	F4F9	59	ADI	-7
0F7A	C81C	60	ST	TOT2
0F7C	94F9	61	JP	53
0F7E	02	62	CCL	
0F7F	F407	63	ADI	7
0F81	9404	64	JP	57
0F83	F4F9	65	ADI	-7
0F85	90F0	66	JMP	53
0F87	C30D	67	ST	ANSWER
0F89	901D	68	JMP	12ND PART1

0F8E	CENT	e.g. 13		
0F8F	YEAR	e.g. 47		
0F90	MONTH	e.g. 00		
0F91	DAY	e.g. 25		
0F92	TOT			
0F93	REMAINDER)			
0F94	VAL)			
0F95	ANSWER)			
0F96	COUNT)			
0F97	YEAR2)			
0F97	TOT2)			all need not be entered.
0F97	ANSWER2)			
0F64	UNK)			
0F74	UNK2)			
0F93	1			
0F99	4			
0F9A	4			
0F9B	3			
0F9C	2			
0F9D	5			
0F9E	9			
0F9F	3			
0FA0	6			
0FA1	1			
0FA2	4			
0FA3	6			
0FA4	4			
0FA5	2			
0FA6	9			
0FA7	6			
0F83	C0EC	1	LD	ANSWER
0F8A	C3EC	2	ST	ANSWER2
0FAC	C401	3	LDI	1
0F8E	C3E7	4	ST	COUNT
0F83	C0E6	5	LD	ANSWER2
0F82	02	6	CCL	
0F83	F0E1	7	ADD	ANSWER
0F85	C3E1	8	ST	ANSWER2
0FB7	B3DE	9	LDL	COUNT
0FB9	94F3	10	JP	4
0FB5	02	11	CCL	
0FBC	C409	12	LDI	9
0F8E	35	13	KPAH	1
0FBF	C403	14	LDI	3
0FC1	31	15	KPAL	1
0FC2	C40F	16	LDI	15
0FC4	36	17	KPAH	2
0FC5	C4D9	18	LDI	-39
0FC7	02	19	CCL	
0FC3	F0CE	20	ADD	ANSWER2
0FCA	32	21	KPAL	2
0FCB	C403	22	LDI	3
0FCD	C3C3	23	ST	COUNT
0FCF	C601	24	LD	0+1(2)
0FD1	C001	25	ST	0+1(1)
0FD3	B3C2	26	LDL	COUNT
0FD5	98E4	27	JP	11
0FD7	90F6	28	JMP	24
0FD9	73(T)	0FE0	3F(D)	3E(C)
0FDA	77(A)	0FE1	37(H)	3E(D)
0FDB	6D(S)	0FE2	79(E)	0FE8
0FDC	54(W)	0FE3	3E(C)	76(H)
0FDD	3E(U)	0FE4	73(T)	73(T)
0FDE	6D(S)	0FE5	3E(D)	30(I)
0FDF	54(W)	0FE6	79(E)	50(R)
				71(F)

{end of issue 6]

COMPLEMENT AND INTERNATIONAL Issue 6 DECEMBER 1979-JANUARY 1980

Bonjour a la SC/MP useres everywhere, cette newsletter est from maintenant on est etre ecrire en Francais. NO? Oh all right! Hard to believe I once passed a French O level. The reason for this foolishness is that Complement and Add has been read in France, Belgium, Denmark (I hope) and South Africa. Not by many people, I will grant you, but still it has travelled further than me. Lots and lots has been happening recently, so here is a quick round up, (as opposed to truncate), firstly on meetings.

In the Canterbury area, John Leach organised a meeting for those in his area. I could not attend, but was informed that only a handful of SC/MP fanatics turned up. In London, as mentioned last issue, though unfortunately too late for some, Martin Stamp hosted an informal meeting. Five of us were there, Me, Paul Kaufman, Bob Bowdler, Martin Stamp (surprise!) and Nick Toop from S of C fame. (Now at Acorn) Bob had his system up and running with a PE VDU. Hopefully I can retrieve some of Bob's software for this mag, though some of it runs on a special monitor for the VDU running in his 1 1/2K ram. For those without such luxuries, he had a rip off of 'Simon' called Charlie, which (for those who haven't played with the WH Smith selection of electronic games) gives you an ever increasing stream of numbers which you must repeat. (pardon) The program blows a raspberry if you get the numbers wrong! Nick Toop was very helpful over explaining certain aspects of the MK14 and VDU. One of the things that came out of the meeting was that the 8070 series may never hit the popular market, nat Semi aren't aiming for the same market as the 8060.

For those who by some fluke of fate, or the P.O., have not seen issue 5 yet, Clive Isbell is still after you lot in East Anglia. See the revised list of members for his address, it would be nice to start up a sub-group over there. Anyone wanting to do the same is of course welcome, let me know in advance when you can hold meetings, in time for the newsletters. Forget what I said about Breadboard tickets, I never did get round to it - its too late now anyway! If you go there, I will be wearing a homemade MK14 badge! I'll probably go on the Saturday around 11.00. Don't think S. of C. will be there, but Acorn will be. Sorry about the delays in writing, I'm often waiting for the return of something you may have asked for, such as the Basic Assembler listing. But please do keep on writing

Staples will no longer appear, as they are only a nuisance to those who (for some strange reason) want to photocopy bits. Also, apologies to anyone receiving more than 1 newsletter, and to those who have not received a particular one. An example of delays that can occur are for instance R, Gilchrist moving Off to the States without letting me know, and someone moving without

2)

giving me a two month notice. Hope that doesn't sound too official! Bits and pieces available

I'm selling these things for stamps, cos it saves me buying them, and also saves the trouble of paying in silly amounts to the bank. I only charge the cost price anyway which is just for the photocopying (6p each).

SC/MP Assembler in basic (v. Odd dialect), 20p of stamps.

Each issue of C&A (4 onwards)

70p of stamps.

Issue 1 35p

Issue 2 45p

Issue 3 50p

11 Maze program 45p

How the above works!! 40p

Mastermind prog (machine guesses) 40p

To borrow any item for a month, 10p of stamps per item.

Oh yes, a couple more, Addressing modes a 4 page wonder for 35p of stamps. same but different - 7 pages

SC/MP label assembler, as in no.5 40p

Life program for PE VDU, for PE VDU soon. 30p

As it is nearing Christmas, here are some Christmas tunes from Brian Johnson intended for use with the music box program. So just put in the bytes listed instead of the standard 'God save our Queen'

[Am not OCRing these, see image]

Bad joke no. 1, via Mike Grundy. He suggested I have a stall at Breadboard called an 'Exchange Program Counter'.

My joke is : What is green & silver and lies in two pieces in the bin? Answer: An MK14 with all the ICs in back to front! Oh sorry...

3)

I've put a page of hasty notes on adding another 1K of ram, I'm not really a hardware fanatic, so if I can expand my system anyone can! The Editor of Recreational Computing has asked if anyone in the group is using a MK14 for education in anyway - so please write if so, By the way, I have now built a VDU like Paul, it seems to work very well. From a hardware point of view its limitations seem to be just to be an incorrect vertical position - the top is just about visible on my trusted portable, whilst the lowest position is plenty inches off the bottom. Last week it was used at a local Christmas Fete, as a Fruit machine, It was so popular with the kids that I was asked to do another 'gig' (joke) at another fete the following week. The original program worked in 1k, but now it does clever things like displaying Machine fruit' or 'Frut Machine and doing a correction! In graphics mode, the only achievement is an image of the USS Enterprise! On a serious note, the VDU can make a good device for showing program output - no multiplexing!

Some Stop press news! I received a letter from Lintronics with a catalogue of their goods, Obviously I can't put it all here - I'm sure they will send a catalogue for a SAE. They are selling ready made MK14's at £58, and a keyboard to fit at £10,00, a quid off the normal price.

The other point of interest is a £16 power supply (5V @ 2A) which doesn't sound too bad. They can also supply other power ranges. Address: Lintronics, 37A, Chiltern AV, Bushey, Herts.

I will now give some details about writing games programs that learn! There is no point in me listing the program here as it was intended for a 1 1/2K ram area! But hopefully the ideas will help you get started on something worthwhile!

Writing Programs for Board Games.

I have unfortunately lost the tape dump of the complete program but I can recall roughly how the program behaved. You must first define the rules of the game, you will always have a board (n by n grid) which should be numbered so that moving from one square to another is a constant addition or subtraction. Also there will be two sets of pieces, one for each player, so the number of these must be known. The first thing to think of is whether to store game information as

1) A picture of the whole board, i.e. each location indicates what piece is at the board position.

2) A picture of each piece and its position number on the board - this is the quickest way of dealing with most games.

Elaborating on this, we can say: To move a piece, just delete the entry and add a new one for the revised position.

If a piece is removed, then put '20' into the current entry. As an example, the game I designed was called Quad pawn, based on Byte Magazine's Hex pawn. The board is a 4 by 4 al-air with 4 black pawns at one end, 4 white pawns at the other. The pawns move like in chess, i.e. forward 1 square, but diagonally forward when taking an opponent.

The object is to either get to the other end (any one piece) or to stop the opponent from being able to move. So by numbering the board as shown, you only need 8 bytes to hold the status of the game, as below.

4)

The best way of validating a possible move, and the way I programmed Quad pawn was to use a subroutine which took two input parameters and produced a code which gave info as to whether the move would go to an empty square, one occupied by you, or a position occupied by the opponent. The major point about Quad pawn is that the game learnt by its mistakes! Whenever it was your turn, you could reply with "GO" to indicate that the machine had lost. The machine would then store the status of both players into a table of bad moves. The table would then be used on every move by the machine, so that it would not repeat a mistake. If there were no possible moves available to the machine, it would resign, storing the previous position onto its table! All these things are easy to program, if you take a bit at a time. Most of the storage was used up in storing bad moves - about 700 bytes in the end. Eventually the machine could not seem to lose, I must resurrect the program some time, just to see if I can beat it some now!

Some Stop Press Stuff

Letter received from DJD of Practical Electronics asking for any interesting material, so have sent him some stuff! Computer Centre are offering the MK14 at £38 if anyone wants another one. Write to: Computercentre, 34-36, St. Helens Rd, Westcliff on Sea, Essex. Leaflet enclosed in one circulation. The next issue is by the way, a birthday, the first newsletter being sent out in February of this year! Page 5 of the newsletter is a rough lay out of some idea for an extra 1K of memory. I am not the hardware sort, so don't expect too much!

Time for another program, from Graham Turrell in Suffolk, The program calculates, then displays, the day of the week of any year between 1752 and 2099. The data should be entered as follows:

First two digits of year: OF8E Last two digits of year : OF8F Month 1 to 12
OF90 Day of month

OF91 All these must be calculated in hexadecimal. E.g. 25th December 79 is 13,4F,00,19. The program was compiled on the SC/MP assembler he is developing.

```
OF12 C4 0F LDI 15
OF14 35 XPAH 1
OF15 C4 97 LDI -105
OF17 31 XPAL 1
OF18 C4 0F LDI 15
OF1A 36 XPAH 2
OF1B C4 93 LDI -109
OF1D 32 XPAL 2
OF1E C0 70 LD YEAR
OF20 C8 76 ST YEAR2
OF22 C4 00 LDI 00
```

5)

```
(12) OF24 C8 71 ST COUNT
OF26 C0 70 LD YEAR2
OF28 02 CCL
OF29 F4 FC ADI -4
OF2B C8 6B ST YEAR2
```

0F2D 94 02 JP 19
0F2F 90 04 JMP 21
(19) 0F31 A8 64 ILD COUNT
0F33 90 EF JMP 12
0F35 C0 60 LD COUNT
0F37 02 CCL
0F38 F0 56 ADD YEAR
0F3A 02 CCL
0F3B F0 55 ADD DAY
0F3D C8 54 ST TOT
0F3F C0 4F LD YEAR
(28) 0F41 02 CCL
0F42 F4 FC ADI -4
0F44 C8 52 ST YEAR2
0F46 94 F9 JP 28
0F48 02 CCL
0F49 F4 04 ADI 4
0F4B C8 47 ST REMAINDER
0F4D 9C 0B JNZ 42
0F4F C0 40 LD MONTH
0F51 02 CCL
0F52 F4 FD ADI -3
0F54 94 04 JP 42
0F56 C4 FF LDI -1
0F58 90 02 JMP 43
(42) 0F5A C4 00 LDI 0
(43) 0F5C C8 37 ST VAL
0F5E C0 31 LD MONTH
0F60 C8 02 ST UNK
0F62 C1 01 LD 1(1)
0F64 02 CCL
0F65 F0 2E ADD VAL
0F67 02 CCL
0F68 F0 29 ADD TOT
0F6A C8 27 ST TOT
0F6C C0 21 LD CENT
0F6E C8 02 ST UNK2
0F70 C2 01 LD +1(2)
0F72 02 CCL
0F73 F0 1E ADD TOT
(57) 0F75 C8 1C ST TOT
(58) 0F77 02 CCL
0F78 F4 F9 ADI -7
0F7A C8 1C ST TOT2
0F7C 94 F9 JP 58
0F7E 02 CCL
0F7F F4 07 ADI 7
0F81 94 04 JP 67
0F83 F4 F9 ADI -7
0F85 90 F0 JMP 58
0F87 C8 0D ST ANSWER
0F89 90 1D JMP [2ND PART]

6)
0F8E CENT e.g. 13 (Century)
0F8F YEAR
0F90 MONTH
0F91 DAY
0F92 TOT
0F93 REMAINDER
0F94 VAL

0F95 ANSWER
0F96 COUNT
0F97 YEAR2
0F97 TOT2
0F97 ANSWER2
0F97 UNK
0F97 UNK2
0F98 1
0F99 4
0F9A 4
0F9B 0
0F9C 2
0F9D 5
0F9E 0
0F9F 3
0FA0 6
0FA1 1
0FA2 4
0FA3 6
0FA4 4
0FA5 2
0FA6 0
0FA7 6
0FA8 C0 EC LD ANSWER
0FAA C8 EC ST ANSWER2
0FAC C4 01 LDI 1
4) 0FAE C8 E7 ST COUNT
0FB0 C0 E6 LD ANSWER2
0FB2 02 CCL
0FB3 F0 E1 ADD ANSWER
0FB5 C8 E1 ST ANSWER2
0FB7 E8 DE DLD COUNT
0FB9 94 F3 JP 4
11) 0FBB 02 CCL
0FBC C4 09 LDI 9
0FBE 35 XPAH 1
0FBF C4 03 LDI 3
0FC1 31 XPAL 1
0FC2 C4 0F LDI 15
0FC4 36 XPAH 2
0FC5 C4 D9 LDI -39
0FC7 02 CCL
0FC8 F0 CE ADD ANSWER2
0FCA 32 XPAL 2
0FCB C4 03 LDI 3
0FCD C8 C8 ST COUNT
24) 0FCF C6 01 LD @+1(2)
0FD1 CD 01 ST @+1(1)
0FD3 B8 C2 DLD COUNT
0FD5 98 E4 JZ 11
0FD7 90 F6 JMP 24

[NOTE 2019: NO IDEA WHAT THE FOLLOWING IS!]

0FD9 78 77 6d 54 3E 6D 54
0FE0 3F 37 79 3E 78 6E 79
0FE7 3E 3E 76 78 30 50 71