

```

/*****/
/*                                     Prototype HP15C Calculator          */
/*                                     James C. Ullrey                    */
/*                                     INRESO                               */
/*                                     © 1990                             */
/*                                     Version      11.97a                */
/*                                     */
/*                                     Version History:                    */
/*                                     */
/* Friday 08/24/90      Started Project      V .89698a                */
/*                                     */
/* Wednesday 09/05/90   Solved display of fractional part formatting. */
/*                                     */
/* Thursday 09/06/90    Can now enter three numbers & use two operators */
/*                                     to produce result.                */
/*                                     */
/* Tuesday 09/11/90     Started V 1.96988a                             */
/*                                     Abandoned dialog box interface    */
/*                                     */
/* Friday 09/14/90      Updates to calculator interface work!          */
/*                                     */
/* Saturday 01/05/91    Started V 2.96988a                             */
/*                                     Reactivated project on fx.         */
/*                                     Debugger does not work!            */
/*                                     */
/* Sunday 01/13/91      Debugger works, rollup rolldown works,          */
/*                                     combinatorial analysis implemented. */
/*                                     */
/* Tuesday 02/05/91     Floating point-fixed point display works after  */
/*                                     a fashion. Random number generator works. */
/*                                     */
/* Wednesday 02/06/91   Started V 3.96988a                             */
/*                                     Committed to use of State vector and masks. */
/*                                     */
/* Sunday 02/10/91      Buggy version runs                             */
/*                                     */
/* Monday 02/11/91      Discovered that HP15C has 20 registers,          */
/*                                     not just 10.                        */
/*                                     */
/* Wednesday 02/14/91   Store & Recall with plus, minus, times and     */
/*                                     divide with those 20 registers are operational. */
/*                                     */
/* Friday 02/15/91      Duplication in numbers of HandleButtonEvent     */

```

/*			reduced by creating function and passing	*/	
/*			parameters.	*/	
/*					*/
/*	Saturday	02/16/91	Started V 4.96988a	*/	
/*			Programmability subproject started!	*/	
/*					*/
/*	Wednesdy	02/20/91	File I/O implemented! 20 registers and the	*/	
/*			operand stack are saved across quits.	*/	
/*					*/
/*	Thursday	02/21/91	Linked list data structure implemented for	*/	
/*			storing flags to be used by the interpreter	*/	
/*					*/
/*	Monday	02/25/91	Designated as V 5.96988a	*/	
/*			Program enunciator displays program count and	*/	
/*			button position for simple instructions	*/	
/*					*/
/*	Friday 03/15/91		Implemented multiple files with the help of	*/	
/*			Eric Slosser of SciComp Software,		*/
/*			publisher of PopUpFuncs.	*/	
/*					*/
/*			->Buy PopUpFuncs<-		*/
/*			To order PopUpFuncs call 1-800-522-5939	*/	
/*					*/
/*	Wednesdy	01/01/92	Program enunciator works better	*/	
/*					*/
/*	Thursday	01/02/92	Added lblWaitFlag so that the LBL	*/	
/*			functionality can be implemented	*/	
/*					*/
/*	Sunday	01/05/92	Designated as V 6.96988a	*/	
/*			Started work on programmability functionality.	*/	
/*					*/
/*	Thursday	01/09/92	Designated as V 7.96988a	*/	
/*			Recognized the need to alter the structure of	*/	
/*			the pNode by adding another field, pos4.	*/	
/*					*/
/*	Sunday	01/19/92	Designated as 8.97a		*/
/*			The long awaited reorganization. Breaking the	*/	
/*			code into smaller segments and adding	*/	
/*			prototypes.	*/	
/*			The start of a professional style application.		*/
/*					*/
/*	Monday	01/20/92	Prototyping complete, compiled with Think C	*/	
/*			v 5.0.2. Major runtime errors	*/	

/*					*/
/*	Monday	01/20/92	Solved the problem of writing to the file in	*/	
/*			the system folder with the help of Eric	*/	
/*			Slosser. Buy PopUpFuncs!!		*/
/*			To order PopUpFuncs call 1-800-522-5939	*/	
/*					*/
/*	Wednesdy	01/22/92	Discovered the STO A matrix functions, the	*/	
/*			STO + A matrix functions, the exchange X with	*/	
/*			Registers 0 • • 9, .0 • • .9, and (i)	*/	
/*			Registers.		*/
/*			The gState variable is becoming impacted.	*/	
/*			sWaitFlag and rWaitFlag can probably be	*/	
/*			eliminated, by, instead of clearing stoFlag,	*/	
/*			and setting sWaitFlag and plus, etc. to	*/	
/*			achieve the same result, leave the stoFlag set	*/	
/*			and set the waitFlag, which is underutilized.	*/	
/*			This will result in the elimination of these	*/	
/*			two flags, allowing the bits to be used for	*/	
/*			other functions.		*/
/*			Discovered that the test function of the minus	*/	
/*			key needs a wait flag and the waitFlag in	*/	
/*			combination with the minusFlag can probably	*/	
/*			serve that function.	*/	
/*			Such a drastic reorganization of the structure	*/	
/*			is best tested on a subsequent version	*/	
/*			of the program.		*/
/*					*/
/*	Tuesday	01/28/92	Added the Index register field and the memory	*/	
/*			status field to the map struct.	*/	
/*					*/
/*	Friday 01/31/92	Designated as 9.97a		*/	
/*			Found it necessary to use an array	*/	
/*			for the registers		*/
/*					*/
/*	Monday	02/03/92	It is necessary to dispense with the idea that	*/	
/*			a linked list can be used for the storage of	*/	
/*			program instructions. This change, however	*/	
/*			opens up the possibilities for storing programs	*/	
/*			as files, allowing program lifetimes longer	*/	
/*			than a single session.		*/
/*					*/
/*	Friday 02/07/92	Started dealing with complex numbers.		*/	
/*					*/

/*	Saturday	02/08/92	Complex multiplication and division work.	*/	
/*			(really early Sunday morning)	*/	
/*					*/
/*	Sunday	02/09/92	Complex square root works(4:47 AM).	*/	
/*					*/
/*	Monday	02/10/92	Created memory manager segment(3 AM)	*/	
/*					*/
/*	Monday	02/10/92	Designated as 10.97a		*/
/*			Discovered that a matrix descriptor can be	*/	
/*			stored and recalled from any of the registers	*/	
/*			including the index register as well as on the	*/	
/*			stack. This makes things difficult in the	*/	
/*			framework of the current structure of the	*/	
/*			program. Methinks that the stack elements can	*/	
/*			be, instead of doubles, structs, with a flag	*/	
/*			field, a double field, and a char field. When	*/	
/*			a matrix descriptor is created as a string,	*/	
/*			which is only done when the key sequence	*/	
/*			RCL MATRIX A is pressed, it is done so that it	*/	
/*			may be displayed in the display window. At	*/	
/*			this time the string can be stored in the char	*/	
/*			field of the struct. The flag field of the	*/	
/*			struct can be set to so indicate that the	*/	
/*			string is present. Checking this field will	*/	
/*			allow the deparser to display the matrix	*/	
/*			descriptor if it is present. Currently, the	*/	
/*			stack manipulations involve copying the value	*/	
/*			from one stack variable to another as the stack	*/	
/*			is scrolled, so to speak, through the display.	*/	
/*			If the string is carried around in the struct	*/	
/*			this may cause problems as copying stirngs	*/	
/*			from one variable to another may involve the	*/	
/*			necessity of using strcpy or some sort.	*/	
/*			I think that this is unnecessary and instead	*/	
/*			of carrying around the string, the struct will	*/	
/*			just reference the matrix, and the deparser can	*/	
/*			regenerate the string when it is needed from	*/	
/*			the information in the matdscr struct.	*/	
/*					*/
/*	Wednesdy	02/26/92	Use LJ Courier 12, tabs 2 for printing code.	*/	
/*					*/
/*	Wednesdy	05/06/92	Resumed writing code after income taxes,	*/	
/*			transmission repair and moving.	*/	

/*					*/
/*	Thursday	06/19/92	Fixed machine dependent timing loop.	*/	
/*					*/
/*	Friday	06/19/92	The code to allow dimensioning of a matrix exists	*/	
/*			only for the matrix A.		*/
/*			A test of storing values in the matrix elements	*/	
/*			for matrix A using USER mode resulted in the		*/
/*			values being stored in the nth + 1 matrix element	*/	
/*			instead of the nth matrix element.		*/
/*					*/
/*	Monday	06/29/92	Recall of the matrix elements in USER mode now	*/	
/*			appears to work correctly for martix A.	*/	
/*					*/
/*	Sunday	07/05/92	Code has been changed to allow dimensioning of	*/	
/*			matrices A, B and C. The preliminary step to allow	*/	
/*			matrix arithmetic	*/	
/*					*/
/*			Enunciators for USER, f, g, C and PRGM work.	*/	
/*					*/
/*	Monday	07/06/92	An imaginary last x register needs to be added.	*/	
/*			Actually, when tested, the imaginary last x		*/
/*			exists.	*/	
/*					*/
/*	Thursday	07/09/92	Modified File so that Open is disabled when the	*/	
/*			calculator window is displayed, Close is enabled.	*/	
/*			Open is enabled when the calculator window is	*/	
/*			closed, Close is disabled. Thanks to Nicky for	*/	
/*			the beta test.	*/	
/*					*/
/*	Tuesday	07/14/92	Now have a basis for the memory manager,	*/	
/*			however it is not working yet. 6:00 AM	*/	
/*					*/
/*					*/
/*	Sunday	07/19/92	2:30 AM Storage in the matrices A, B, C and D	*/	
/*			appear to work transcending heap compaction and	*/	
/*			expansion		*/
/*					*/
/*	Monday	07/20/92	Greg Dow method for passing array of structs	*/	
/*			to functions	*/	
/*			typedef struct m	*/	
/*			{		*/
/*			...		*/
/*			...		*/

```

/*          } m;                                */
/*                                                */
/*          typedef m a[5];                      */
/*                                                */
/*          a w[5];                             */
/*                                                */
/*          void  FunctionName(a w); in prototype section */
/*                                                */
/*          To pass a struct to a function, just pass the address */
/*          of the struct.                                */
/*                                                */
/* Tuesday    07/21/92    0149 hours: Matrix multiplication has worked in */
/*          two instances. In the first case a 2 x 2 matrix */
/*          in A was multiplied by a 2 x 2 matrix in B to give */
/*          the correct result in C. In the second case a */
/*          2 x 3 matrix in A was multiplied by a 3 x 4 matrix */
/*          in B to give the correct result in C. */
/*          2337 hours: fixed the label buttons so that */
/*          RCL DIM label will put the matrix's #'s of rows in */
/*          the y register, the #'s of cols in the x register. */
/*          It was trivial. NOT!! */
/*          Wednesday    07/22/92    0402 hrs: STO and RCL with the stack now works. */
/*          Wednesday    07/22/92    1427 hours: Designated as version 11.97a */
/*          Sunday        07/26/92    Store and recall indirect appear to work with */
/*          matrix descriptors and numbers. Testing appears */
/*          to be formidable */
/*          Tuesday       07/28/92    Finger cursor works. Numeric keypad entry works, */
/*          as well as f, g, F, G, s or S for sto, r or R for */
/*          recall, the arrow buttons for roll up, roll down, */
/*          erase and exchange x and y, +, -, *, /, and enter. */
/*          36073 lines of code. */
/*          Wednesday     07/29/92    STO (RCL) { +, -, *, / } (i) seems to work. */
/*          0200hrs. */
/*          STO (RCL) { +, -, *, / } I seems to work. */
/*          0250hrs. */
/*          Monday        08/17/92    Dragging the window in the second monitor

```

/*			now works.	*/	
/*			Thank you Spec Bowers of Bowers Development	*/	
/*					*/
/*	Wednesdy	09/30/92	Fixed obscure bug that caused negative numbers	*/	
/*			to appear in floating point format. Change was in	*/	
/*			Deparse()		*/
/*					*/
/*	Thursday	10/22/92	In the EtoXButton there are two statements	*/	
/*			gState ^= rcl. The first turns the flag off as it	*/	
/*			should, the second turns it back on. One of these	*/	
/*			is excess. The first one is the excess one.	*/	
/*					*/
/*	Saturday	10/24/92	Added a new menu and found out about a gotcha	*/	
/*			with ResEdit whereby when adding menus, one	*/	
/*			must not only "Create New Resource" under the	*/	
/*			RESOURCE menu and change its ID with the Get	*/	
/*			Info, but also open the resource using "Open Using	*/	
/*			Template..." menu item in the RESOURCE menu and	*/	
/*			change the ID there as well. Thanks once more to	*/	
/*			Eric Slosser	*/	
/*					*/
/*	Sunday	10/25/92	Created window resources for six windows, one	*/	
/*			each for each of the matrices and one for the	*/	
/*			registers. Added code to make them appear and	*/	
/*			disapear when the Windows: menu items are	*/	
/*			checked or unchecked.	*/	
/*					*/
/*	Monday	10/26/92	When in USER mode or complex mode, the	*/	
/*			enunciators were not updated after another	*/	
/*			window was removed from in front of the	*/	
/*			calculator window. They were also not maintained	*/	
/*			across quits. This has been fixed.	*/	
/*					*/
/*			When one of the matrix windows is the front	*/	
/*			window, it gets closed when the close menu item	*/	
/*			from the file menu is selected, a use which was	*/	
/*			not intended.	*/	
/*					*/
/*			The matrix, register and calculator windows now	*/	
/*			appear to work correctly across closing and	*/	
/*			opening selections of the menu, and the check	*/	
/*			marks for the items in the window menu work	*/	

/*			correctly.	*/	
/*				*/	
/*	Saturday	10/31/92	The map struct had a bug whereby 65 registers	*/	
/*			were declared, and then map.reg[65] was used,		*/
/*			when actually only the registers 0 thru 64		*/
/*			existed, and when a value was written to	*/	
/*			map.reg[65] it was written to the next struct		*/
/*			that was declared, which happened to be the lastx		*/
/*			struct. map.reg[65] is used in the matrix	*/	
/*			operations, specifically in multiplication. This		*/
/*			bug did not show up until the arithmetic buttons	*/	
/*			were used. I started using them to expedite the	*/	
/*			entry of numbers into the matrices, and when I did	*/	
/*			the lastx register was affected, and thus the value		*/
/*			of map.reg[65], which was supposed to hold one of		*/
/*			the matrix elements. Without the alteration of the		*/
/*			lastx register the problem didn't show up and the		*/
/*			matrix multiplication worked using the lastx		*/
/*			memory location. A quick fix was affected by		*/
/*			delairing the registers to 66 thus avoiding having		*/
/*			to fix the addressing modes for the registers.		*/
/*					*/
/*	Tuesday	11/10/92	The matrix and register windows appear to work		*/
/*			correctly Matrix multiplication by scalars works	*/	
/*			when result is operand.		*/
/*					*/
/*	Thursday	11/12/92	When a matrix dimension is expanded, the new		*/
/*			matrix elements created contain zeros.		*/
/*					*/
/*	Friday 11/13/92	Addition of two matrices works.		*/	
/*					*/
/*	Tuesday	11/17/92	The matrix and register windows seem to work	*/	
/*			correctly after several extensive debugging		*/
/*			sessions in which two serious bugs in the memory		*/
/*			manager were discovered and corrected. Now a		*/
/*			matrix can be created in any order, its dimension		*/
/*			changed, reduced to 0 x 0, recreated, be subjected		*/
/*			to alteration by its being the result matrix,		*/
/*			participate in arithmetic operations, and the		*/
/*			matrix window will display the correct contents,	*/	
/*			and the register window will also display the		*/
/*			correct contents, shifting the values to different	*/	


```
/* registers as the applications' heap has its  
/* integrity maintained. The matrix operations work
```

```
*/  
*/
```

/*			save for matrix inversion.	*/	
/*					*/
/*	Wednesdy	11/25/92	mStrkPtr = (struct mStrk *)NewPtr(sizeof(mStrk));	*/	
/*			mStrkHan = (struct mStrk **)NewHandle(sizeof(mStrk));	*/	
/*			A pointer or handle MUST be typecast so the	*/	
/*			compiler will know how to evaluate an expression	*/	
/*			"myStrkPtr.number" or "***myStrkHan.number"	*/	
/*					*/
/*	Wednesdy	12/02/92	Designated as 12.97a		*/
/*			Started the addition of the help module, using as a	*/	
/*			model code from Chassis 4.3.1, authored by	*/	
/*			Charles A. Hoffman, A person who I had known by proxy	*/	
/*			for 27 years, but only met last June at MacHack.	*/	
/*					*/
/*	Sunday	12/06/92	Implemented the Transmogrifier button, the key-click	*/	
/*			combination of the holding down the command key and	*/	
/*			clicking on the HP icon. Selecting this option converts	*/	
/*			the calculator face to a smaller face, thus	*/	
/*			accomodating smaller screens while using the matrix	*/	
/*			and register windows.		*/
/*					*/
/*	Wednesdy	12/09/92	Started programmability project	*/	
/*			Fixed a bug that caused the program enunciator display	*/	
/*			to be replaced by the stack value after updates	*/	
/*			following the calculator window being obscured by	*/	
/*			another window.	*/	
/*					*/
/*	Friday	12/11/92	Added spash screen. The main display is now a bitmap.	*/	
/*			Now screen refresh is faster.	*/	
/*					*/
/*	Saturday	12/12/92	Can now transmogrify while entering a string into the	*/	
/*			display with out losing the keystrokes.	*/	
/*			Fixed a bug whereby when in program mode, selecting	*/	
/*			g MEM would cause, after display of the memory config,	*/	
/*			display of the stack value instead of the program step	*/	
/*			enunciator.	*/	
/*					*/
/*	Tuesday	12/15/92	The monitor is tested for the presence of color.	*/	
/*			A color splash screen is displayed for color machines,	*/	
/*			a black and white splash screen is displayed for black	*/	
/*			and white machines. Both the regular and transmogrified	*/	
/*			calculator faces are dragable by the HP logo box. The	*/	
/*			transmogrified calculator face is automatically	*/	

/*			positioned in the lower left corner of the screen	*/	
/*			dynamically at run time.	*/	
/*					*/
/*	Wednesdy	12/16/92	Added Save Program and Load Program items to the file	*/	
/*			menu.	*/	
/*					*/
/*	Thursday	12/17/92	GTO CHS nnn goes to line number nnn.	*/	
/*			GTO I	*/	
/*			GTO 0 thru 9, .0 thru .9 will branch to location	*/	
/*			established by, in prgm mode f LBL 0 thru 9,	*/	
/*			.0 thru .9 . similarly for labels A thru E.	*/	
/*			f 4 (x><) LBL exchanges x with	*/	
/*			The % & delta % functions now work.	*/	
/*					*/
/*					*/
/*	Sunday	12/20/92	The new program enunciator seems to work	*/	
/*					*/
/*	Thursday	12/24/92	General cleanup of the gState vector maintenance in	*/	
/*			progress.		*/
/*					*/
/*	Wednesdy	12/30/92	Put code under the clear flag, set flag and test flag	*/	
/*			functionalities. Put code under the x<=y	*/	
/*			button(g divide), the x=0 button(g times) and the TEST	*/	
/*			button(g minus). Programmability seems to work.	*/	
/*			Extensive testing is necessary.	*/	
/*					*/
/*					*/
/*	Thursday	12/31/92	Program editing for insertion of instructions in the	*/	
/*			middle of an existing program works.	*/	
/*					*/
/*	Friday 01/01/93		Program editing for deletion of instructions in the	*/	
/*			middle of an existing program works.	*/	
/*					*/
/*	Tuesday	01/05/93	Programmability seems to work. Multiple and nested	*/	
/*			GSB (Go to SuBroutine) calls have been tested using both	*/	
/*			the labels A..E and some of the n and .n labels.	*/	
/*					*/
/*					*/
/*****					