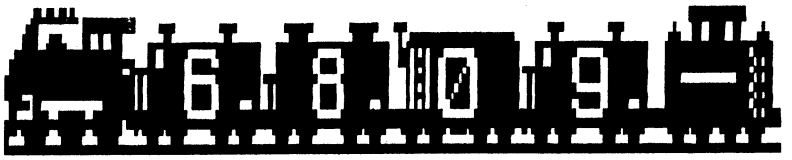
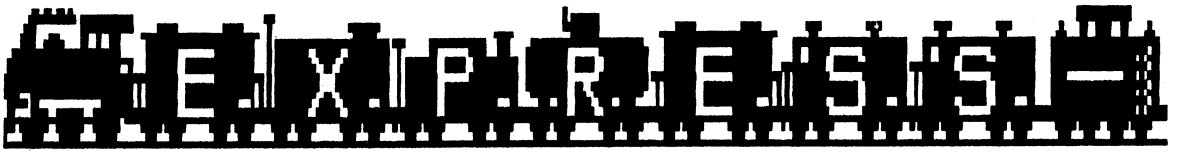




The Official Publication of The

PENN-JERSEY COLOR COMPUTER CLUB

MAVERICK BBS

215-760-0456

MAY 1993

Rick Hengeveld Announces

BIG CHANGE FOR PJCCC

The PJCCC has reached a critical point in it's history. This is one of the longest running user groups in the area, while the computer industry kept moving along and went through great periods of upheaval the PJCCC remained a constant. To exist for over ten years and support the Tandy Color Computer is a testament to both the venerable Coco and to the members of the PJCCC.

Ten years is a long period of time and it has taken it's toll on both the Coco and our membership rolls. As we all know the Coco is no longer in production and both hardware and the soft stuff are getting harder to come by. I've also noticed that our membership has dwindled over the last 3 or 4 years. Membership has decreased for a number of reasons, however the most cited reason for leaving our ranks seems to be that members are moving on to different systems and the PJCCC no longer serves their

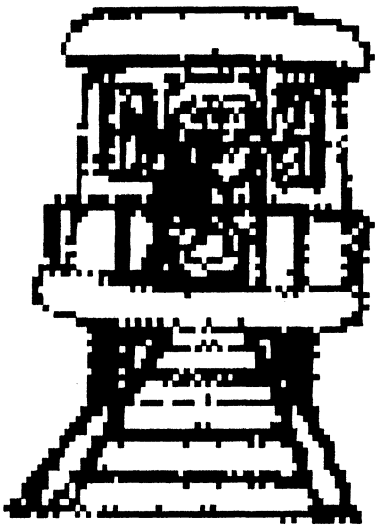
H.UNKS

purposes.

With that in mind the membership voted to revise the Constitution and open the club to anyone who is interested in computing. This, hopefully will bring in new members to revitalize our club. While this change opens great possibilities it also opens the door to the unknown. Anytime you change something that has been a sure thing for many years you enter into uncharted territory. There are many variables involved such as presentation themes and trying to gel people with widely varied opinions of computers into one solid users group. To put it bluntly, this change will either make or break this club! With the people we have in the PJCCC I feel this change will be successful! But it is going to take a good deal of effort and patience. I will ask all our members as of today to think about anyone they know who has or is interested in computing. Please take the time to invite them to a PJCCC meeting. Tell them to simply come and observe the club and they can decide for themselves if they wish to join us, no pressure.

Also the PJCCC Executive committee will be having a meeting in the near future to plan for the hopeful expansion of the club.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX



THE LIBRARY CAR

by Al Wagner

Welcome to the 6809 Library Car. Pull up something to sit on and we'll get started. First let me say that this is taking a lot longer than I had anticipated. Also due to illness of both myself and my family, as well as the Trenton Computer Festival, I have not been able to create the programs I had wanted to present this month. All I can say is I will make an effort to finally wrap up this database program we have been working on as soon as possible.In setting up to begin this installment, I found a small bug in one of the programs that were published in the last "Express". In the basedelete procedure there is a WHILE loop that allows the user to decide whether they really wanted to delete this

base or not. I found that though the procedure will work as originally published, often the loop didn't seem to catch the key that was entered. Pressing the key twice rapidly seemed to make it work. I think this is because the computer is so busy whipping around the loop that it doesn't allow enough time for the INKEY statement which is only a small part of the loop. The patch is to enter another small WHILE loop around just the INKEY statement that simply checks for the variable ans to be something other than a null. This causes the computer to pay almost exclusive attention to the keyboard until the smaller WHILE loop is satisfied. The following is an excerpt from that program that should enable you to find and correct the glitch.@@@@@@@RUN printat(20,7)1PRINT "Do you really want to delete "; filenameRUN printat(20,8)PRINT "(Y/N): " ans:=""!WHILE ans<>"Y" AND ans<>"N" DO RUN printat(28,8) PRINT " " RUN printat(28,8) WHILE ans="" DO RUN INKEY(ans) ENDWHILE RUN makeupper(ans) RUN printat(28,8) PRINT ansENDWHILE This only proves a point I recently read in a book on programing. "Although program testing may show the presence of bugs, it can't guarantee their absence."We have come to a point where we must make a decision as to how many items we will allow in each database. I had put this off as long as I could because I was trying to figure a way around having to come up with a fixed number of items. This is one command where BASIC09 lets us down a little. In C and some other languages, one can dimension an array with a variable size passed to the procedure with the call that invokes the procedure. BASIC09 does not allow for this. Hence, we must decide on the number of slots to allow for our data. Let's pick 100 as a number. This is not as bad a limit as it might first seem. Realize that this is the number of items in a particular database. Let's say you want to inventory your furniture. Do you have more than 100 pieces of furniture? If you find that the number 100 is really too small, you can make it larger. Remember though that arrays have a way of consuming memory in a hurry. Perhaps a better way of expanding the listings is to create different bases for different categories of possessions such as, furniture, computers, computer accessories, dinning stuff (china sets, utensil sets, etc.) or whatever other categories you might need. Then when you wish to access the data all you need do is call up the appropriate file. My original intent was to have everything in one file, but now that I think more about it, this may even be a better way to handle the problem.(The sort routine will have to be modified slightly

from that published last time. For one thing that was for integers and we will be sorting text. The parameters low, high, and ndata() can be fixed as dimensioned variables rather than as parameters because we will no longer have to pass this data back and forth. We will however need a parameter to pass the filename of the database we wish to sort. As we have four index files, we will have to run the sort routine four times, once for each index file. The array will be single dimensional array of a complex (vs. simplex) data type to allow the storage of both the keyword, by which to sort, and the record number. Making the data type a complex type allows us to check one part of it for sorting purposes and then move the entire entity (both keyword and record number) with one call to the routine swapping the data around.. As we won't actually be sorting the database itself, we need to refer to the records in the database by record number. To display the data in order by the selected keyword, we will call the appropriate index file and call the records in the order indicated in the index. This allows us to call the entire index file into memory, search rapidly through just the keywords, and then call just the required record into memory from the main database file.(The need to search for a specific keyword also brings up another challenge. How can we look for a particular keyword without having to look at each and every keyword? The clue is that we have the index in sorted order. Just about the quickest way to find anything in a sorted list, if you have no idea where it is, is to divide the list in half and decide if the item you are looking for is higher or lower in the list than the item in the middle. You may get very lucky and find the middle item is equal to the one you are seeking.

Let's say it is higher. Now the upper half of the original list becomes the list you are working on.

Repeat the process and divide the new list in half again. Decide again is the middle item higher, lower, or equal to the item for which you are looking. If it is not equal, pick the half of the current list in which the item should occur and repeat the process. No matter how large the list is, you will eventually come down to one item. If this is not the item sought, the item does not exist in this list. How many steps would this take in our 100 item list? Let's try dividing our list by 2 as many times as we can. 100/2=50 50/2=25 25/2=13 (Round up when you get a fraction as half an item isn't legal.) 13/2=7 7/2=3 3/2=2 2/2=1 That was 7 repeats of the search process and we would have to have our item or know it wasn't here to be had. That was alot quicker than looking at each item, wasn't it? The larger the list, the more time is saved using this

method. On a list of 1600 items, it would require only 4 more repeats of the search!Let's get into coding our indices. We haven't yet actually created an index. Yes, in the makebase routine we created a few files, but they're empty! We need to fleshout their file structure and maybe put something in them. We've already decided to make each index record contain a keyword and a database record number. I refer to a "keyWORD". It doesn't have to be a word at all. Anything we can compare to decide greater than, less than, or equal to will work just fine. It could be an integer, a real number (these can get tricky and are not recommended), a letter, a word, a line of text, or an even larger chunk of text, as long as it can be compared. The reason real numbers are not recommended is that they have 39 digits! They consist of 10 significant and 29 zero digits in scientific notation. These are all there even if we don't see them. If just one of those digits doesn't match what we expect, even if its close enough for us, the computer says,"No match."Well, due to the reasons given at the start of the column, that's it for this month. I will try to do better next time. Until then, happy computing.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

The Maverick Report

by Rick Hengeveld

The Maverick BBS Has handled over 1000 calls and continues to serve the PJCCC's needs. The overblown userlogs were recently purged down to the current active users. All user data had to be re-entered by hand, so if you have any problem logging in just reapply as a new user and leave a note addressed to "sysop" and I will sort out the problem. Now on to somewhat happier news. The Trenton computer fair has yielded a "new" system to run the BBS on. A MS-DOS system was picked up, this will allow for much greater file storage and we will finally get up

to 2400 baud speed. The only hold up seems to be a nonfunctional monochrome monitor. Poor little sucker died after 20 minutes of use! So if you have or know of where a monochrome TTL monitor can be had "cheap" let me know. While these monitors can still be had brand new the stress of running 3 systems in one household will have an adverse affect on both my wallet and my life, since my wife has threatened great bodily harm to your friendly neighborhood sysop if he drops to much cash!

Invite your friends, your relatives, your loved ones to PJCCC ! We meet at 7pm on the last Friday of each month.

ROOM 105 Northampton County Community College

The PJCCC Executive Committee