CLUB ROOMS: Playground & Recreation Assocn. H.Q. Bldg.
Love Street, Spring Hill, Brisbane.

CONTENTS

DIARY FOR JULY

Group meeting on Tuesday, 2nd July 1985, at 7.30 pm in our club rooms. Visitors are very welcome!

Rob Adamson will give an expanatory talk on
Public Domain Disk "UB"

The Beginners Corner session for new members will be conducted by Ray King

----

Workshop meeting on Sunday, 14th July 1985, from 1 pm till 5 pm in the Guidance Officer's Training Centre, Bayswater Rd. Milton.
Bring your programming- or hardware problems, as well as your own computer equipment!
Opportunity to copy the group's Public Domain Disks.

Please note that workshop meetings are for members only!

REGIONAL MEETINGS

Cannon Hill meets on the 4th Saturday of the month (7.30 pm) in the Cannon Hill State School.
Contact: Barry Wilson (VIC) - Ph.399 6204 or Augy Norman (C-64) - Ph.399 2080, a.h.

Pine Rivers meets on the 2nd and 4th Sunday of the month (1 pm - 5 pm) in the Strathpine High School (rear entrance).
Contact: Clayton Lancaster - Ph.285 4157, a.h.

Redcliffe Peninsula meets on the 1st Friday of the month (7 pm) in the Redcliffe High School.
Contact: Geoff Baillie - Ph.203 5088, a.h.

Sherwood meets on the 2nd Friday of the month (7.30 pm) in the Graceville State School.
Contact: Leigh Winsor - Ph.379 2405, a.h.

Springwood meets on the 3rd Wednesday of the month (7.30 pm) in the Springwood Central Primary School, Dennis Rd., Springwood.
Contact: Terry Steer - Ph.200 5926, a.h.

The Gap meets on the 3rd Wednesday of the month (7.30 pm) in the Gap State School
Contact: John Johnston - Ph.30 5240, a.h.

Wavell Heights meets on the 2nd Tuesday of the month (7.30 pm) in the Wavell Heights High School (library) in Brae St.
Contact: Robert Adamson - Ph.266 6353, a.h.

Killarney: meets on the 2nd Monday of the month in the Killarney State School.
Contact: Roger Frazer - Ph.(076) 641370

Maryborough: This sub-group is now up and running. Contact Terry Baade (16 Mouquet Lane, Maryborough, 4650) at 21 2271 (W) or 21 5059 (H).

Toowoomba: This sub-group is coming on line soon! See next month's newsletter for details.

We are still looking for one or more of our members to start a Sub-Group in the Sunnybank or Mt.Gravatt area !!!

IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
IMPORTANT NOTICE: Copying of Commercial Software is *not allowed* at our meetings or workshops.
Failure to comply with this regulation will result in loss of membership!
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII

SPECIAL INTEREST GROUPS

Business Sub-Group meets on the 3rd Tuesday of the month (7.30 pm) in the West End State School.
Contact: Max Bean - Ph. 208 1225, a.h.

Primary Education Sub-Group meets on the 3rd Tuesday of the month (7.30 pm) in the Aspley State School.
Contact: Bill Weeks - Ph. 208 8620 (working hours) or at 341 2523, a.h.

Adventure Games Sub-Group meets on the 1st Tuesday of the month, (during main meeting - in our club rooms).
Contact: Trevor Mancktelow - Ph. 262 4602, a.h.

EDITORIAL

Reports in American publications keep on painting a picture of gloom and doom in the micro- and home computer market. Apple is laying off staff and Commodore's sales are way below predictions etc. etc. Yet eighteen months ago these selfsame publications were predicting a very rosy future for these computers - right up to the 21st century. You know the sort of thing: "In Every Home a Computer" etc.

What has gone wrong? This is rather difficult to answer, but I would like to try all the same.

When I bought my Commodore PET in 1981 the micro computer scene was still in it's infancy. In the first place they were expensive: 16K Pet + Datasette $1200 - a single disk drive $600 - the cheapest Commodore dot matrix printer was $1100!

There was very little commercial software or books available. So I joined that recently founded Commodore User Group, and, with the other 20 odd members, tried (more or less successfully) to program the thing. We were truly "hackers".

Then came the VIC-20, followed sometime later by the C-64, and the whole picture changed. Now we had colour and sound, hurray! Time for the commercial software boys to come into the picture, and what started out as a trickle became a veritable flood of commercial software. Games, utilities, more games, word processors, more games, spread sheets, data bases, and still more games. They were telling you that you could just about do everything with your computer, except for making the bed. The other big 'come-on' was that you had to get one of those fabulous micro computers for your kid's education - without one of them your children would be disadvantaged! So Dad or Mum brought home this fantastic 'Educational Aid', and off we went playing games and more games. Dad and Mum soon discovered that the thing was pretty useless as an address book or recipe file. After all a pocket diary and a cook book are a lot easier to access!

So we went on buying and playing games till we got sick of playing games, and that of course is the crux of the matter!

Yes, an occasional youngster (usually of high school age) did get interested in computer programming, and they are the new breed of "hackers", and very welcome they are too. But the other 95% have lost interest and found other ways to fill in their spare time.

And what about those much vaunted packages of "Application Software"? Well, with the exception of word processors, they have turned out to be rather a sham. The truth is that neither the VIC nor the C-64 (nor the Atari, Amstrad, Apple 2, Tandy Colour Computer etc.) were ever designed as business machines and never will be business machines! Anybody who has ever used that (very good) spread sheet program "Multiplan" on a C-64 will know how slow it is. Originally written for much larger systems it had to be cut down to size to fit it in the limited memory space of a C-64. By not cutting down on it's features the only way to fit it into memory is by loading in one module at a time. Thus to access the program's facilities you just wait and wait and wait, till it is ready. Hardly the thing for the busy executive! A data base program on the VIC or C-64? Great if you have a 100 items, but you need plenty of patience if you have to sort a 1000 items over several indexed fields. On top of everything we are settled with a slow and limited capacity disk drive, thus limiting the business capabilities even further.

So, back to playing games. But now that a C-64 is down to $300.00 in price, do our friends at Imagineering, Ozisoft etc. really think that people will go on paying $50.00 for games? Realistically computer games should be no more than $20.00 or so, but our software distributors are rather a greedy lot. Result: lots of illegal copying - loss of sales for the dealers.

By now it must be obvious what the end result will be. When the game craze is over, the VIC's and C-64's end up in the cupboard, side by side with the Atari Game Consoles. End Of Story.

Yes I know that the above scenario is grossly exaggerated and simplified, but I feel that it is basically factual. Obviously the future does not lie with games and so-called home applications.

Both Commodore and Atari have given notice that they will now attack the small business computer market, currently controlled by the likes of IBM and Apple. This attack will be largely based on price.

However, to be successful they will have to offer at least equal or even better quality at a lower price. The average business man is certainly not going to put up with hardware as unreliable as a 1541 disk drive. If this is not forthcoming I foresee a very dim future for these companies.

And what about User Groups over the next couple of years? My chrystal ball shows a large reduction in membership (all those game players remember!). There will remain a core of members interested in programming as such, as well as the new breed of business computer users who will try to get the best out of their newly aquired business systems.

It is of course possible that my chrystal ball is just as fallible as the one used in the USA some eighteen months ago.

Only time will tell!

Ralph De Vries

NOTES & GOSSIP

June Meeting

This was our first meeting in our new club rooms. The cold weather was probably responsible for a relatively small attendance, although there were still some hundred members present. The new rooms were being painted, but by next month everything should be shipshape. We have not yet finalized positions for the libraries and the beginners corner, but we are working at it! The official part of the meeting was over in 45 minutes. This must be an alltime record! Greg Canning took charge of the beginners corner, and Fay King gave a well attended talk on Public Domain Disk no.5.

Printer News

Rumour has it that Commodore is about to release a replacement printer for the 802 (formerly the 1526), with lots of extra features. No further details known at this stage.

Epson has released a new printer, model LX80, which works at 100 cps (draft mode) and 10 cps (near letter quality mode), and has a 1K. print buffer. Price not known at this stage. Epson supplies a Commodore interface (about $130.00), which is fitted inside the printer, and draws it's power from the printer. Printer is friction feed only.

Star Micronics new SG-10 printer is available in the USA in a Commodore-ready version. This one is known as the SG-10C. It runs at 120 cps (draft mode) or 30 cps (near letter quality mode), and has a 2K. print buffer. Like the Star Gemini 10X it has both friction and tractor feed, and still uses conventional (read cheap) type writer ribbons. Regrettably the Australian distributor knows nothing about the SG-10C version, which may mean that they don't intend to import this model into Australia.

C.W. Electronics expect to receive shortly another Commodore-ready printer. Wait for further details.

Membership Discounts

If you intend to purchase computer related products from our advertisers and wish to obtain your discount, you **must produce your current membership card.** Remember: no membership card - no discount!

The New C-128 Computer

If you are interested in this new computer (expected arrival some time in August) leave your name and address with C.W. Electronics, and they will contact you on it's arrival. No deposit required and you are under no obligation to purchase.

Photo Copy Service

Our member Terry Steer (sub-group co-ordinator - coffee maker - bulk cassette copier - printer - software publisher etc. etc.), has just installed a new photo copier which does straight copying, (in black or single colour runs), as well as reduction- and enlarged copies. Terry has also facilities for ring-binding manuscripts etc. All these services are available to members at extremely attractive prices. Give Terry a ring at 808 2424 for further details.

Beginners Corner

This newsletter should have a beginners corner, but we do need somebody to write it! Any volunteers?

Basic Computer Courses

The Ithaca T.A.F.E. runs regular "Programming In Basic" courses for Commodore Computers.
These are divided into a BASIC 1 course (for beginners), and a BASIC 2 course (following on from BASIC 1).
For more details ring the T.A.F.E. college at 3699011 and ask for Betty.

Sub-Groups

Our sub-groups are pretty autonomous as regards the way they are run by their local members. If you feel however that your local group could be improved it might be a good idea to contact Terry Steer, our Sub-Group co-ordinator, at 808 2424 (after hours), who will treat your request(s) with all possible sympathy.

REVIEWS

Epyx Fast Load Cartridge for C-64 - R.R.P. $45.00
Our review copy by courtesy of Chandlers 43 Adelaide St. Brisbane

Fast (Turbo) loading for both tape and disk have been very much in the news lately. The Epyx Fast Load Cartridge is the latest entrant in the fast disk loading stakes, and definitely one of the better entries.

When you have plugged the cartridge in (remember: computer switched off when inserting a cartridge) you get the standard Commodore Basic message with all your 38911 bytes intact. Below the word "READY" is the word "FASTLOAD" - this is the only difference. To load a program you either use the Commodore key, combined with the Run/Stop key, which is equivalent to load "*",8,1 and RUN. The other forms of loading are equivalent to the load and save commands as used by the DOS Wedge. Included are the "$" directory command (leaves program in memory intact), and the ">" or "@" command to read the error channel. To send a disk command it is only necessary to type @ **COMMAND**, thus eliminating the OPEN 15,8,15, "COMMAND": CLOSE 15. So, basically you have a fast load and DOS Wedge combined.

For comparison here are some programs and the times it took to load them, both with and without Fast Load:

PROGRAM:	NORMAL LOAD:	FAST LOAD:
Easy Script	65'	55'
Consultant	73'	15'
Paper Clip	60'	17'
Print Shop	78'	33'
Jumpman	94'	25'
Multiplan	118'	94'
14K. Basic Program	35'	11'

As can be seen from the above the speeding up can be up to 5 times faster, but in certain programs the time saving is only minimal. It does load protected software such as "Print Shop", but in the unlikely event of a program not loading it is simply a matter of pressing the "D" key to disable the Fast Load Cartridge. This is a nice feature, as it means that you don't have to remove the cartridge if you don't want to use it.

I did have problems though with my data base program "The Consultant". The program loaded alright, but when trying to write new records I ran into disk access problems. Without the Fast Load cartridge these problems disappeared.

Epyx's advertisement states that "Fast Load can load, save and copy your disks five times faster than normal." This is not really true as saving a program takes place at normal (read "slow") 1541 speed. The cartridge will however copy an entire disk in just under 9 minutes, which is certainly about 3 times faster than some of the standard disk copy programs. Additionally the cartridge allows one to copy or delete a single file, "lock" or "unlock" a file (to stop unauthorized users from scratching a file), and rename a file.
There is also a simple track and sector editor built in, which is an additional plus for this cartridge.

Last but not least the cartridge has a machine code monitor on board, but this is probably not very useful, as it has a set of non-standard commands. As an example "S" in a 'standard' monitor stands for SAVE, but here "S" executes one M.L. Op-Code. Similarly "W" means walk (single step) through a program, but in Fast Load "W" stands for a Binary Save!
Although this monitor appears to have certain nice features it's overall non-compatibility with other (standard) monitors make it rather less desirable.

For those users who are looking for a good fast loading disk utility this Fast Load cartridge can be thoroughly recommended. The advantages are that the cartridge can be left in place and, if need be, completely ignored. The additional features like the Wedge commands also add considerably to it's appeal.

If you feel that you must have both fast load and save facilities than be prepared to spend a lot more than the $45.00 asked for this cartridge.

Ralph De Vries

Book Review

Easy Interfacing Projects for the Commodore 64.
by Jim Downey, Don Rindsberg and William Isherwood

Our preview copy by courtesy of Prentice Hall - R.R.P. $19.95

A very interesting collection of soft- and hardware articles on the I/O ports of the 64 which would be of use to a reader as a reference book. The articles are not written for the beginner, although the preface would lead you to believe otherwise: some reasonable knowledge of either programming, electronics, or both, would be needed if the projects are to be attempted without frustration.

Most of the projects have been printed in a similar form at some other time in various technicial magazines, but in this book the articles are accompanied by supporting software to complete the project, a combination not often found in other publications. Also, most projects appear to be built around components available from Tandy Electronics Stores in the USA, which can be very convient, but a check did reveal that a couple of "programmed ic's" are not available here.

Even if a project does not appeal, reading the article will reveal some interesting ideas for other projects. The same can be said for the accompanying software, as quite often only modifications to an existing program are needed to obtain maximum benefit from the project.

The book is recommended as interesting reading for people interested in the I/O ports of the 64, and a good reference guide for those doing serious interfacing with the 64.

Roger Haigh

IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII

MEMBERS' ADVERTS

FOR SALE

Cardprint + G Printer Interface $60.00
Contact Greg Shea at 345 2799 (after hours)

HES "TURTLE GRAPHICS II" by David Malmburg
Cartridge for C-64, with 136 page manual $50.00
Contact Peter Gadsby, 69 Galloway St., Armidale, NSW, 2350 - (Ph.067/721360)

Commodore PET 4032 Computer (9" screen) c/w Datasette and
Toolkit & Commando Chips, Software & Documentation - $350.00
1520 Printer/Plotter - $60.00
Contact Lex Hinkley at 393 3140 (after hours)

"CAL-KIT" - New Spreadsheet for C-64 (Disk & Dongle) --- $30.00
Contact Ralph de Vries at 30 3477.

****

<u>REMEMBER</u>: Member's Advertisements for Computer Equipment
(For Sale - Wanted - Swap) are free of of charge!
Post your advert to the editor to reach him by the second Tuesday of the
month for insertion in that month's issue.

When one has a large mass information it is generally essential to have the data in alphabetical or numeric order. How would we find a name in the telephone book if there was no alphabetical order? When using the computer to store information, the same factors apply, it is quicker and easier to access an item from an ordered list. The problem is however that data is rarely entered in an ordered sequence. Somewhere along the line, we must sort the data into the order we require.

Here the problems begin. One inescapable feature of every sorting routine is that the time required for the sort increases exponentially with the number of items. In BASIC, some algorithms which will sort 10 items in one second will take two minutes to sort 100 items and three hours to sort 1000 items! This is plainly unacceptable.

Another problem is the order of our initial data. Are the items almost in order, randomly distributed, or have they been previously sorted into the opposite order to that now required?

Both the number of items and their initial order as well as the type of data, strings of characters or numbers, must be taken into account when choosing the best sort for the job.

"This is all irrelevant, I'll just use the latest machine code sort from XXXX magazine." I hear you say. In some cases this is all well and good. But is it a case of finding the best solution for the particular job or simply applying the job to the solution?

Sorting in BASIC has got a bad name but unjustifiably so. It is usually the case that the wrong sorting method has been chosen. A little care in selecting the best sorting algorithm for a given application will give results in acceptable times and allow much greater flexibility in program design. For example, the time for sorting 1000 numbers can easily be reduced from three hours to just over four minutes (27 seconds with Petspeed!), if the right algorithm is used. Considering the fact that most applications require a sort of roughly 100 items or less, an the fact that a BASIC sort can perform this operation in about 20 seconds (two seconds with Petspeed), why bother with indecipherable machine code routines! (If you do wish a very fast machine code sort, see Paul Blair's article on SSORT in the Feb. edition the 'Commodore Magazine' V.4, no.3, page C6.)

Let's look at five of the most common sorting methods in BASIC which are in general use and compare their performance under three different conditions:

> Sorting a random list (the most common requirement).
> Taking a sorted list and resorting it into the reverse order.
> Adding one or two extra items to the end of the list and sorting it into place.

You will probably be surprised at the results. The last condition is actually every special case. If data is to be entered one item at a time (from the keyboard for example), and must be compiled into a sorted list, by far the best idea is to not use a standard sort at all but to simply insert the new item in the correct place. We will discuss this in more depth later.

Each of the following sorts is written in standard CBM BASIC with reasonable care for efficient programming but do not include any special tricks to speed up the execution beyond those typically in use by the average programmer. (The BASIC code can be optimized further to some degree.)

Sorting Lists

As we know, sorting is the process where different items (often known as elements) in a list are compared, and, if they are out of order, swapped. The efficiency of all sorting procedures depend on the total number of items, the number and type of comparisons, and then the number of swaps which are required. For the present, let's talk about sorting a series of randomly generated numbers (the methods apply equally to randomly input string data such as names etc. with only a few minor penalties) and for the purposes of discussion, assume we wish to sort into alphabetical or ascending order (i.e. 1,2,3,4 ..)

Bubble Sort

The most primitive sorting method is the Bubble or Ripple sort. This repetitively compares 'adjacent' items in the list. If any adjacent items are out of order, they are swapped. The process is repeated until no more swaps occur. On the first pass through the list all N items must be compared. If the data is badly out of order, it will take up to N-1 passes to sort and require N/2 comparisons and maybe N/4 swaps. Total sorting time is directly proportional to the square of the number of items. That is, for every doubling of the number of items, the sorting time increases by a factor of four. Sorting 10 random items takes roughly 1 second, 100 items roughly 105 seconds while 1000 items takes 10500 seconds or 3 hours! Adding one extra item to the bottom of

the list an resorting takes roughly half as long as a full random sort itself and a reverse sort 1.3 times as long. Not a very useful procedure.

Exchange Sort

The Exchange or Shuttle/Interchange sort is a common variation on the Bubble sort which attempts to minimise the extremes. Instead of repeatedly swapping adjacent elements until the list is in order, the first pass compares all N elements shuttles the lowest element to the head of the list. The second pass now compares the remaining N-1 elements to find the next lowest. The third pass requires only N-2 comparisons and so on. The total number of comparison passes is therefore only factorial N or N*(N-1)*(N-2)*(N-3)... and so on, which is roughly half that required in the worst case for the Bubble sort. Even so, the time for this sort is still proportional to the square of the number of items. Sorting 10 random items requires roughly one second, 100 items roughly 80 seconds and 1000 items just over two hours. Adding one extra item and resorting takes half as long as a full random sort whereas sorting the list into reverse order takes 1.5 times.
This type of sort is commonly used by beginners and is presented in most introductory books on BASIC programming. It is easy to understand, easy to program in only three lines, and a time of roughly 80 seconds to sort 100 items is often quite acceptable where the sort is performed only rarely. We can do significantly better however.

Shell-Metzner

Named after its inventors, this algorithm adopts a more daring approach. It begins by comparing elements that are far apart on the assumption that swaps made over a bigger difference will be more efficient than swapping adjacent elements. This distance is called the gap. It starts out as N/2. Each time through the main loop this gap is cut in half eventually reaching 1. In other words on the last pass it performs like a simple Bubble sort. The Shell-Metzner sort is reasonably efficient but doesn't particularly like random data. It performs much better if most of the elements are grouped in some rough order. Sorting times are roughly proportional to N to the power 5/4 or $N^{1.25}$. To sort 10 random items requires roughly one second, 100 items roughly 26 seconds and 1000 items roughly 7.5 minutes. Adding one extra item takes roughly half as long as the full sort but, because of its efficiency, a resort into reverse order takes only 75% of the time for the random sort. A good general purpose sort for a reasonable number of items, but can be bettered, especially for random data.

Heapsort and Quicksort

Both these sorting algorithms are somewhat complicated in operation and unfortunately we cannot devote several pages to explain how they work. If you are really interested, take a stack of 5 or so numbers and work through the programs with pencil and paper. The overall idea behind these algorithms is not difficult to understand. The problem with a Bubble sort is that the sorting time increases in proportion to the square on the number of items. If we could find some method whereby the time increased in direct proportion, our task would be easier. Both Heapsort and Quicksort essentially start from the premise that it is quicker to sort, for example, 10 separate lists of 10 items each rather than one single list of 100 items. In this case sorting time would increase only by a factor of 10 compared with 100 as for a Bubble sort.
This is very similar to the way many people sort 'naturally'. Imagine you have a large stack of unsorted library cards and wish to sort them into alphabetical order. What you can do is toss them into piles labelled A-M and M-Z. The original problems has now been split into two smaller problems. The process can now be repeated splitting each of the smaller piles into two giving a total of four piles. Depending on the total number of cards in all, the process can be repeated a few more times. When each of the small piles is of a reasonable size you can use something like a Bubble sort tidy up. For example, if 1000 items are split into 16 piles of 62 each then the Bubble sort time has reduced from 1000 or 1000000, to 16*62 or roughly 62000. This is still fairly large, but you can see we are heading in the right direction.
Although neither Heapsort or Quicksort achieves this result, the improved mathematical efficiency of the Quicksort algorithm in particular makes significant gains, even over the Shell-Metzner sort.
I only mention Heapsort here for comparison. In fact, I can see little reason to chose it over the Shell-Metzner and particularly the Quicksort routines. Although roughly 10-20% faster than the Shell-Metzner for random data, where it behaves in a similar manner with sorting times being proportional to $N^{1.25}$, Heapsort takes roughly the same time to add an extra element, or to resort the list into the reverse order, as for the initial sort of random data. (If supplied with a list that is already in sorted order it actually takes 10% LONGER than for random data!)
Quicksort on the other hand is significantly the fastest algorithm for random data and the sorting time is proportional to N*LOG(N). It will sort 10 items in roughly 1 second, 100 items in 17 seconds and 1000 items in just over 4 minutes. Adding an extra item or resorting a list into reverse order take roughly 70% as long as the full random sort. With random data, the more items in the list, the greater are the gains over the Shell-Metzner routine. For less than 50 items there is no significant difference but above this the Quicksort is appreciably faster. However it is worth noting that the Shell-Metzner is always faster at adding one extra item.

Adding Extra Items to a Sorted List - Insert Sorting.

Once the main list has been sorted into the required order, (and probably saved to disk?) how should you add extra items?
If you wish to add more than 25 extra items, add the new ones to the end of the list and re-sort with Shell-Metzner or Quicksort.
More usually, however, the problem is to add just one or two extra items to a sorted alphabetical list of names, account numbers or the like, or, to enter items one by one from the keyboard and build them up into a sorted list. It is definitely very inefficient and time consuming to simply add the new item to the end of the list and re-sort by any method. A far simpler method is to take each new item as it is entered, search the list to find where the new entry should go, move all the remainder down one place, and insert the new item. For example, assume we have 1000 items in the list and wish to add one more. If added to the end of the list and resorted, the approximate sorting times are Bubble, one hour, Shell-Metzner, 3 minutes, and Quicksort, 4 minutes. Obviously a waste of time! An insert routine on the other hand would take only 4 seconds!

I have included two insert sorting routines. The first uses a FOR/NEXT loop to search the list to find where the new item should go. The second uses a technique called a binary or dissection search. This is theoretically more efficient but because Commodore computers can execute FOR/NEXT loops faster than IFs and GOTOs, it is only useful to bother with this extra coding if the list contains 50 or more items. With both routines, once the correct position is found, a FOR/NEXT loop is used to shuffle all items down one place to open up a place for the new item.
The binary search routine works by dividing the list in half and checking whether the new item is less than the centre item. If so, the left half part is again split in half and the centre item compared again. This process continues until the correct place is found. It is very efficient, taking only 4 divisions and comparisons for 10 items, 7 for 100 items, 10 for 1000 items and only 13 for 10000 items. As can be seen, the efficiency over a FOR/NEXT search increases dramatically as the number of items increase.

Such a binary search method can also be used for direct searching of a list to check whether a given item already exists. For example, when entering or reviewing names on a User Group membership list, a quick check can be made to find if the name already exists.
There are a number of other techniques for sorting special types of data. For example, disk sorting when the number of items is too large to fit into the computer, sorting using linked lists, and sorting key indices to data records, to name just three. Unfortunately, we will have to leave these to a later date. Garbage collection routines on the C64/VIC often causes havoc and slows sorting of large string arrays to a snail's pace. The easiest way to overcome this is to compile the program in Petspeed.
Just to prove that I haven't made all this up, here are two tables showing the time taken to sort lists of various lengths using the programs provided below. Times are in jiffies or 1/60ths of a second.

Approximate Sorting Times on C64 for 1-Dimensional Array
        of Random Numbers in BASIC on C64

Sort Type	Time in Jiffies			
No Items	10	20	50	100
Bubble	50 (41)	252 (129)	1548 (695)	6337 (2432)
Exchange	43 (32)	192 (116)	1183 (873)	4761 (2621)
Shell-Metzner	43 (48)	154 (113)	604 (340)	1616 (791)
Heap	65 (67)	168 (172)	556 (674)	1262 (1413)
Quicksort**	56 (63)	158 (139)	511 (404)	1065 (934)
Simple Insert	(7)	(11)	(23)	(47)
Binary Insert	(12)	(18)	(27)	(38)

(Numbers in brackets are the times taken to re-sort the list after one extra element (0.5) has been added to the end of the list.)

Same program compiled in Petspeed on C64

Sort Type	Time in Jiffies			
No Items	50	100	500	1000
Bubble	166	737 (368)	18665 (8941)	73649 (35429)
Exchange	138	632 (426)	15959 (10525)	63326 (41933)
Shell-Metzner	64	166 (85)	1188 (557)	2742 (1251)
Heap	63	147 (157)	1006 (1060)	2251 (2363)
Quicksort**	56	125 (91)	750 (600)	1650 (1300)
Simple Insert		(6)	(30)	(60)
Binary Insert		(6)	(20)	(38)

Greg Perry

NOTES & GOSSIP

Repair Survey

The response to last month's repair survey has been absolutely dismal. Does this mean that members have no problems or praise for repair services? We find this hard to believe because in conversation with members we have heard plenty of harrowing stories regarding shoddy repairs. Were those all tales without substance? If so, don't spread them around. You do a dis-service to some very reputable repairers. If you do have some real gripes though, fill in last month's form and return it to the secretary. If none are forthcoming your committee will take no further action in this matter.

Visitor

It was nice to see our old friend and member Carl Holzberger turning up during our May meeting. Carl, formerly a Brisbane member, but now living in Broken Hill, made use of the NSW school holidays to pay us a visit. That old mining town seems to agree with Carl, and it was good to hear that he is playing a prominent part in the Broken Hill computer scene.

The Well-Travelled Secretary

Our secretary Norm Chambers, in the course of his profession does a fair bit of travelling. Instead of sitting in a lonely motel room he visits Commodore User Groups. During the last month he has visited Toowoomba, Townsville and Cairns. In Toowoomba Norm was present during the foundation meeting of the new Commodore Users Group, and in Townsville and Cairns he re-established contacts made on an earlier visit. Great work Norm!

Attention Affiliated User Groups:

The committee has decided to make our newsletter "CURSOR" available to other user groups at attractive rates. Copies would be packaged in bulk and posted directly to the user group in question, for distribution to their own members.

Skaie Disk Drive

Good reports have reached us about this new Japanese disk drive, which is 1541 compatible. Originally supplied with an external power supply, new stocks are coming through with an inbuilt power supply.
We hope to obtain one of these units in the near future for review.

C-16 and Plus/4 Computers

Our group is not particularly rushed with new members who own either of the above computers. We know of a few C-16 owners, but the Plus/4 owners are staying away in droves - or do they?
If you are the owner of one of these computers, how about telling us about your experiences with this computer?

Speakers Required

We urgently require the services of members who are using their computers in new and interesting ways. Do you use your computer in some unusual capacity associated with your work or other hobbies? If so, come forward and share your interests with your fellow members. Don't automatically assume that nobody is interested in your particular field of application. You may be pleasantly surprised!

Public Domain Software

Expect a considerable increase in our range of Public Domain Software in the near future! We will keep you informed.

CPM

With the pending arrival of the C-128, CPM software will become a familiar concept amongst Commodore users. There are literally hundreds of CPM programs in circulation, such as DBase II, Wordstar etc. There is also a huge range of Public Domain Software in the CPM format. In the near future we hope to have a guest speaker from a local CPM users group who can give us an introductory lecture on CPM software and hardware.

DIRECTORY

MANAGEMENT COMMITTEE

President:	GREG PERRY	Ph. 36 3295
Secretary:	NORM CHAMBERS	Ph. 341 5651
Treasurer:	LESTER BENNETT	Ph. 200 1243

SECONDED TO MANAGEMENT COMMITTEE

Technical Co-ordinator:	ROGER HAIGH	Ph. 399 6037
Sub-Group Co-ordinator:	TERRY STEER	Ph. 200 5926
Chief Librarian:	CLIFF POTTINGER	Ph. 277 4520
Newsletter Editor:	RALPH DE VRIES	Ph. 30 3477

COMMITTEE MEMBERS

Assistant Treasurer:	Position Vacant	
Vic-20 Co-ordinator	BARRY WILSON	Ph. 399 6204
Education Officer:	DEREK FARRELL	Ph. 359 8559
Sysop:	RAY KING	Ph. 208 1409
BBS:	--- ----	Ph. 808 2125
Librarian - Books:	CLIFF YULE	Ph. 356 7571
Librarians - Disks:	BILL BOHLEN	Ph. 208 3729
	MAX BEAN	Ph. 208 1225
Librarian - VIC:	JOHN JOHNSTON	Ph. 30 5240

Please enclose a stamped self-addressed envelope, when contacting committee members by mail.

Please address all editorial matter to: P.O. Box 384, Ashgrove, QLD, 4060. (NOT to P.O. Box 274, Springwood please!)
Deadline for any particular month is the second Tuesday of that month.

---

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx