CSI2131 - File Management
2001 Midterm Test
Instructor: Lucia Moura
Solution

1 File processing in C++

Answer key = A

`f << c;` would result in error because the fstream `f` was opened for input only.

2 Records and Fields

Answer key = E

A, B, C and D are clearly correct.

E is incorrect because RRN can only be used to calculate the byte offset for **Fixed length Records**.

3 Disks

Answer key = C

Disk 1

Records per sector = 2
Sectors per track = 25,600/128 = 200
Records per track = 2 × 200 = 400
Records per cylinder = 4 × 400 = 1,600
Cylinders = 16,000/1,600 = **10**

Disk 2

Records per block = 50
Bytes per block = 50 × 50 + 60 = 2,560
Blocks per track = 25,600/2,560 = 10
Blocks per cylinder = 4 × 10 = 40
Records per cylinder = 40 × 50 = 2,000
Cylinders = 16,000/2,000 = **8**

Disk 3

Records per block = 10
Bytes per block = 10 × 50 + 300 = 800
Blocks per track = 25,600/800 = 32
Blocks per cylinder = 4 × 32 = 128
Records per cylinder = 128 × 10 = 1,280
Cylinders = 16,000/1,280 = **12.5**

4 Tape

Answer key = B

Block size in bytes = 10 × 30 = 300

Block length in inches:
- 3,000 bytes in one inch (density) means that 300 bytes accupies 0,1 in.
- Block length = 0.1(data) + 0.2(gap) = 0.3 in.

Calculating **Effective Density (bpi)**
300 bytes end up taking 0.3 in
⇒ **Effective Density** = 1,000 bpi

Effective Transmission Rate
= Effective Density (bpi) × Speed (ips)
= 1,000 bpi × 200 ips = 200,000 bps = **<u>200</u>** KB/sec.

5 CD-ROM

Answer key = C

A and B are ruled out because seeking and transfer rates are **<u>slow</u>** rather than fast.

D is ruled out because CLV is indeed a weakness (it causes slow seek performance).

E is ruled out because CD-ROMs are read-only.

C is the correct one: the storage capacity of a CD-ROM is large relatively to its cost and size.

6 Basic I/O Software and Hardware

Answer key = D

A, B, C and E are clearly correct statements.

D is incorrect: Buffers are not only used when the disk is not available for writing; They are also (and mainly) used for accumulating lots of bytes before going to the disk (to reduce the number of disk accesses).

7 Buffering Strategies

Answer key = A

A is clearly **correct**.

What is **wrong** with the others ?

B. least recently used is advisable rather than most recently used.

C. double buffering actually reduces the time for programs doing read and write (alternating).

D. buffering makes I/O operations more efficient specially when I/O-CPU overlapping **is** possible.

E. this is wrong. Indeed, closing an output file is important because E is false.

8 Lempel-Ziv Compression

Answer key = C

```
1 2 3  4 5  6  7
a|b|ab|c|aa|bc|abc
```

• Original message takes $8 \times 12 = 96$ bits

• Computing number of bits for encoded message:
- bits for letters = 7 (indexes) $\times 8 = 56$
- bits for numbers =
0(index1) + 1(index2) + 2(index3) + 2(index4) + 3(index5) + 3(index6) + 3(index7) = 14

Total bits for encoded message = 56 + 14 = 70 bits

Saving: 96 - 70 = **26** bits.

9 Huffman Compression

Answer key = B

1) Use greedy method to build tree (each step combine two trees with smaller weight)

Building Tree:

```
Subtrees:   (E,11)  (D,14)  (C,15)  (B,20)  (A,40)


Subtrees:   (C,15)  (B,20)  (25)  (A,40)
                            /  \
                       (E,11)  (D,14)


Subtrees:    (25)            (35)       (A,40)
             /  \            /  \
        (E,11)  (D,14)  (C,15)  (B,20)


Subtrees:   (A,40)         (60)
                         /      \
                       /          \
                    (25)          (35)
                    /  \          /  \
               (E,11)  (D,14)(C,15)  (B,20)


Final Tree:        (100)
                 /       \
           (A,40)         (60)
                        /      \
                      /          \
                   (25)          (35)
                   /  \          /  \
              (E,11)  (D,14)(C,15)  (B,20)
```

2) Use tree to find codes:

A:0 - B:111 - C:110 - D:101 - E:100

10 Data Compression

Answer key = D

D is correct.

A is clearly wrong.

B is wrong since smaller size **decreases** transmission time.

C is wrong because run-length encoding is most effective for image compression.

E is wrong since the Lempel-Ziv encoding algorithm makes no use of the symbols' probabilities.

11 Reclaiming Space

Answer key = D

1. record R3 is deleted

21 goes to avail list
- Avail list = 21

2. record R5 of size 25 is added

added to the end of file (25 cannot use 21)
- Avail list = 21

3. record R4 is deleted

31 goes to avail list
- Avail list = 31,21

4. record R6 of size 17 is added

worst fit chooses 31 and places leftover (31 - 17 = 14) back to list
- Avail list = **21,14**

OBS.: The mention of **External Fragmentation** being reduced by **Worst Fit** implies that we are placing "leftovers" back into list.

12 Fragmentation

Answer key = E

Statements A, B, C and D are clearly correct.

E is wrong because **Variable-Length** records make external fragmentation worse than fixed-length (fixed-length records can be easily re-used, for variable-length we studied strategies to try to reduce external fragmentation e.g. worst fit)

13 Sorting and Searching

Answer key = A

A is clearly correct (keysorting only store keys).

Why are the others wrong ??

B is wrong: when a file does not fit into main memory, we are not doing internal sorting.

C in wrong: binary search takes less time ! while "reading a file sequentially is faster than seeking through the file" is true, the number of seeks is $\log_2 n$ rather than n in the worst case.

D worst case in $\log_2 n$.

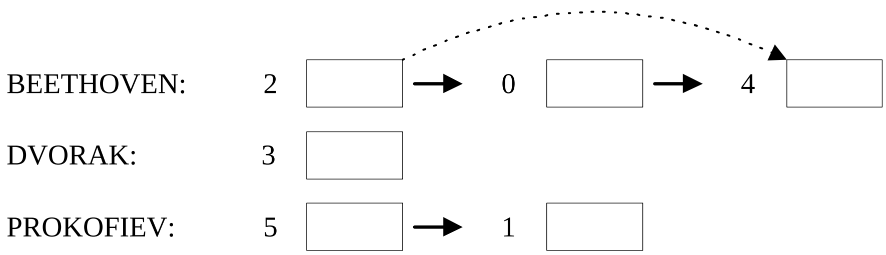
E internal sorting requires $2n$ accesses to secondary storage. An n grows $2n \ll n \log_2 n$.

14 Indexing

Answer key = D

You had to recall how **Inverted Lists** are organized.

References are showing linked lists.



Removing the data record above requires the deletion of record 0 from the list of BEETHOVEN.

The changes are:

0	* * * * *	*
1		
2		4
3		
4		
5		

15 Co-sequential Processing

Answer key = A

List 1	List 2	Reads	Comparisons
B	A	2	1
E	B	1	1
F	F	2	1
K	G	1	1
L		2	1
P	Attempt to read detects EoF: match is abandoned.	**8**	**5**
Z			