(Total time 120 minutes, Total Points = 100 points)
from 1:00 – 3:00 pm

Name: (please print) Nick Land

This examination is an open book/notes/homework exam. However, you may not collaborate in any manner in the exam and you can't use Internet. In recognition of and in the sprit of the Oakland University Honor Code, I certify that I will neither give nor receive aid during the exam.

Signature: Nick D Land

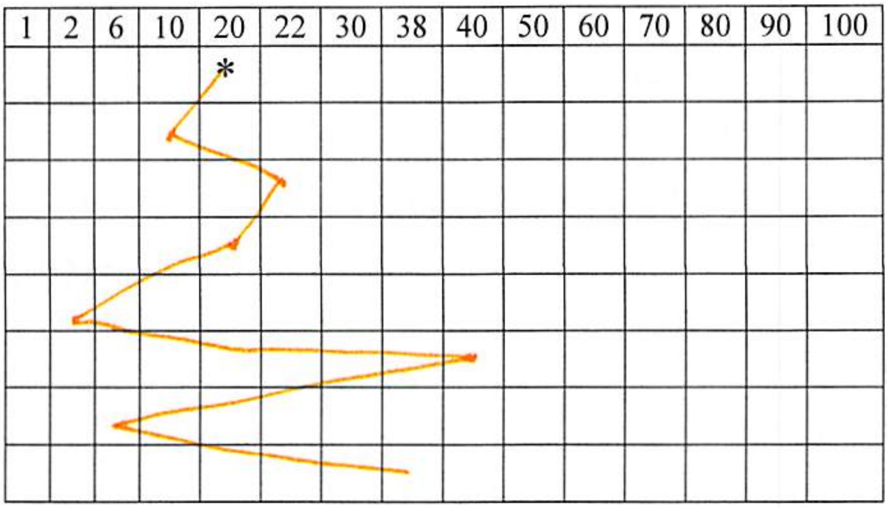
Hints:

1. Put your name on the exam books NOW!
2. Read the questions clearly and think it through before you are answering.
3. You have 120 minutes to complete the exam. Be a smart exam taker! Remember not all the points are born equal. So, if you get stuck on one problem go on to another problem.
4. Please write your answers **clearly** for correct grading.

First Attempt = Apache Sunset

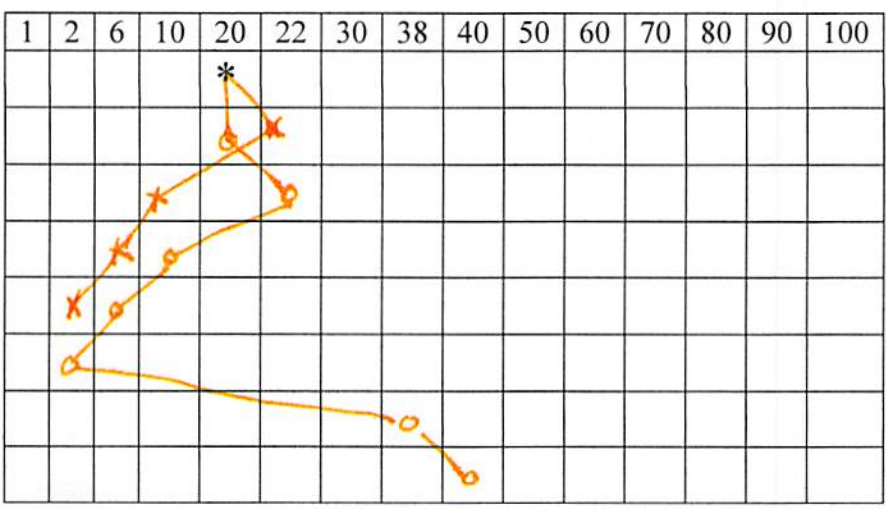
Question 1. (20 points) **[Disk Scheduling]** Disk requests come into the disk driver for cylinders 10, 22, 20, 2, 40, 6, and 38 in that order. Assume that the disk has 100 cylinders, from 1 to 100. A seek takes 6msec per cylinder moved (e.g., the disk arm move from third to fourth cylinder) and we assume the linear relationship between the moving distance and seeking time. In all the cases, the arm is initially at cylinder 20. Please show in each diagram the moving of the arm and compute the total seek time for the request sequence given above for the following scheduling schemes.

a) First-come, First-served (FCFS)



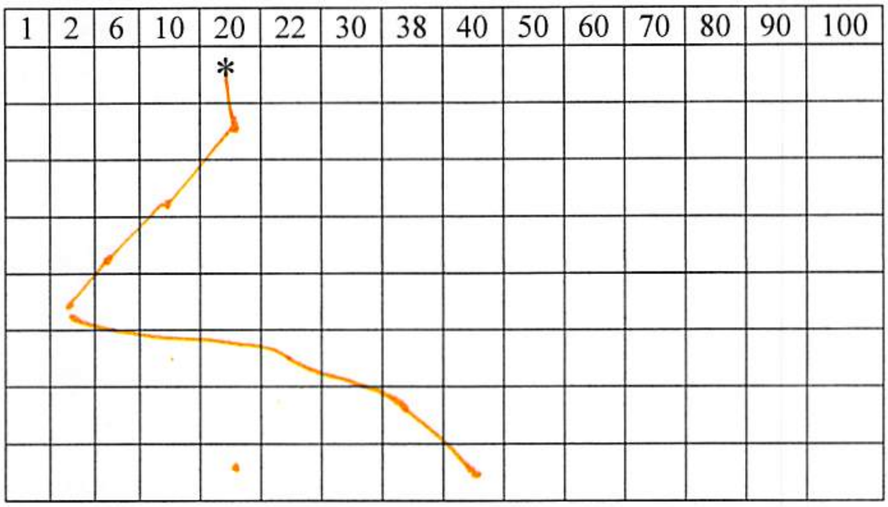
Total seek time for FCFS is 876 MS 146 total cylinders moved

b) Shortest Seek Time First (SSTF)



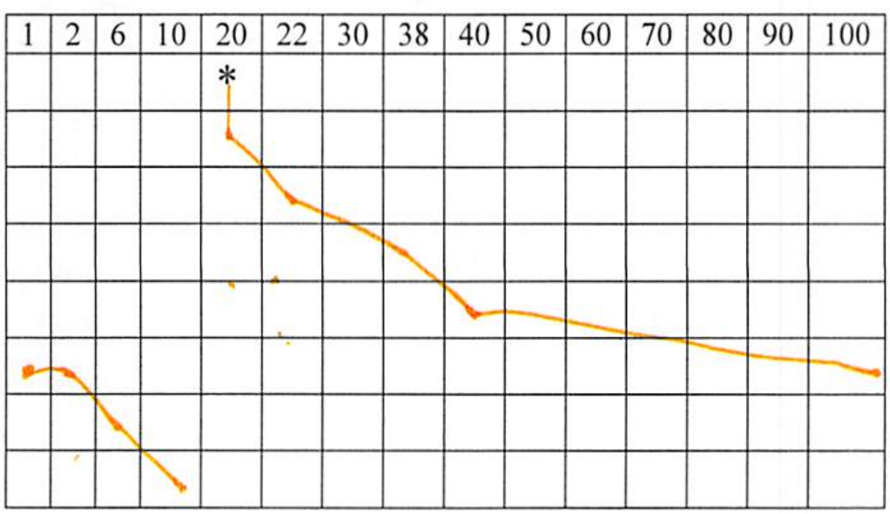
Total seek time for SSTF is 348 MS 58 total cylinders moved

c) SCAN (with the disk-arm initially moving towards higher number cylinders from lower number cylinders)



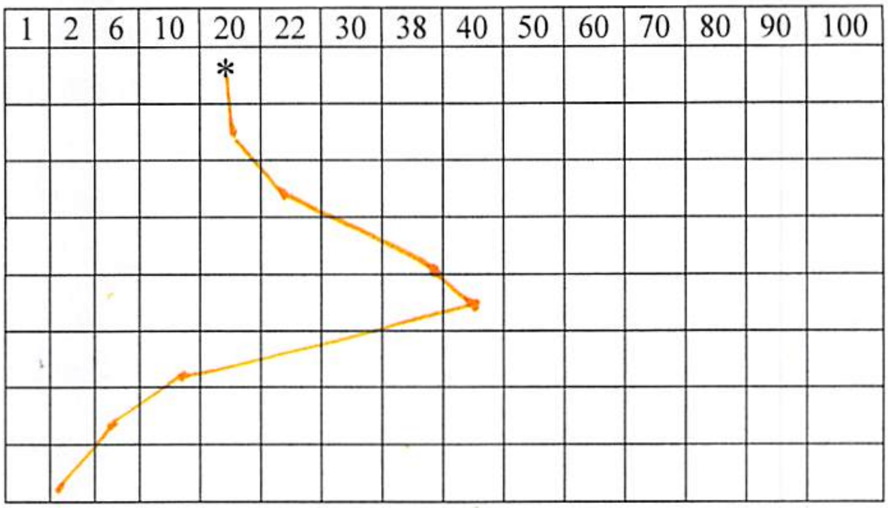
Total seek time for SCAN is 324 MS, 54 total cylinders moved

d) C-SCAN



Total seek time for C-SCAN is 540 MS, 90 cylinders msec.

e) C-LOOK



Total seek time for C-LOOK is 336 MS, 56 cylinders msec.

Question 2. [15 points] **The following table shows the core map of a virtual memory system, which has a page size of 100.**

Frame number	Process ID	Page number
0	1	2
1	1	1
2	2	1
3	3	0
4	1	3

a. To which physical address does virtual address 130 of process 1 map? If this virtual address does not map to any physical address, write "does not map".

logical = $(1 \times 100) + 30$

Physical = Frame # = 1

$(1 \times 100) + 30$ = Physical address of $\boxed{130}$

b. To which physical address does virtual address 17 of process 2 map? If this virtual address does not map to any physical address, write "does not map".

virtual address 17 $\boxed{\text{does not map}}$

c. Which virtual address of which process maps to physical address 50?

Physical address 50 = $(\underset{\text{frame}}{0} \times 100) + 50$

So logical is $(\underset{\text{page \#}}{2} \times 100) + 50 \Rightarrow \boxed{250}$

Question 3. [10 points] A computer system whose processes have 1024 pages in their address spaces and page tables are stored in main memory. The overhead required for reading a word from the page table is 500 nsec. To reduce this overhead, the computer has an associative memory, which holds 32 (virtual page, physical page frame) pairs, and can do a lookup in 100 nsec. What hit rate is needed to reduce the mean overhead to 200 nsec? **[Comments: an equation needs to be formulated first then plug in the numbers for your solution]**.

Equation: $100h + 500(1-h)$

$= 100h + 500 - 500h = 200$

$= -400h + 500 = 200$

$= -400h = -300$

$= h = \frac{-300}{-400}$

$h = \frac{3}{4} = 0.75\%$ hit rate

Question 4. [15 points] A virtual memory system supports 12 bits length virtual addresses. It uses pure segmentation with a maximum segment size of 1024 (2^{10}) bytes. Suppose that the segment table for the currently running process looks as follows:

Seg.number	V	P	Start	Len
0	1	1	0	200
1	1	0	1000	160
2	0	0	0	0
3	1	0	200	300

In the table above, V and P are the segment "valid" and "protection" bits, "Start" represents the physical address of the start of the segment and "Len" is the segment's length. P = 1 indicates that a segment is read-only.

Consider the following read and write operations. If the specified operation would succeed given the segment table shown above, give the physical address to which the specified virtual address would translate. If it would not succeed, state the reason that it would fail.

a. A write to virtual address 150.

b. A read from virtual address 1025.

c. A write to virtual address 4000.

Question 5. [15 points] A process is allocated 4 frames, initially empty. For each of the following page replacement algorithms, determine which pages are in memory after the last page is referenced and give the total number of page faults. The tables are given for you to work out your solutions. However, we will only grade the answers in the final column and your answer for the total number of page faults.

(i) FIFO

Whoops. Its 4 frames, not 3. Pay attention

Referenced Page	1	2	1	2	40	39	38	37	2	4	5	38	1
Frame 1	1	1	1		40	40	40	37	37	37	5	5	5
Frame 2		2	2		2	39	39	39	2	2	2	38	38
Frame 3			1		1	1	38	38	38	4	4	4	1
Frame 4													
Fault?													

Total number of page faults = _____

(ii) Optimal – fewest possible page faults (break ties with lower page number)

Referenced Page	1	2	1	2	40	39	38	37	2	4	5	38	1
Frame 1													
Frame 2													
Frame 3													
Frame 4													
Fault?													

Total number of page faults = _____

(iii) LRU

Referenced Page	1	2	1	2	40	39	38	37	2	4	5	38	1
Frame 1													
Frame 2													
Frame 3													
Frame 4													
Fault?													

Total number of page faults = _____

Question 6. [25 points] Consider a File system that maintains unique index node for each file in the system. Each index node includes **10** direct pointers, a single indirect pointer, and a double indirect pointer. The file system block size is 1024 bytes, and a block pointer occupies 4 bytes.

a) (7 points) What is the maximum file size that can be supported by the index node?

$1024 * (10 + 2^8 + 2^8 * 2^8)$

$= 2^{10}(10 + 2^8 + 2^{16}) \Rightarrow 2^{10}(2^3 + 2 + 2^8 + 2^{16})$

$\cancel{10 + 2^{18} + 2^{26}}$

$\boxed{\cancel{= 2^3 + 2 + 2^{18} + 2^{26}}}$

direct pointers

$\boxed{2^{13} + 2^{11} + 2^{18} + 2^{26}} \rightarrow 67381248$

indirect pointer

b) (8 points) How many disk operations will be required if a process read data from the N^{th} block of a file? Assume that the file is already open, the buffer cache is empty, and each disk operation read a single file block. Your answer should be given in terms of N.

$0 \leq N < 8$, one operation

$8 \leq N < 256 + 8$, two operations

$256 + 8 \leq N \leq 2^{13} + 2^{11} + 2^{18} + 2^{16}$, three operations

c) (10 points) Suppose the above file system is modified so that the index node does not contain a double indirect pointer. Instead, if the file size is larger than can be represented by the direct pointers and single indirect pointers, the last pointer in the indirect block will be used to point to another indirect block containing more pointers to data blocks. If that second indirect block fills up, its last pointer is used to point to another indirect block. This chain of indirect blocks can grow as long as necessary to accommodate a large file. Under this new indexing scheme, how many disk operations will the file system have to perform if a process reads data from the N^{th} block of a file? Use the same assumptions were used in the previous question.