1705005

19-2-21

CSE-314

Quiz

Ans. to 1(a)

2^5 2^{20}

m →

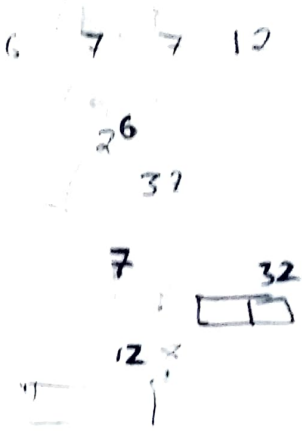
n →

p →

k → $2^6 - 1$

j → $2^7 - 1$

i → $2^7 - 1$



(b)

E 79 EAA 5B

0x $\underbrace{0101}_{5}$ $\underbrace{1001}_{9}$ $\underbrace{0011}_{83}$ $\underbrace{1000}_{8}$ $\underbrace{0010}_{2}$ 453

pde → pgdir[5] → &(E79EAA5B + 5)

Ans. to 2

Part B

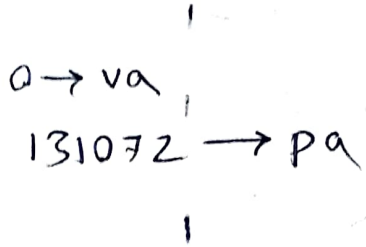
1. d)

2. c)

3. c)

4. b)

5.



6. c

7. c

8. d)

~~4096~~

2nd page

1+1+1

8*2+1

8191

Ans. to 2

```
if find -name *.{$1} -type f | while read val; do
    cp $val .*{$2}
    filename=$(basename -- "$val")
    cp ${filename}.$val} filename.*{$2}
    rm $val
done
```

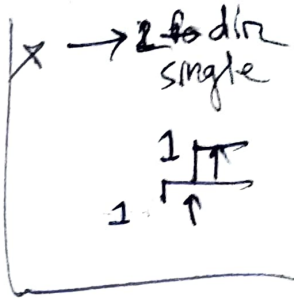
Ans. to 3

a)
```
count=0
while read line; do
    $count=$((count+1))
    if (($count >= 15)); then
        if ((count <= 22)); then
            echo $line
        fi
    fi
done < 'input.txt'
```

b)
```
touch grep.txt
history | grep "grep" | while read usage; do
    usage >> grep.txt
done
```

Ans to 4

(a)



This implementation fails because it allows multiple person to be in the same step in the same direction. It doesn't achieve the target of constraining one person at a single step in one direction

(b)

sem_up [X]
sem_down [X]

} 2X semaphores, each binary

here at each step i of x sem_up [i] will fulfil the constraint of only one person moving upward. Same goes for any i in sem_down [x]

Ans. to 5

1) allocuvm → ~~wt~~ i) a new page is allocated
   → ~~if RAM has room~~
   → if total page <= MAX_PSYCH PAGES
      → add to TLB
   → else
      → swap pages
      → remove the page that has been swapped from TLB if it exists there
      → add new page to TLB

2) deallocuvm → ii) a page is deallocated
   → if page is in RAM
      → remove from TLB if it exists

3) trap 14 → iii) page fault
   → find least imp. RAM page
      → remove from TLB
      → write to swap
   → read from swap
      → add to TLB

add to TLB: if TLB has no room for new page entry
   → remove oldest entry
   → update flags
   → add

Ans. to 6

No, if interrupts are not enabled, a process might run for forever or get stuck b. When enabled, a process waiting for an interrupt will release the CPU scheduler & ~~gives~~ let ~~s~~ other process run. ~~otherwise the~~ when ~~the interrupt is desire~~

Ans. to 7

~~a.~~ It may go to sleep even if ~~a~~ there is a child process which is already in the state ZOMBIE. ~~This will work~~