$\int 204 P$

$\Rightarrow$ page offset) $= 11$ bits $\rightarrow$

32 pages $\Rightarrow$ 5 ~~pag~~ bits $\rightarrow$
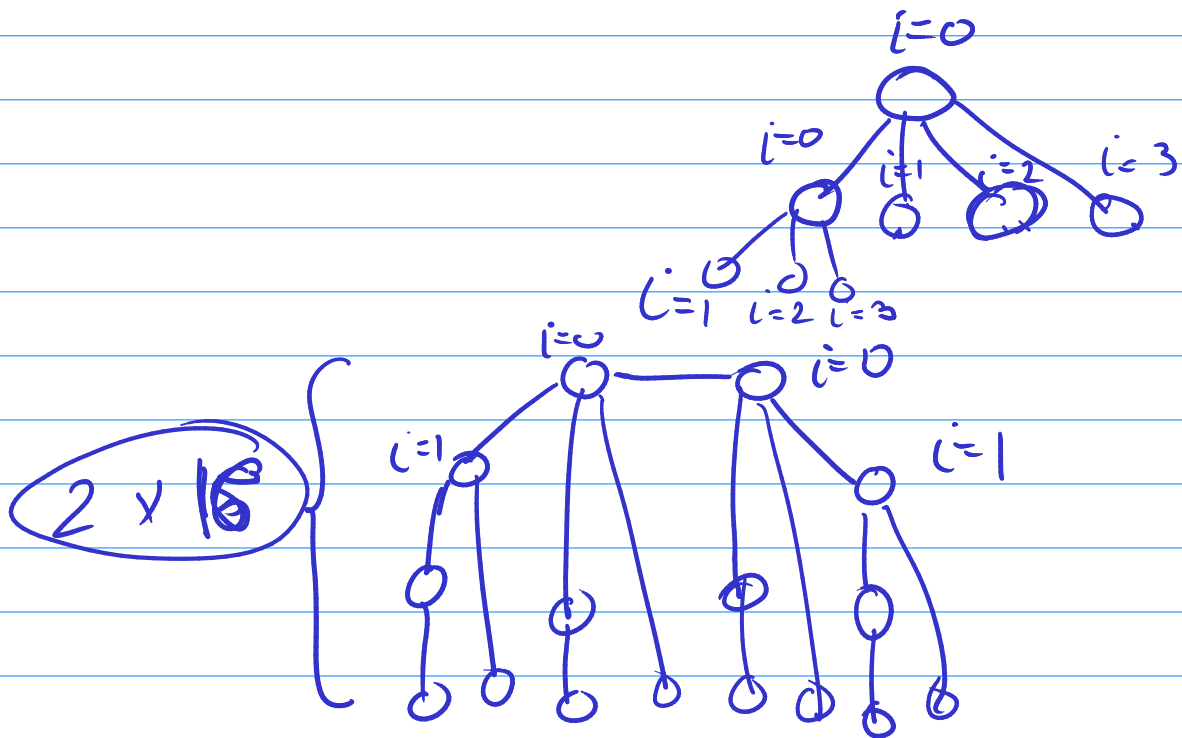
$i=0$

$i=0$ $i=1$ $i=2$ $i=3$

$i=1$ $i=2$ $i=3$

$i=0$ $i=0$

$\boxed{2 \times 16}$ $i=1$ $i=1$

Hello $\times 32$

$32 \times 4 = 128$

$(0.5) 6 + (0.5) 10$
$= 8$

$t_0 = 6$
$t_1 = 8$ 8 8
$t_2 = 10$ 8
$t_3 = 4$ 4 9^{10} $\widehat{6.5}^{8}$

R2 , R1 , R3 , R4

P $\to$ P2

bash: /home/borsha/Documents/a.out : No such file or directory
127
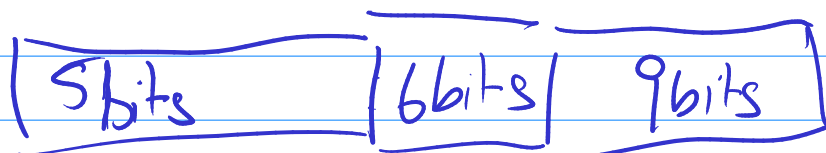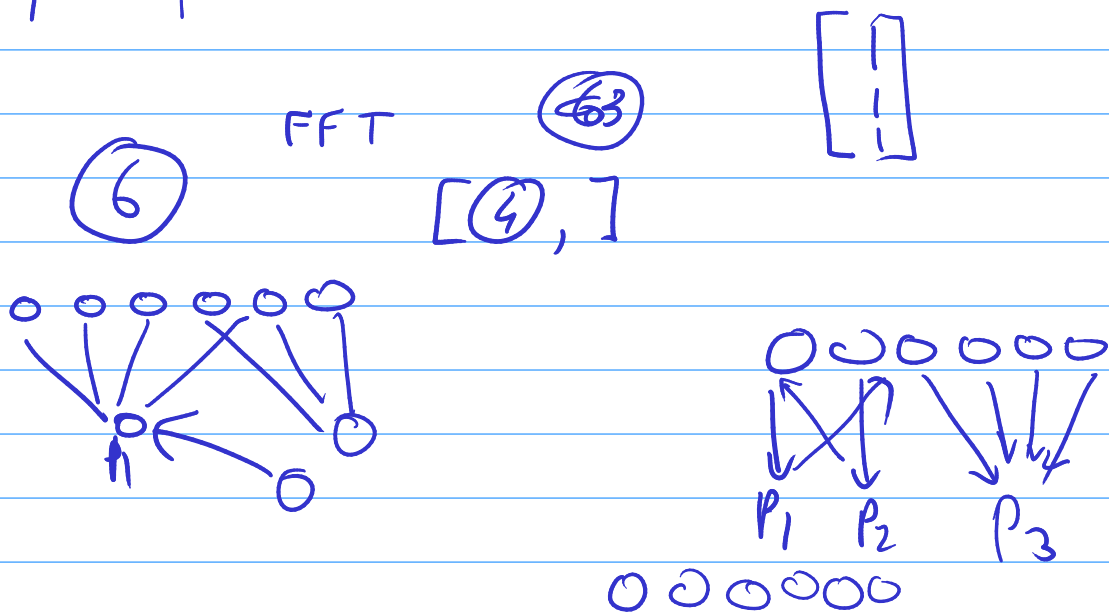
(n) Work = m   R m   5 ^{6} (13)   6

finish = F F F ·

P1	3	1
P2	4	2
P3	6	6

(6) 5

work = 6

FFT   (63)   [ ]

(6)   [(4),]

5 bits | 6 bits | 9 bits

$$\frac{500 \times 150}{2500}$$

80

400   500   (30)

1750   850   (81)

1800   400   (24) $$\frac{4 \cdot \times 150}{25}$$

2500   700   (42)

       50   (3)

150 B

(1) [250]   [250] [300]

     ~~350~~   [350] [200]
                      [100] ———→ 100 hole
     [150]

     [400]

     [350]

     [250]   [600] 200

     ~~350~~   [   ] 350

     [150]

     [400]

     [350] (50)

P=150

300

200

250

300

→ 300 is freed

150

→ 50 is left   vs

250

300

} 350
   is freed

150

250

300

Lets say new process of 350kB enters
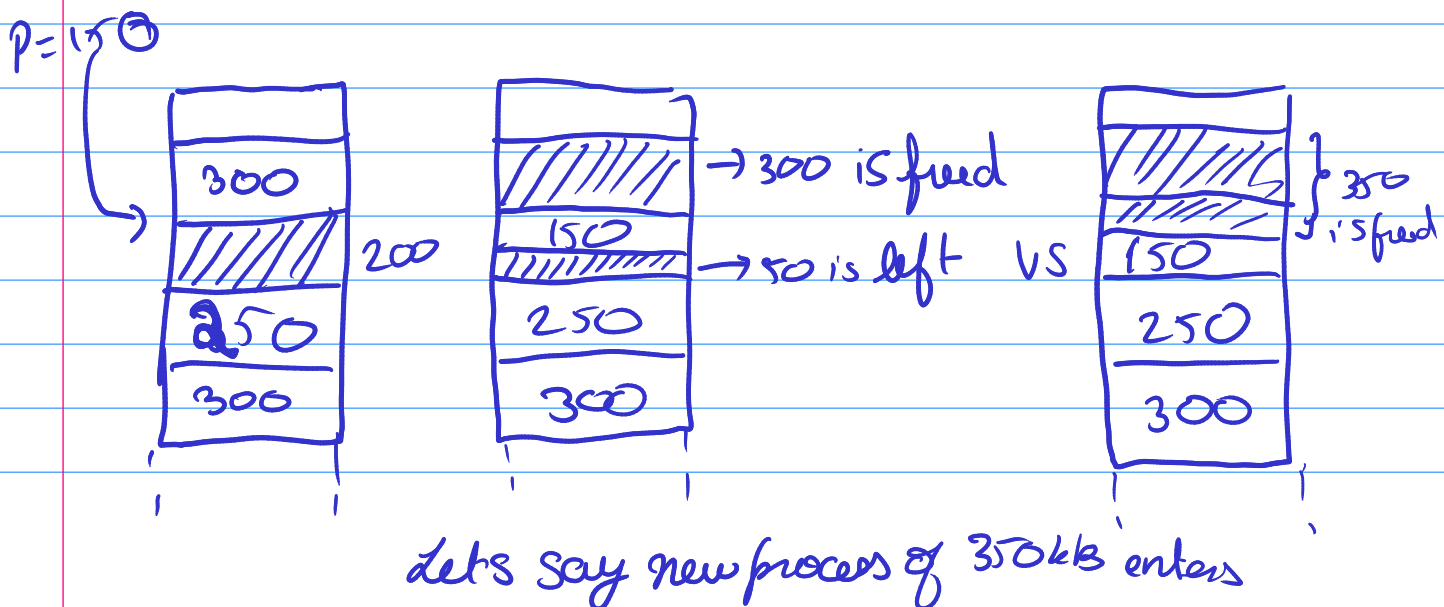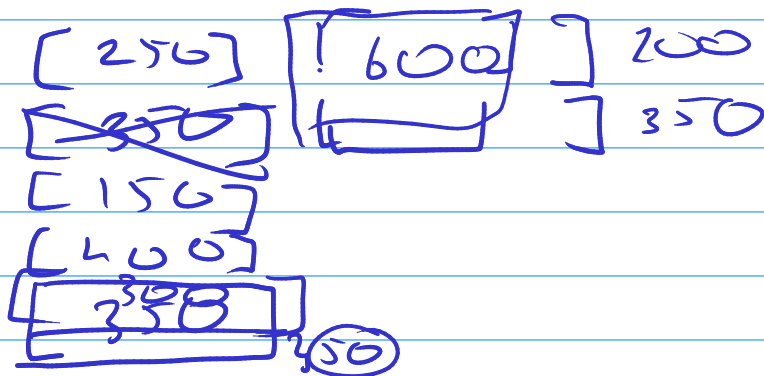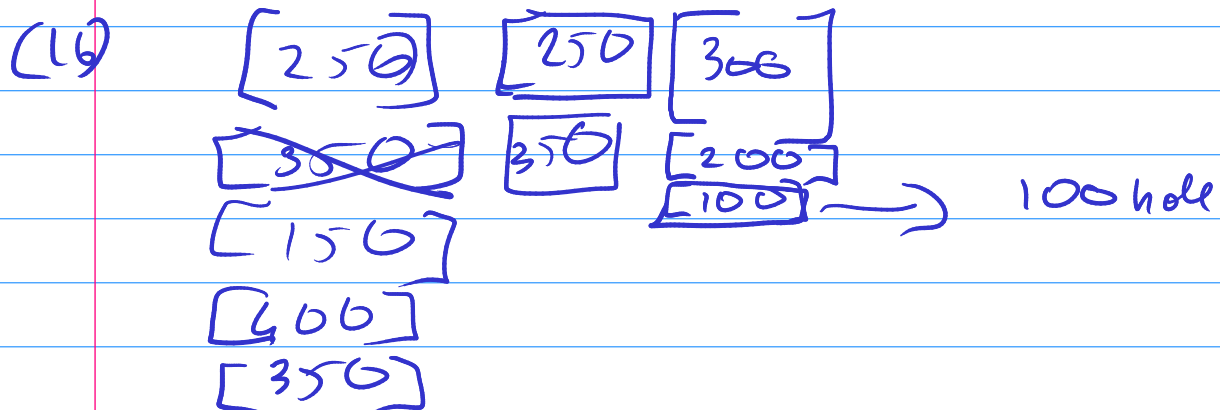
```c
#include<unistd.h>
#include<errno.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<pthread.h>
#include<sys/wait.h>
void* thread_func(void* a){
        int flag = *(int*)a, i = 1;
        if(flag){
                i+=5;
        }
        for(int count = 0; count < 5; i++,count++){
                printf("%d ",i);
                sleep(1);
        }
}
int main(void){
        int pfds[2];
        char buf[30];
        if(pipe(pfds) == -1)
        {
                perror("No u");
        }
        if(fork())
        {
                close(pfds[0]);
                strcpy(buf,"Please no u");
                write(pfds[1],buf,30);
                wait(NULL);
                if(fork() == 0){
                        pthread_t t1,t2;
                        int flag = 0,flag2=1;
                        pthread_create(&t1,NULL,thread_func,(void*)&flag);
                        pthread_create(&t2,NULL,thread_func,(void*)&flag2);
                        pthread_join(t1,NULL);
                        pthread_join(t2,NULL);
                }
                else{
                        wait(NULL);
                        printf("Parent process exiting\n");
                        exit(0);
                }

        }
        else{
                close(pfds[1]);
                read(pfds[0],buf,30);
                printf("%s\n",buf);
        }
}
```
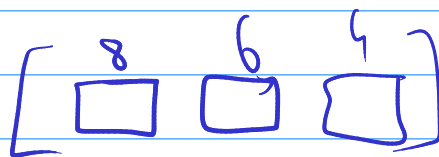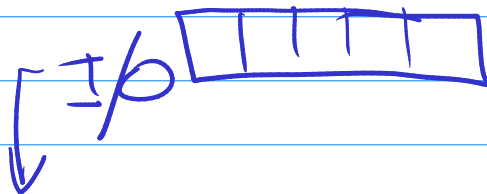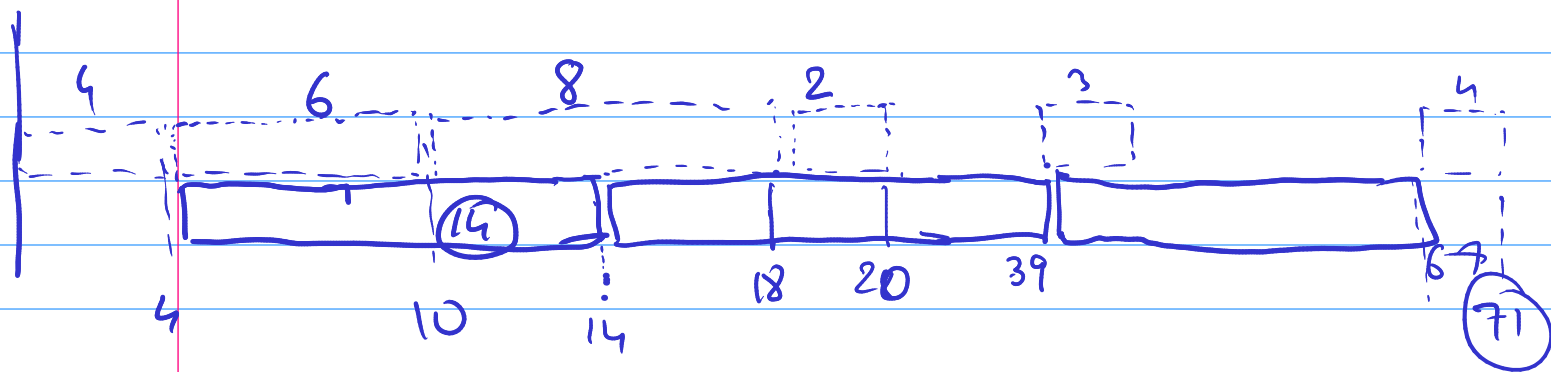
{DMA gotta revise

First 3 Page Faults then 6

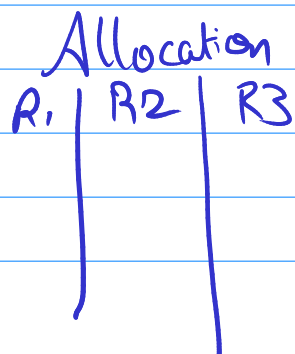P₁   20   20% IO 70% CPU   10% IO
P₂   30
P₃   40

2)

[I/O ⎡⎢⎣ | | | | | ⎤⎥⎦

[8   6   4
  ▢   ▢   ▢]

                          39 + 28

4   6 ---   8 ----   2 ---   3 ----   4
|⎡   ⎤
|⎣ [14 | | 18 20 39 | 64 ⎦ 71]
4   10   14

≈ 11.26% CPU IDLE

Allocation
R1 | R2 | R3

2.d)   work: 2 1 3

Need:
$$\begin{bmatrix} 1 & 0 & 3 \\ 3 & 3 & 4 \\ 1 & 3 & 0 \\ 3 & 6 & 3 \end{bmatrix}$$

4 33

Fin : [F F F F]

Process:	Allocation		
	R1	R2	R3
P1			
P2			
P3			
P4			