

NAME → ASHISH RAWAT

CAMPUS → Dehradun

UNIV ROLL → 2023040

Sub → O.C.

STID → 20051025

Practical

```

Ans 1) int frag[10], blocks[10], process[10];
        int m, n, block-number, process-number,
        temp, top = 0;
        static int block-arr[10], process-arr[10];
        printf("\nEnter no of blocks: \t");
        scanf("%d", &block-number);
        printf("\nEnter total No of processes: \t");
        scanf("%d", &process-number);
        printf("\nEnter the size of the blocks: \n");
        for (m = 0; m < block-number; m++)
        {
            printf("Block No. [%d]: \t", m+1);
            scanf("%d", &blocks[m]);
            printf("\nEnter the size of the processes: \n");
            for (n = 0; n < process-number; n++)
            {
                printf("Process No [%d]: \t", n+1);
                scanf("%d", &processes[n]);
            }
        }
    }

```

3


```
for (m = 0; m < process number; m++)
```

```
{
    for (n = 0; n < process number; n++)
    {
        if (block_arr[n] != 1) { temp = blocks[n]
            - process[m];
            if (temp >= 0)
```

```
{
    if (top < temp)
```

```
{
    process - arr [m] = n;
    top temp > 3
```

```
}
}
```

```
frag [m] = top;
block_arr [process_arr [process_arr [m]]] = 1;
```

```
top = 0;
```

```
}
```

```
}
```

```
printf (" \n process Number \t process
size \t Block Number \t Block size \t
Fragment ");
```

```
for (m = 0; m < process number; m++)
```

Ashish


```
{  
    printf("m %d l %d l %d l %d l %d  
    l %d", m, process[m], process-arr  
    [m], blocks[process-arr[m]]  
    fragments[m]);  
}
```

```
    printf("m %d",  
    return 0;
```

```
}
```

Ashish

