Question 1 (10 marks)

a) Complete the following table:

Current State	Event	Next State
New	Resources allocated	
Ready	CPU becomes available	
Running	Process requests I/O	
Waiting	I/O completes	
Running	Process completes	

b) Fill in the blanks:

i. Give any two advantages of multi-tasking: (1) (2)

ii. Privileged instructions can only be executed in the ________________ mode.

iii. When an application executes a ________________, the control is transferred to the kernel.

iv. During a context switch, process state is saved into a structure called as __________________.

Question 2 (10 marks)

Consider the following processes along with their burst times (in milli-seconds):

P1	P2	P3
20	15	25

a) Assuming the order of the arrival of the processes is P1,P2 and then P3. Use a Gantt Chart to explain the order of execution according to the Round Robin scheduling algorithm. The time slice (quantum size) is 10 milli-seconds.

b) Also compute waiting time for each process, and finally calculate the average waiting time.

P1 waits for =

P2 waits for =

P3 waits for =

Average waiting time =

Question 3 (10 marks)

You need to write a program to search an array using concurrent processes. Following is the code skeleton:

```
int main() {
        const int N = ...;
        int a[N] = ...;
        int key, i;
        cout << "Enter a number: ";
        cin >> key;

        // Write code here to search the array

        cout << "The number found at index: " << i;
        return 0;
}
```

The program searches the array for a given number, and prints the index of the number in the array. If the number is not found then the program shall output -1.

You are required to use two child processes to speed up the search. The first child shall search the first half of the array, while the second shall examine the other half.

You need not use any fancy algorithm for the search; rather you can use the simple linear search. Assume no duplicates are there in the array.