Q1: weightage 5%, 30 Minutes

- [4 Marks] Optimize the following code for better performance; use as many optimizations as possible

```
#define PI 3.142
vector <float> V(10000);//Assume initialized
for( int i=0; i<V.size(); i++){
        if (i%2==0) V[i]=V[i]+sin(PI/10);
        else V[i]=V[i]-sin(PI/20);
}
```

- [4 Marks] Given a computing machine with P_{peak} = 20 T F/s and peak bandwidth of slowest path is 5 GB/s. The machine needs to execute the following code, calculate the performance.

```
//assume a[] and s are double, and c[] and d[] are float
        for(int i=0;i<NLarge;i++) a[i]=a[i]+s*(c[i]-d[i]);
```

- Scheduling [2+5 Marks] //Assume all the task arrived at time 0
  - There are N tasks and 1 processor, how many different ways we can schedule the N tasks on one processor? // task are non-preemptive
  - There are N tasks and 2 processors, each task is non-preemptive and has a fixed execution time of either 5s or 10s. Schedule these N tasks on 2 processors such that overall execution time is minimized. (Assume n1 task with execution time 10s and n2 task with 5s and n1+n2=N)

Q1: weightage 5%, 30 Minutes

- [4 Marks] Optimize the following code for better performance; use as many optimizations as possible

```
#define PI 3.142
vector <float> V(10000);//Assume initialized
for( int i=0; i<V.size(); i++){
      if (i%2==0) V[i]=V[i]+sin(PI/10);
      else V[i]=V[i]-sin(PI/20);
}
```

1. V.size(); // Can be taken out of the loop
2. Sin(PI/10) and sin(PI/20) can be precomputed and put in to look up table (LUT)
3. For loop can be spit in to two for loop one for if and other for else
4. After loop splitting loops can written using iterator to use AVX/simidization

Q1: weightage 5%, 30 Minutes

- [4 Marks] Given a computing machine with P_{peak} = 20 T F/s and peak bandwidth of slowest path is 5 GB/s. The machine needs to execute the following code, calculate the performance.

```
//assume a[] and s are double, and c[] and d[] are float
        for(int i=0;i<NLarge;i++) a[i]=a[i]+s*(c[i]-d[i]);
```

  - Ppeak=20 TF/s, bs=5GB/s
  - Memory movement : 16B for a[i], 4B for c[i], 4B for d[i], 0 for s and total 24B
  - FLOPs per iteration : 3
  - Is=3/24 F/B, Bc= 8B/F

- Performance = $P = \min(P_{peak}, I*b_s)$

  $=\min(20\ TF/s, 3/24\ F/B * 5\ G.B/s)$

  $=\min(20\ TF/s, 0.625\ GF/s)$

  $=0.625\ G\ F/s$

Q1: weightage 5%, 30 Minutes

- Scheduling [2+5 Marks] //Assume all the task arrived at time 0
- There are N tasks and 1 processor, how many different ways we can schedule the N tasks on one processor? // task are non-preemptive
  - **ANS: N!**
- There are N tasks and 2 processors, each task is non-preemptive and has a fixed execution time of either 5s or 10s. Schedule these N tasks on 2 processors such that overall execution time is minimized. (Assume n1 task with execution time 10s and n2 task with 5s and n1+n2=N)
  - ANS**: Sort all the task based on Longest Task First and allocate one by one to least loaded processor. This will produce optimal result.**
  - ANS: (Another Approach but same as earlier) Take first all n1 10s tasks divide tem equally, if n1 is even both the processor have equal amount of work. But if n1 is odd, one processor have 10s of extra work. For n2 5s of works.
    - Case n2=0, previous one is optimal
    - If n2=1 or n2=2, then put the n2 task to less loaded processor
    - Else schedule n2-2 5s tasks as done for 10s tasks