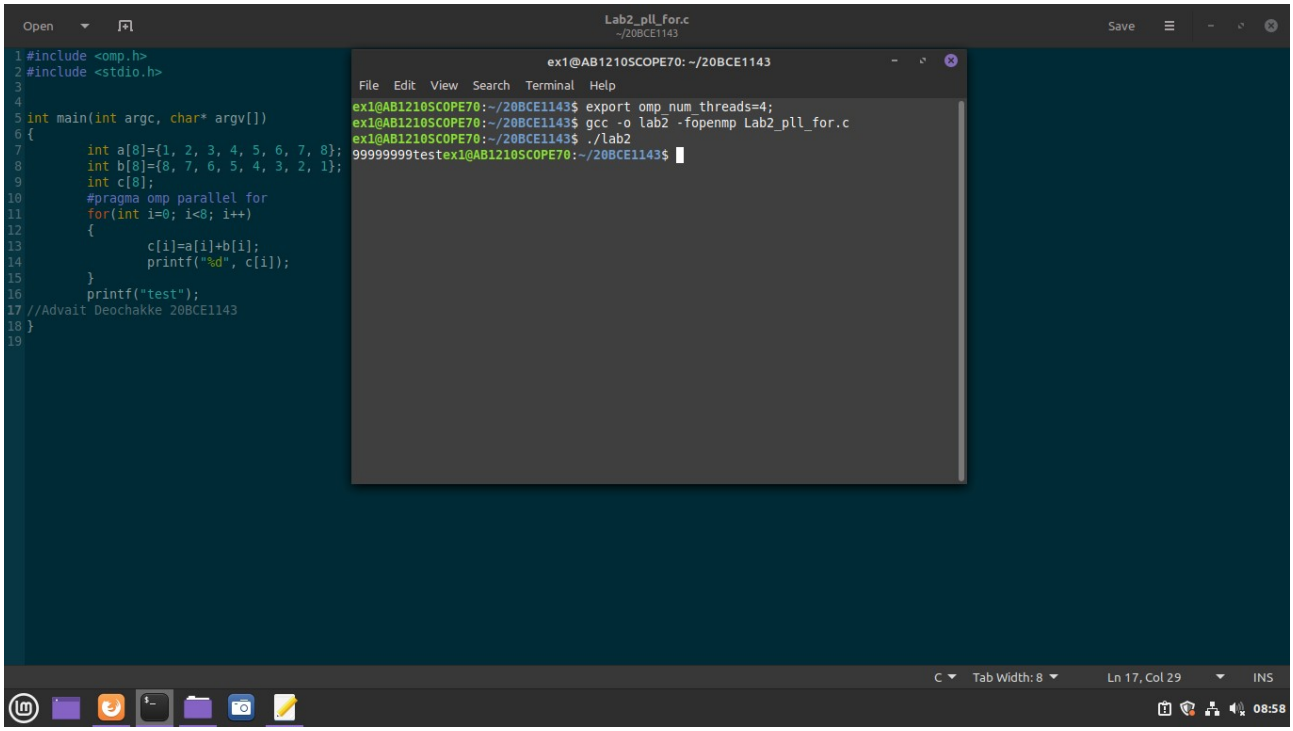
PDC Lab 2

Advait Deochakke

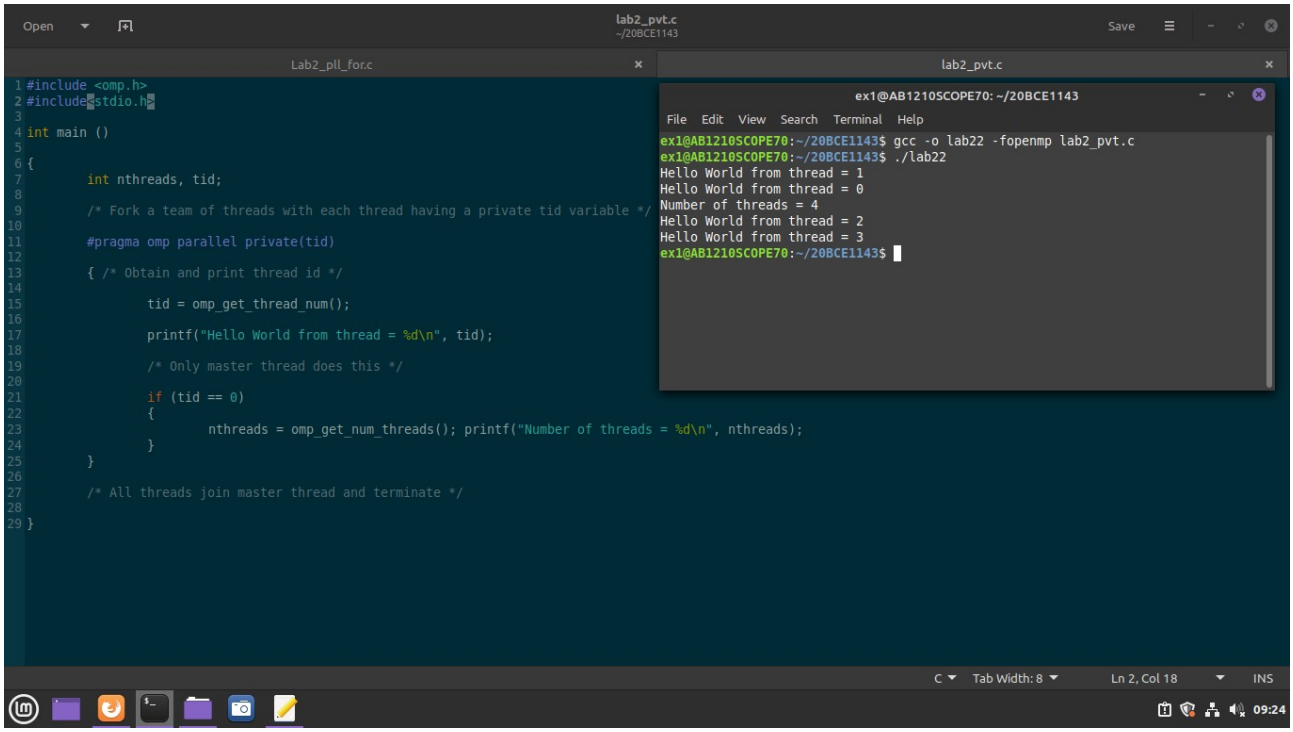
20BCE1143 – 2 August ‘22

Question 1: Array Addition using parallel for



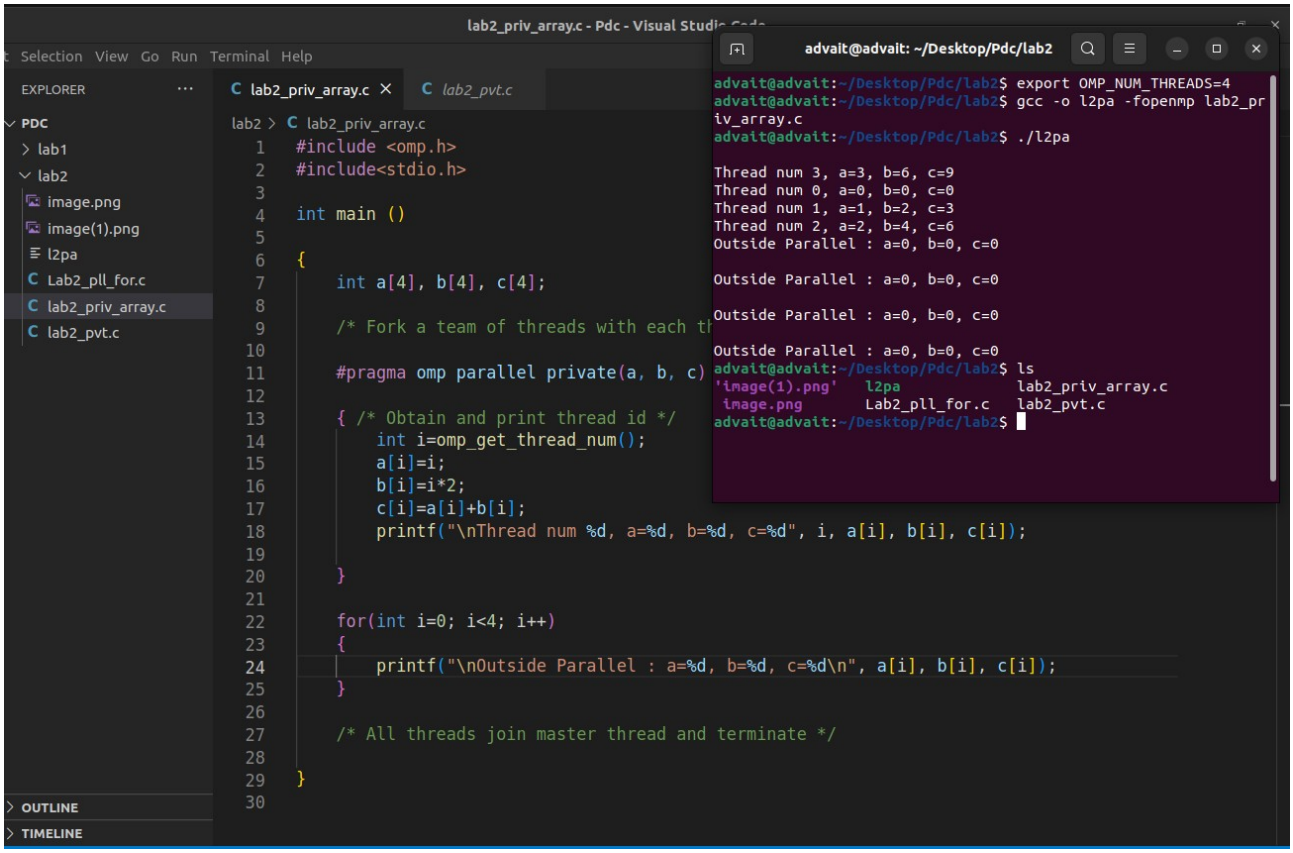

```
#pragma omp parallel for
  for(int i=0; i<8; i++)
  {
    c[i]=a[i]+b[i];
    printf("%d", c[i]);
  }
```

Question 2 & 3: Sample for Shared and Private Variable




```c
#include <omp.h>
#include<stdio.h>
int main ()
{
  int nthreads, tid;
  /* Fork a team of threads with each thread having a private tid
variable */
  #pragma omp parallel private(tid)
  { /* Obtain and print thread id */
    tid = omp_get_thread_num();
    printf("Hello World from thread = %d\n", tid);
    /* Only master thread does this */
    if (tid == 0)
    {
      nthreads = omp_get_num_threads(); printf("Number of threads = %d\
n", nthreads);
    }
  }
  /* All threads join master thread and terminate */
}
```

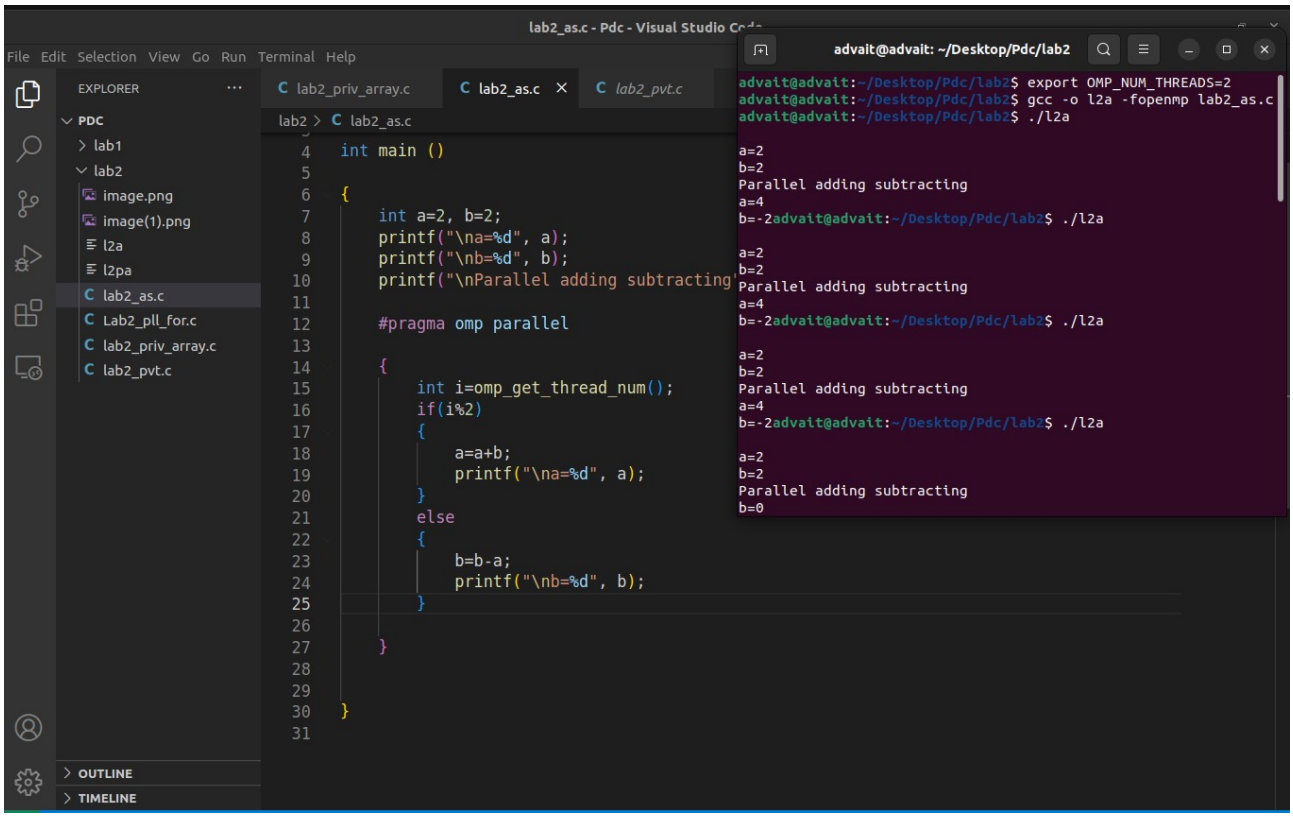
Question 4: Array addition using parallel for with a, b, c as private arrays




```c
#include <omp.h>
#include<stdio.h>
int main ()
{
  int a[4], b[4], c[4];
  /* Fork a team of threads with each thread having a private tid variable */
  #pragma omp parallel private(a, b, c)
  { /* Obtain and print thread id */
    int i=omp_get_thread_num();
    a[i]=i;
    b[i]=i*2;
    c[i]=a[i]+b[i];
    printf("\nThread num %d, a=%d, b=%d, c=%d", i, a[i], b[i], c[i]);
  }
  for(int i=0; i<4; i++)
  {
    printf("\nOutside Parallel : a=%d, b=%d, c=%d\n", a[i], b[i], c[i]);
  }

  /* All threads join master thread and terminate */
}
```

Question 5: Parallelize addition and subtraction of two integer variables a and b




```
#pragma omp parallel

  {
    int i=omp_get_thread_num();
    if(i%2)
    {
      a=a+b;
      printf("\na=%d", a);
    }
    else
    {
      b=b-a;
      printf("\nb=%d", b);
    }

  }
```