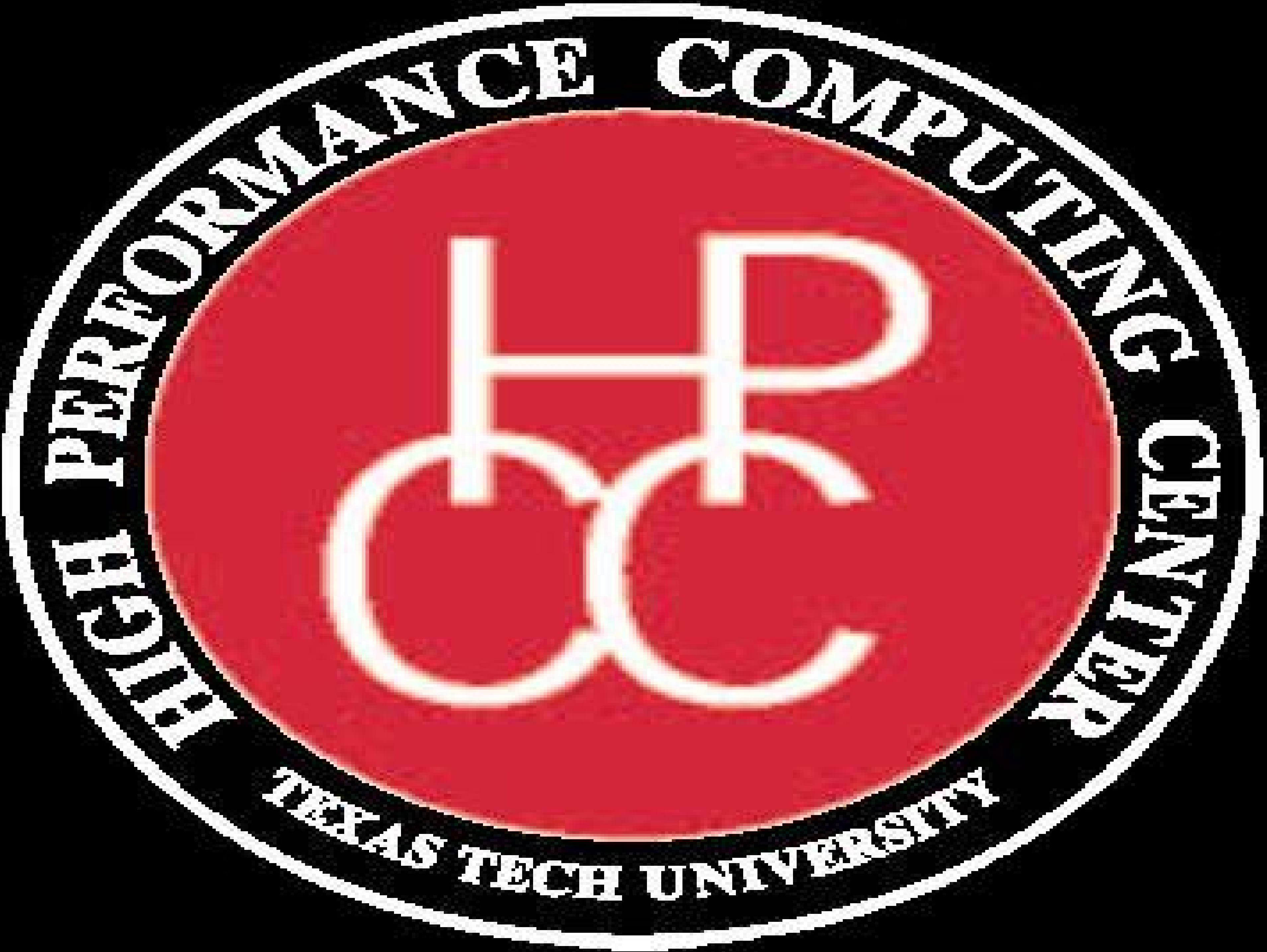

HIGH PERFORMANCE COMPUTING CENTER
HPCC
TEXAS TECH UNIVERSITY

Quadrature.c

```
#include<stdio.h>
#include <omp.h>
#define PI  3.141592653589793238462643
double f(a)
double a;
{
  return (4.0 / (1.0 + a*a));
}
main(int argc, char *argv[])
{
int  i,n;
double h, pi, sum, x;
```

```
# pragma omp parallel private (i,x)
{
for (;;)
{
 # pragma omp single
{
printf("Enter the number of intervals:(0 quits)");
scanf("%d",&n);
printf("%d \n", n);
}
if (n==0) break;
 # pragma omp single
{
h = 1.0/n;
sum = 0.0;
}
}
# pragma omp for reduction (+:sum)
        for(i=1;i<=n;i++)
        {
        x = h*(i-0.5);
        sum += f(x);
}
# pragma omp single
{
pi =h*sum;
printf("Value of PI is %f \n",pi);
}
```

Compile and Run

```
/scratch//scratch/MA5345/OPENMP
```

To Compile

```
./compileopenMP quadrature.c
```

To run

```
./runsgi
```

Quadrature.f

```
program pisimpsons
     integer :: Nints, Nints2, i
     double precision :: deltax, halfdeltax, x, mult, area, pi
     integer :: count, count_start, count_stop, clock_freq
     real :: time
     write (6,*) 'How many intervals?'
     read (5,*) Nints
     deltax = 2.0d0 / Nints
     halfdeltax = 0.5d0 * deltax
     Nints2 = 2 * Nints
     area = 0.0d0
```

Continued

```
!$omp parallel do shared(deltax,halfdeltax,Nints2)
  private(x,mult) reduction(+:area)
   do i = 1, Nints2-1
    x = -1.0d0 + dble(i) * halfdeltax
     mult = (dble(mod(i,2))*2.0d0) + 2.0d0
    area = area + mult * sqrt(1.0d0 - x*x) * halfdeltax
  end do
  area = area / 3.0d0
  pi = area * 2.0d0
  write (6,*) 'The value of pi is ', pi
  end
```

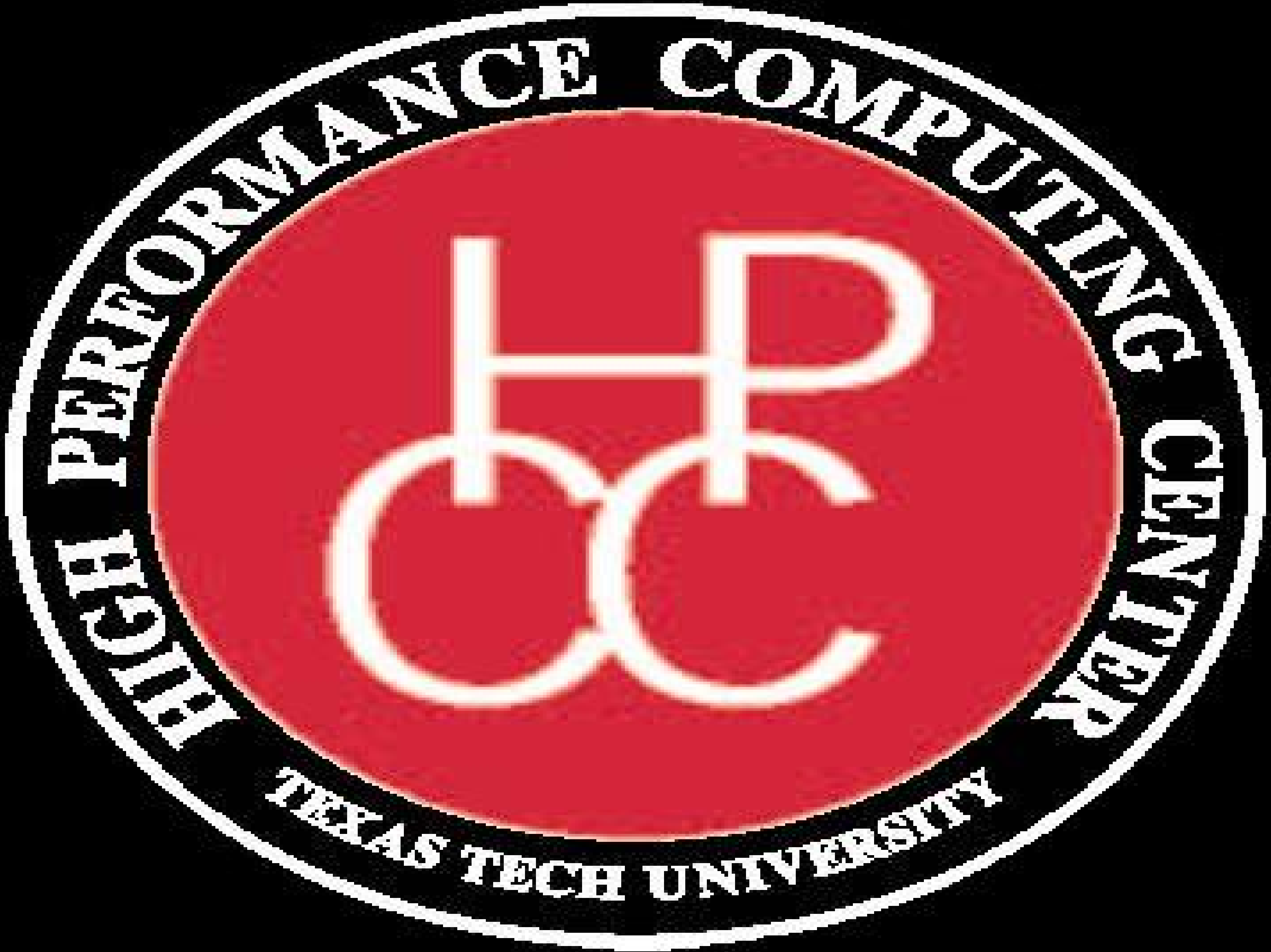
Compile and Run

Compile

```
g77 quadrature.f
```

Run

```
./runsgi
```


HIGH PERFORMANCE COMPUTING CENTER
HPCC
TEXAS TECH UNIVERSITY