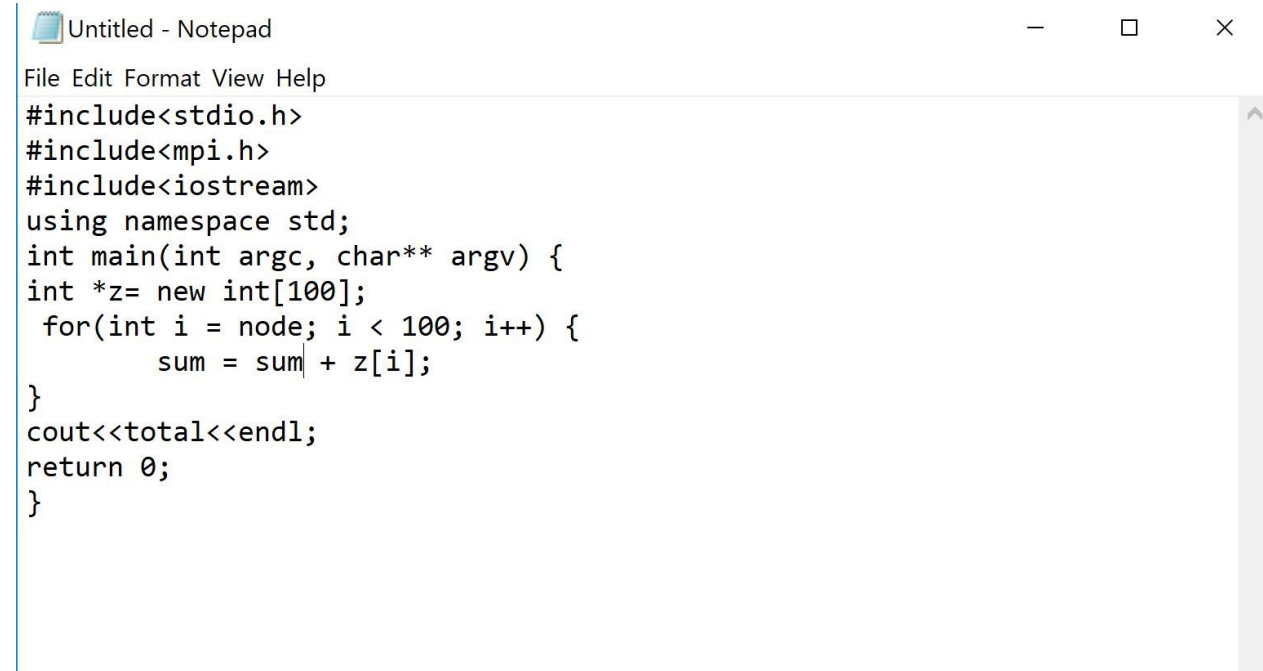


Question 22

Sequential

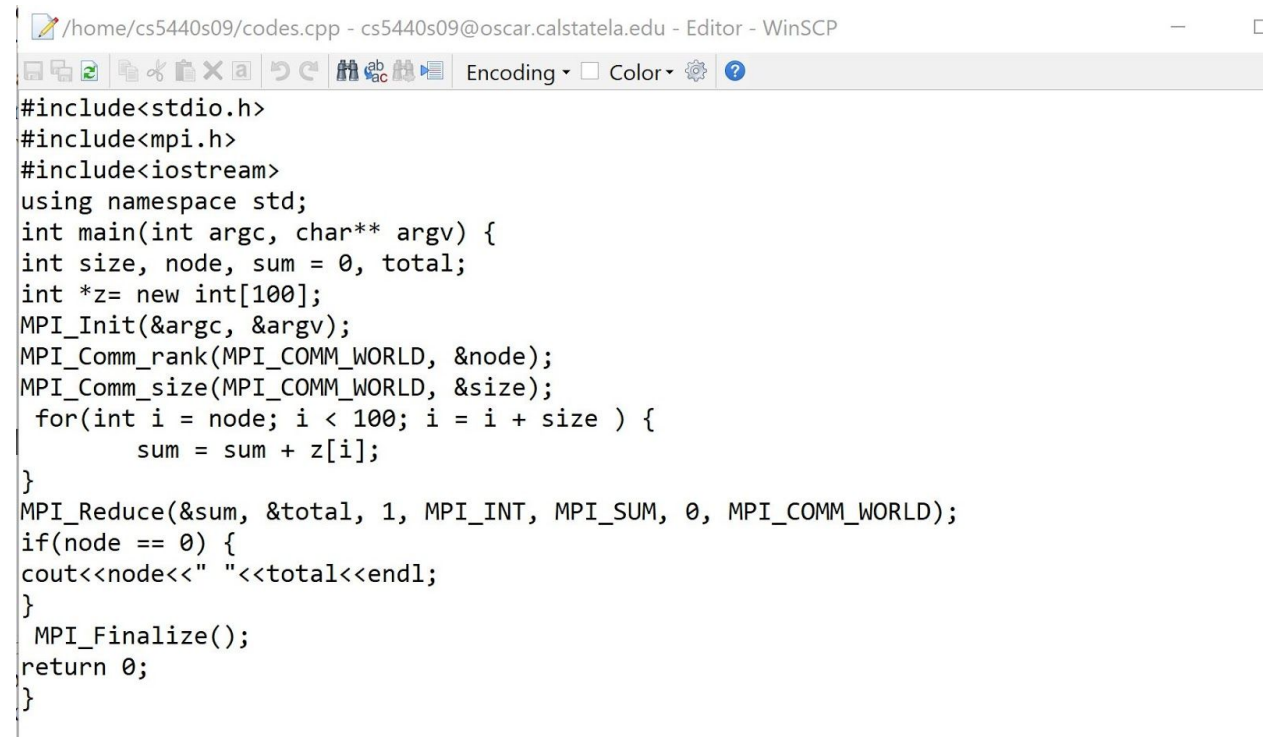


Untitled - Notepad

File Edit Format View Help

```
#include<stdio.h>
#include<mpi.h>
#include<iostream>
using namespace std;
int main(int argc, char** argv) {
int *z= new int[100];
    for(int i = node; i < 100; i++) {
        sum = sum + z[i];
    }
cout<<total<<endl;
return 0;
}
```

Parallel



/home/cs5440s09/codes.cpp - cs5440s09@oscar.calstatela.edu - Editor - WinSCP

Encoding Color

```
#include<stdio.h>
#include<mpi.h>
#include<iostream>
using namespace std;
int main(int argc, char** argv) {
int size, node, sum = 0, total;
int *z= new int[100];
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &node);
MPI_Comm_size(MPI_COMM_WORLD, &size);
    for(int i = node; i < 100; i = i + size ) {
        sum = sum + z[i];
    }
MPI_Reduce(&sum, &total, 1, MPI_INT, MPI_SUM, 0, MPI_COMM_WORLD);
if(node == 0) {
cout<<node<<" "<<total<<endl;
}
    MPI_Finalize();
return 0;
}
```

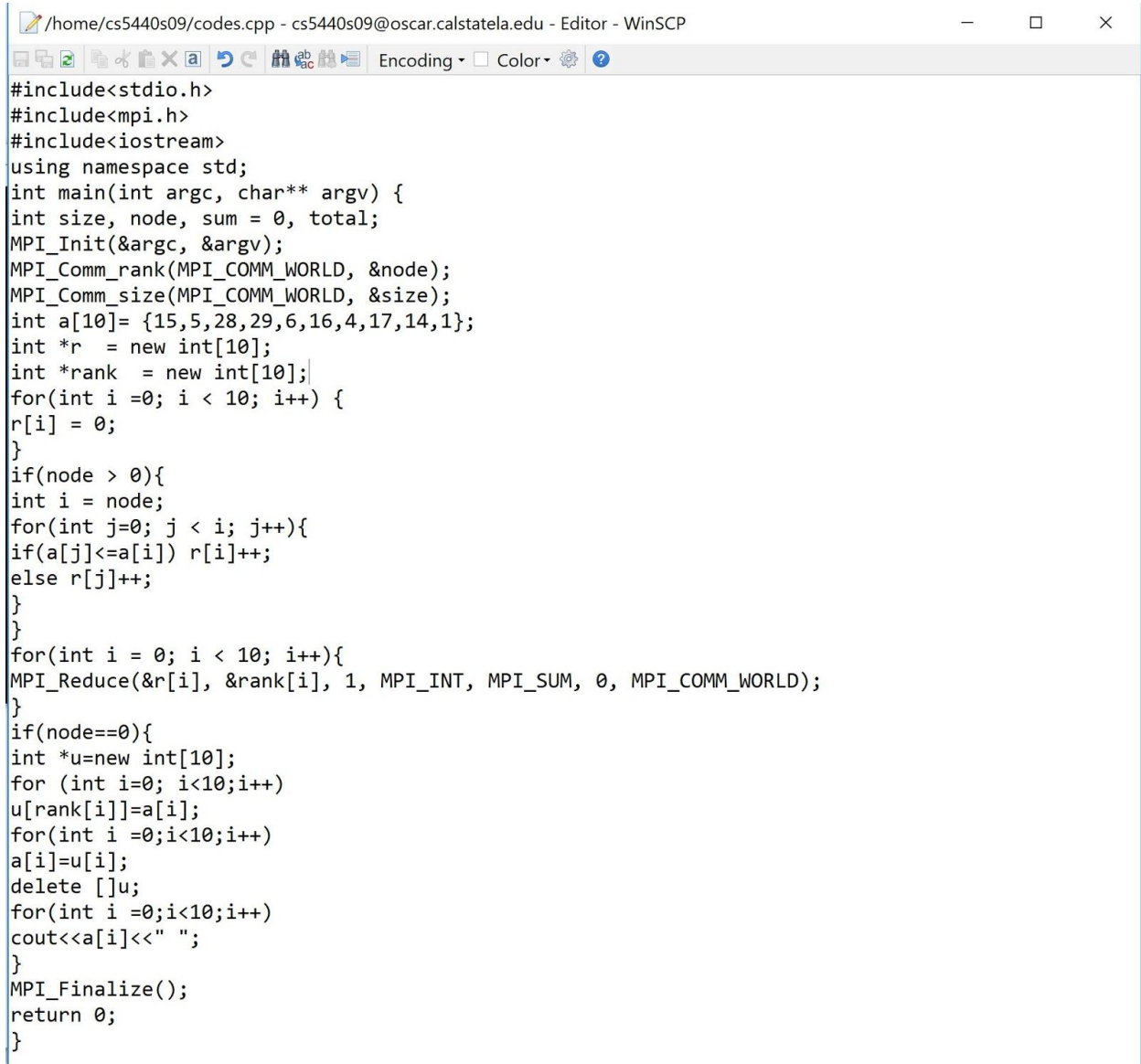
Question 23

Sequential

[File](#) [Edit](#) [Format](#) [View](#) [Help](#)

```
#include<stdio.h>
#include<mpi.h>
#include<iostream>
using namespace std;
int main(int argc, char** argv) {
int a[10]= {5,2,8,9,3,6,1,7,4,0};
int *r  = new int[10];
for(int i =0; i < 10; i++) {
r[i] = 0;
}
for(int i=1; i<10; i++){
for(int j=0; j<i; j++){
if(a[j]<=a[i]) r[i]++;
else r[j]++;
}
}
int *u=new int[10];
for (int i=0;i<10;i++)
u[r[i]]=a[i];
for(int i =0;i<10;i++)
a[i]=u[i];
delete []u;
for(int i =0;i<10;i++)
cout<<a[i];
return 0;
}
```

Parallel

A screenshot of a WinSCP Editor window. The title bar shows the file path "/home/cs5440s09/codes.cpp - cs5440s09@oscar.calstatela.edu - Editor - WinSCP". The window contains C++ code for a parallel MPI program. The code includes headers for stdio, mpi, and iostream, and uses the std namespace. It defines a main function that takes argc and argv, initializes MPI, and implements a parallel sorting algorithm using MPI_Reduce. The array 'a' contains the values {15, 5, 28, 29, 6, 16, 4, 17, 14, 1}. The program sorts the array in parallel across nodes and prints the result on the master node (node 0).

```
#include<stdio.h>
#include<mpi.h>
#include<iostream>
using namespace std;
int main(int argc, char** argv) {
int size, node, sum = 0, total;
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &node);
MPI_Comm_size(MPI_COMM_WORLD, &size);
int a[10]= {15,5,28,29,6,16,4,17,14,1};
int *r = new int[10];
int *rank = new int[10];
for(int i =0; i < 10; i++) {
r[i] = 0;
}
if(node > 0){
int i = node;
for(int j=0; j < i; j++){
if(a[j]<=a[i]) r[i]++;
else r[j]++;
}
}
for(int i = 0; i < 10; i++){
MPI_Reduce(&r[i], &rank[i], 1, MPI_INT, MPI_SUM, 0, MPI_COMM_WORLD);
}
if(node==0){
int *u=new int[10];
for (int i=0; i<10;i++)
u[rank[i]]=a[i];
for(int i =0;i<10;i++)
a[i]=u[i];
delete []u;
for(int i =0;i<10;i++)
cout<<a[i]<<" ";
}
MPI_Finalize();
return 0;
}
```

Question 24

Sequential

```
1 #include<stdio.h>
2 #include<mpi.h>
3 #include<iostream>
4 using namespace std;
5 int main(int argc, char** argv) {
6
7     float x, sum = 0.0;
8
9     for(x = 0; x <=1; x=x+0.02) {
10         sum = sum + 1/(1+(x*x));
11     }
12
13     cout<<sum<<endl;
14
15     return 0;
16 }
```

Parallel

```
/home/cs5440s09/codes.cpp - cs5440s09@oscar.calstatela.edu - Editor - WinSCP
Encoding ▾ Color ▾ ?

#include<stdio.h>
#include<mpi.h>
#include<iostream>
using namespace std;
int main(int argc, char** argv) {
    int size, node;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &node);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    float increment = size * 0.02, init = node * 0.02, sum = 0.0, total;

    for(float x = init; x <= 1.0; x = x + increment){
        sum = sum + (1/(1+(x*x)));
    }

    MPI_Reduce(&sum, &total, 1, MPI_FLOAT, MPI_SUM, 0, MPI_COMM_WORLD);

    if(node == 0){
        cout<<total<<endl;
    }

    MPI_Finalize();
    return 0;
}
```

Question 25

Sequential

```
1 #include<stdio.h>
2 #include<mpi.h>
3 #include<iostream>
4 #include <math.h>
5 using namespace std;
6
7 long factorial(int x) {
8     if(x==0 || x ==1) return 1;
9     else return x * factorial(x-1);
10 }
11
12 int main(int argc, char** argv) {
13
14     int x=5, accuracy = 100;
15     float sum = 0, total;
16
17     for(int i = 0; i < accuracy; i++){
18         sum = sum + (pow(x, i)/factorial(i));
19     }
20
21
22     cout<<sum<<endl;
23
24
25     MPI_Finalize();
26     return 0;
27 }
```

Parallel

```
/home/cs5440s09/codes.cpp - cs5440s09@oscar.calstatela.edu - Editor - WinSCP
Encoding Color
#include<stdio.h>
#include<mpi.h>
#include<iostream>
#include <math.h>
using namespace std;

long factorial(int x) {
if(x==0 || x ==1) return 1;
else return x * factorial(x-1);
}

int main(int argc, char** argv) {
int size, node;
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &node);
MPI_Comm_size(MPI_COMM_WORLD, &size);

int x=5, accuracy = 100;
float sum = 0, total;

for(int i = node; i < accuracy; i= i+ size){
sum = sum + (pow(x, i)/factorial(i));
}

MPI_Reduce(&sum, &total, 1, MPI_FLOAT, MPI_SUM, 0, MPI_COMM_WORLD);

if(node == 0){
cout<<total<<endl;
}

MPI_Finalize();
return 0;
}
```