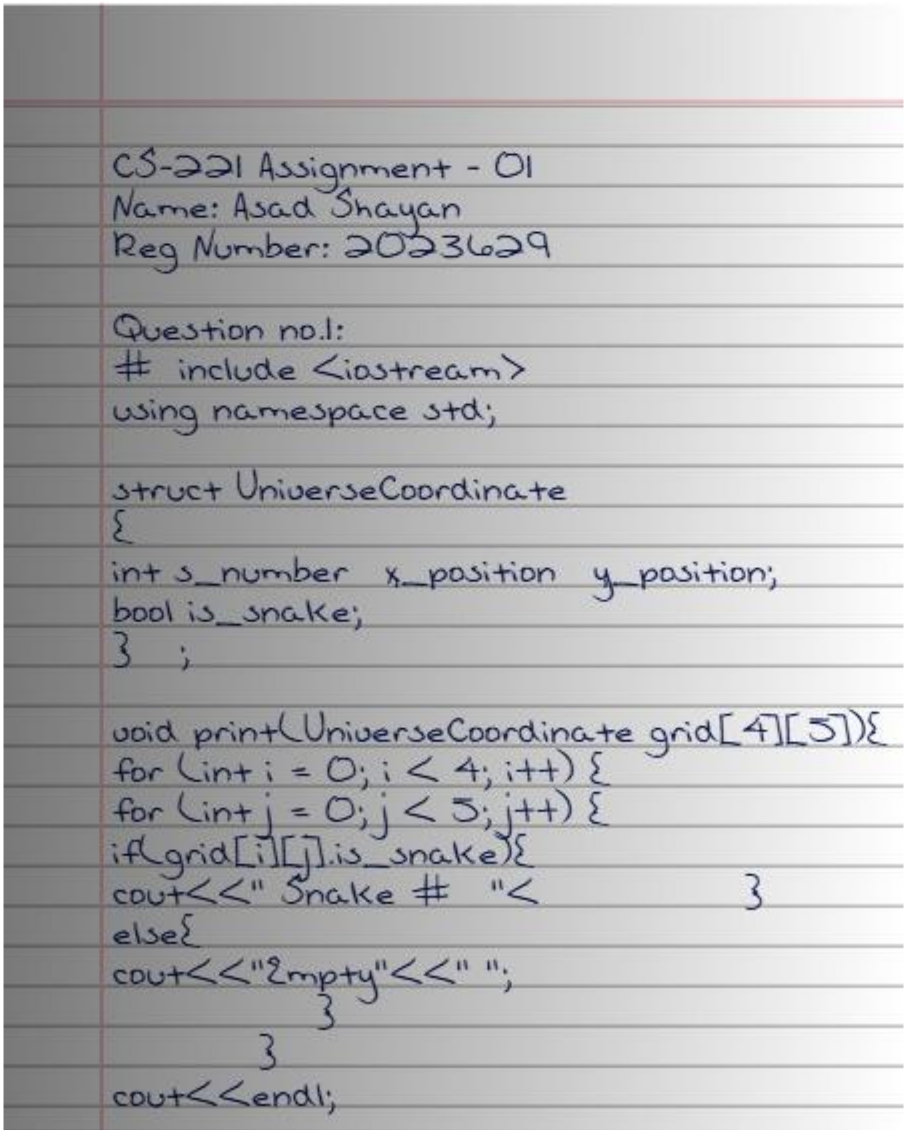
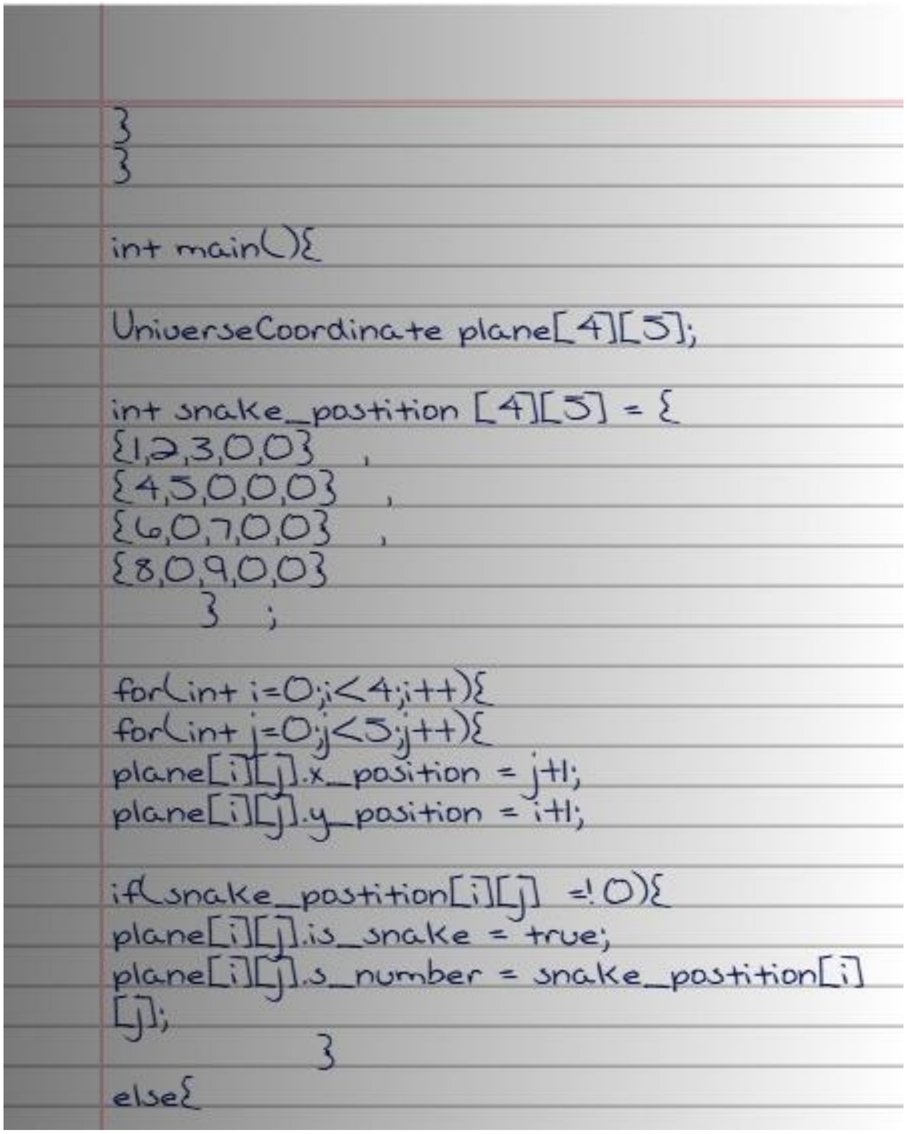


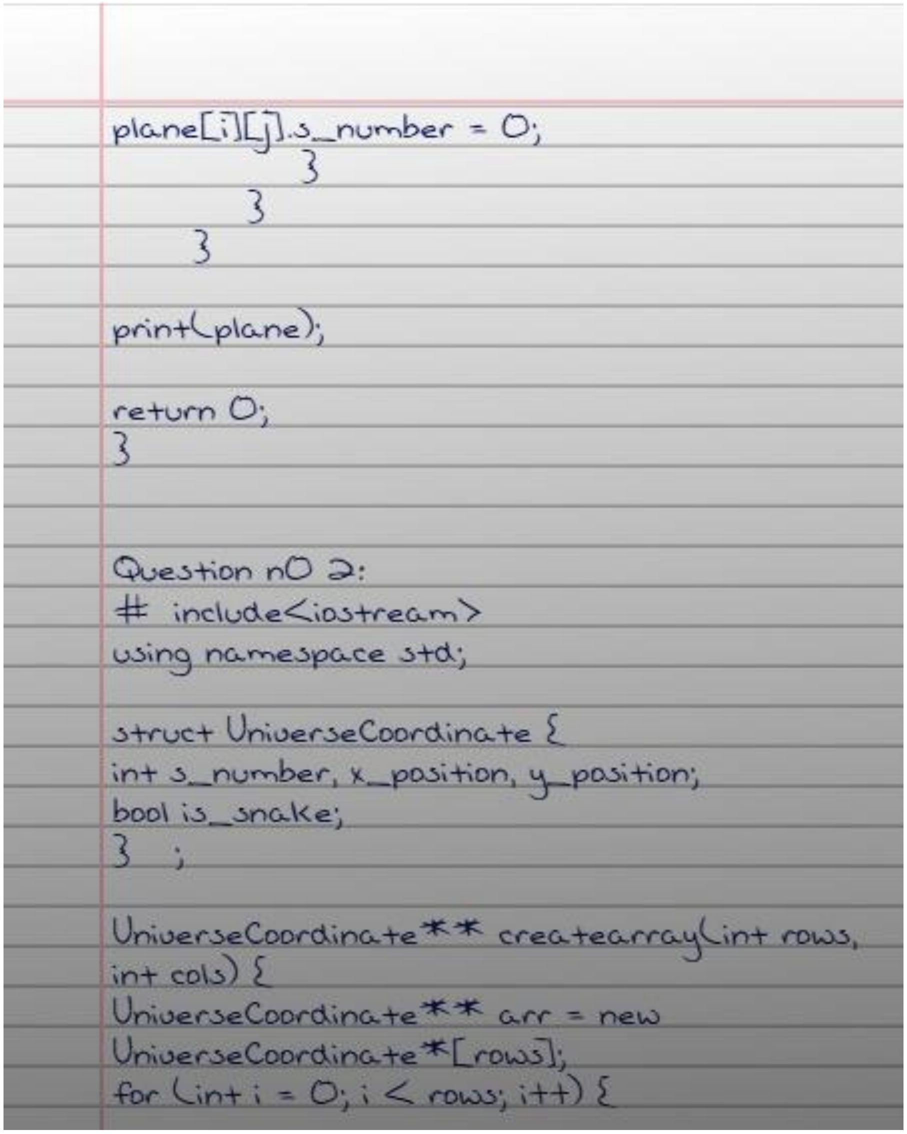
CS-221 Assignment - 01
Name: Asad Shayan
Reg Number: 2023629

Question no.1:

```
# include <iostream>
using namespace std;

struct UniverseCoordinate
{
int s_number  x_position  y_position;
bool is_snake;
}  ;

void print(UniverseCoordinate grid[4][5]){
for (int i = 0; i < 4; i++) {
for (int j = 0; j < 5; j++) {
if(grid[i][j].is_snake){
cout<<" Snake #  "<                }
else{
cout<<"2mpty"<<" ";
          }
      }
cout<<endl;
```



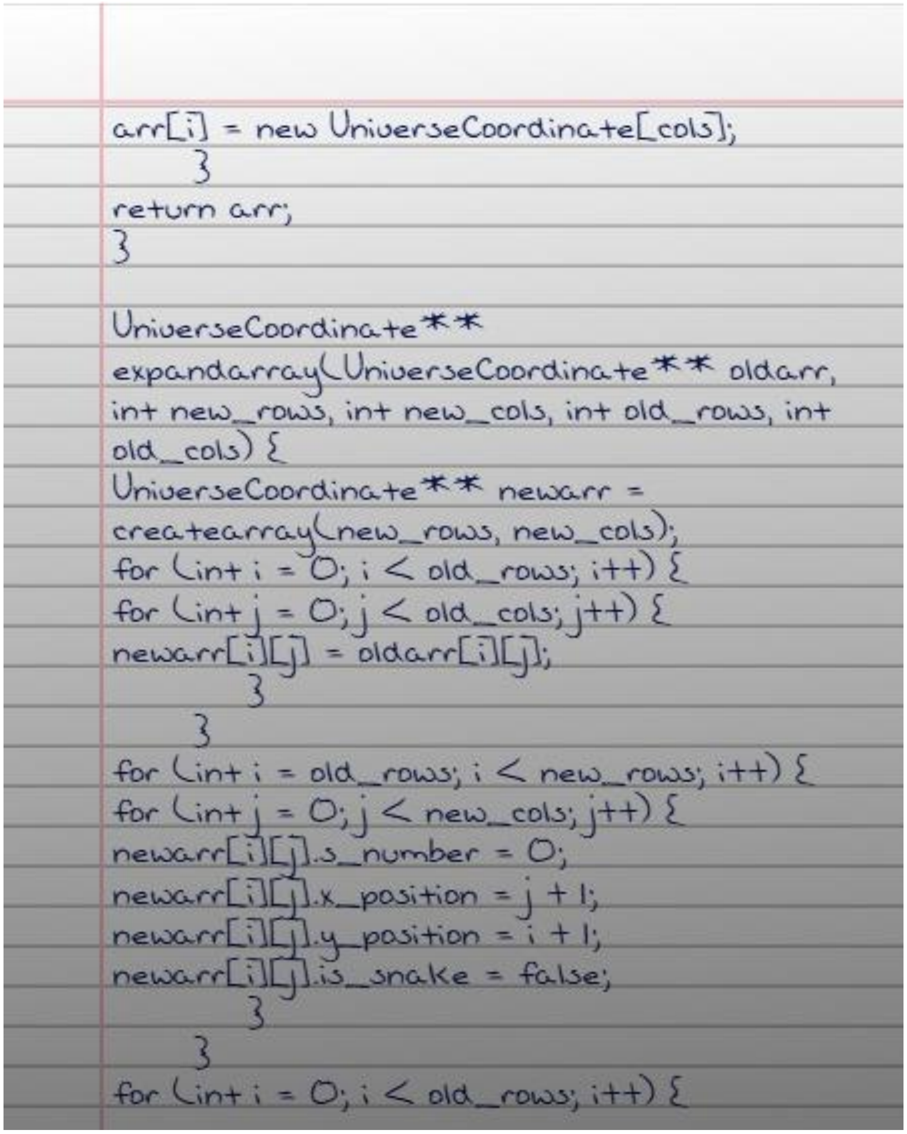
```
}
}

int main(){

UniverseCoordinate plane[4][5];

int snake_postition [4][5] = {
{1,2,3,0,0}   ,
{4,5,0,0,0}   ,
{6,0,7,0,0}   ,
{8,0,9,0,0}
      }  ;

for(int i=0;i<4;i++){
for(int j=0;j<5;j++){
plane[i][j].x_position = j+1;
plane[i][j].y_position = i+1;

if(snake_postition[i][j]  =! 0){
plane[i][j].is_snake = true;
plane[i][j].s_number = snake_postition[i]
[j];
            }
else{
```



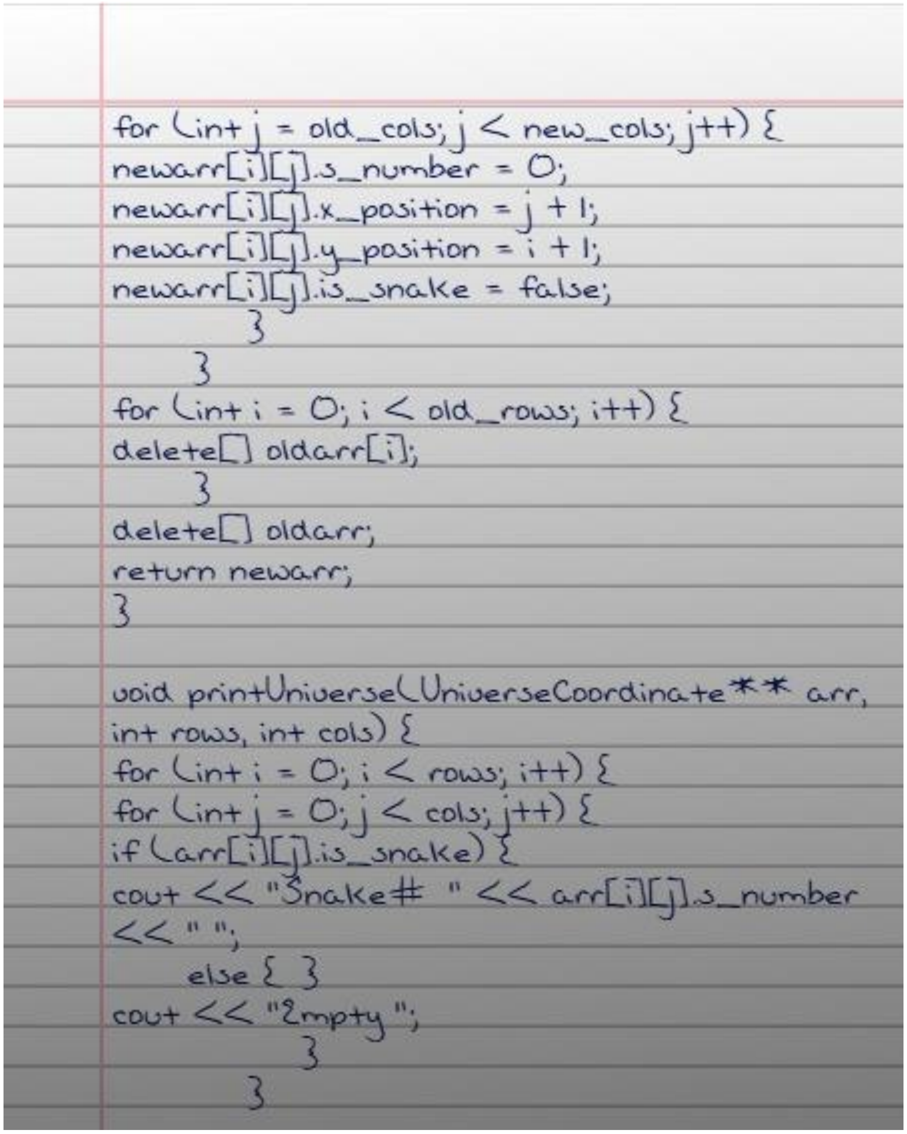
```
plane[i][j].s_number = 0;
                }
            }
        }

print(plane);

return 0;
}
```

Question nO 2:

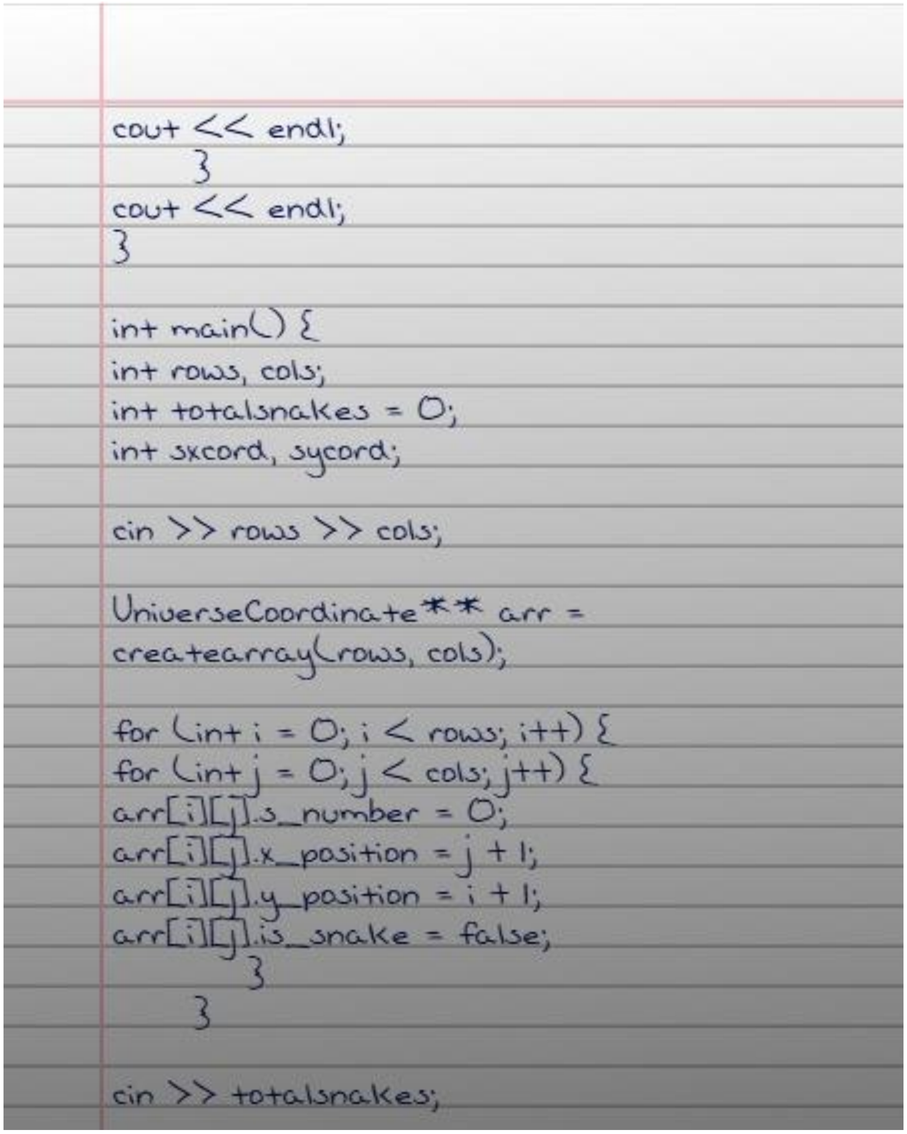
```
# include<iostream>
using namespace std;

struct UniverseCoordinate {
int s_number, x_position, y_position;
bool is_snake;
}   ;

UniverseCoordinate** createarray(int rows,
int cols) {
UniverseCoordinate** arr = new
UniverseCoordinate*[rows];
for (int i = 0; i < rows; i++) {
```



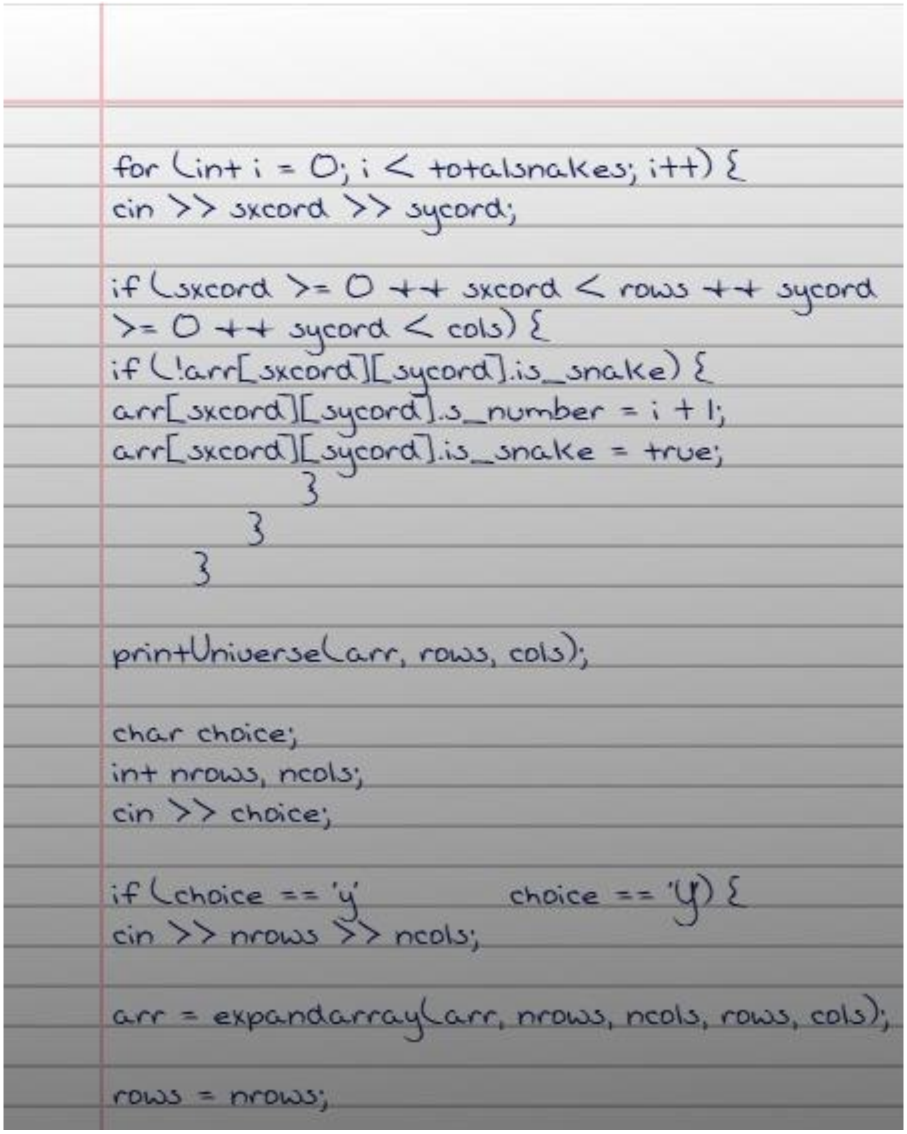
```
arr[i] = new UniverseCoordinate[cols];
      }
return arr;
}

UniverseCoordinate**
expandarray(UniverseCoordinate** oldarr,
int new_rows, int new_cols, int old_rows, int
old_cols) {
UniverseCoordinate** newarr =
createarray(new_rows, new_cols);
for (int i = 0; i < old_rows; i++) {
for (int j = 0; j < old_cols; j++) {
newarr[i][j] = oldarr[i][j];
            }
      }
for (int i = old_rows; i < new_rows; i++) {
for (int j = 0; j < new_cols; j++) {
newarr[i][j].s_number = 0;
newarr[i][j].x_position = j + 1;
newarr[i][j].y_position = i + 1;
newarr[i][j].is_snake = false;
            }
      }
for (int i = 0; i < old_rows; i++) {
```



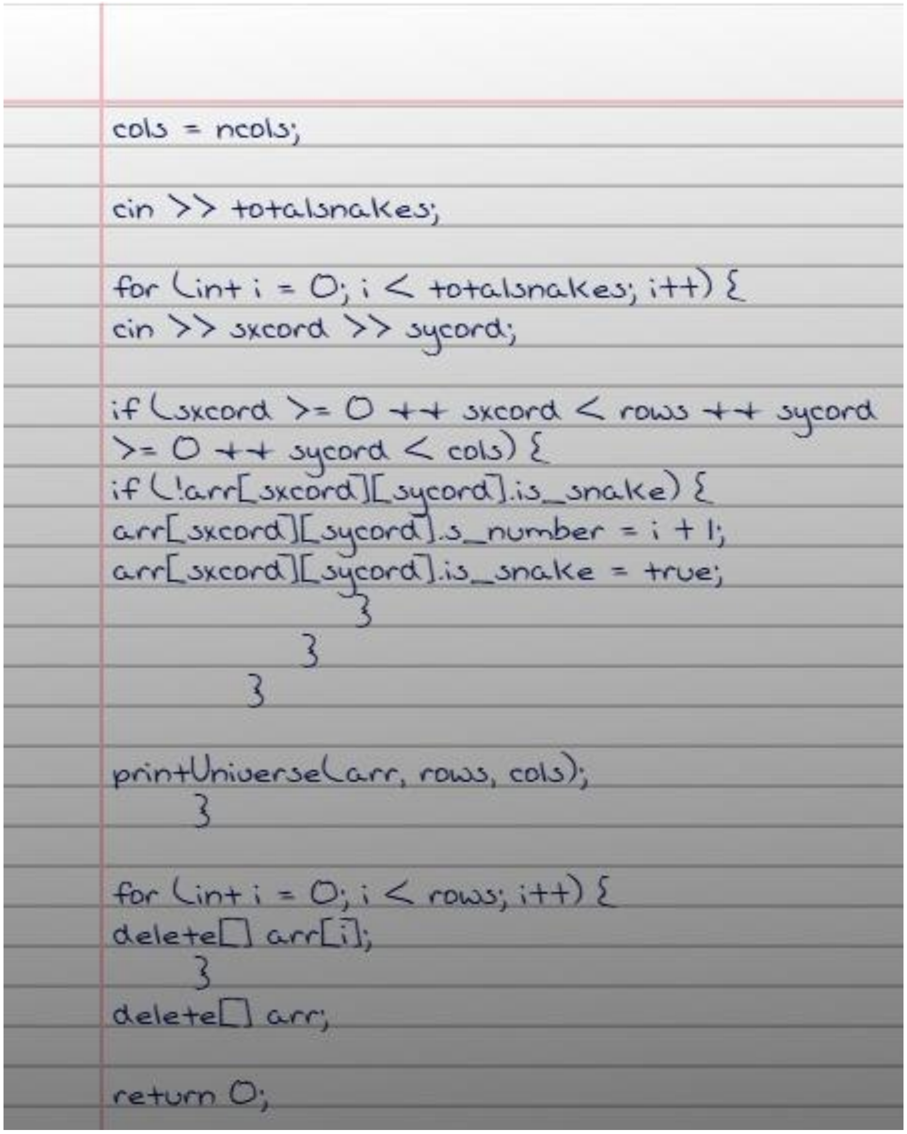
```
for (int j = old_cols; j < new_cols; j++) {
newarr[i][j].s_number = 0;
newarr[i][j].x_position = j + 1;
newarr[i][j].y_position = i + 1;
newarr[i][j].is_snake = false;
          }
      }
for (int i = 0; i < old_rows; i++) {
delete[] oldarr[i];
      }
delete[] oldarr;
return newarr;
}

void printUniverse(UniverseCoordinate** arr,
int rows, int cols) {
for (int i = 0; i < rows; i++) {
for (int j = 0; j < cols; j++) {
if (arr[i][j].is_snake) {
cout << "Snake#  " << arr[i][j].s_number
<< " ";
     else { }
cout << "Empty";
              }
          }
```



```
cout << endl;
    }
cout << endl;
}

int main() {
int rows, cols;
int totalsnakes = 0;
int sxcord, sycord;

cin >> rows >> cols;

UniverseCoordinate** arr =
createarray(rows, cols);

for (int i = 0; i < rows; i++) {
for (int j = 0; j < cols; j++) {
arr[i][j].s_number = 0;
arr[i][j].x_position = j + 1;
arr[i][j].y_position = i + 1;
arr[i][j].is_snake = false;
        }
    }

cin >> totalsnakes;
```



```
for (int i = 0; i < totalsnakes; i++) {
cin >> sxcord >> sycord;

if (sxcord >= 0 ++ sxcord < rows ++ sycord
>= 0 ++ sycord < cols) {
if (!arr[sxcord][sycord].is_snake) {
arr[sxcord][sycord].s_number = i + 1;
arr[sxcord][sycord].is_snake = true;
            }
        }
    }

printUniverse(arr, rows, cols);

char choice;
int nrows, ncols;
cin >> choice;

if (choice == 'y'          choice == 'Y) {
cin >> nrows >> ncols;

arr = expandarray(arr, nrows, ncols, rows, cols);

rows = nrows;
```



```
cols = ncols;

cin >> totalsnakes;

for (int i = 0; i < totalsnakes; i++) {
cin >> sxcord >> sycord;

if (sxcord >= 0 && sxcord < rows && sycord
>= 0 && sycord < cols) {
if (!arr[sxcord][sycord].is_snake) {
arr[sxcord][sycord].s_number = i + 1;
arr[sxcord][sycord].is_snake = true;
                }
              }
           }

printUniverse(arr, rows, cols);
     }

for (int i = 0; i < rows; i++) {
delete[] arr[i];
     }
delete[] arr;

return 0;
```