GHULAM ISHAQ KHAN INSTITUTE (GIKI)

Name: Muhammad Haris
Section: Bachelor's of Computer Science.
Batch 33 ; 2023428.

ASSIGNMENT No.1

CS-221 COURSE · DATA STRUCTURES & ALGORITHMS.

QUESTION NO.1.

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

struct Universe Coordinate{

int s_number; // Snake identification number
int x-position;  // x-coordinate.
int y_position;  // y-coordinate.
bool is_snake; // True if a snake is found at a position
                  false other wise.

};
```

Explanation :-

As per the instructions in the question, initialization of struct to unite different variables to store necessary information of the snake map.

3 integer type variable and 1 boolean type variable to show the precense of the snake.

QUESTION NO. 2

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

struct Universe_Coordinate
{
    int s_number;
    int x_position;
    int y_position;
    bool is_snake;
};
// Function to create a 2D array of Universe coordinates.
Universe_Coordinate** createUniverse(int width, int height)
{
    Universe_Coordinate **UniverseData = new Universe_Coordinate*[width];
    for (int i=0; i< width ; i++)
    { universeData[i] = new UniverseCoordinate [height];
      // create an array for each row.
    for (int j=0 ; j <height ; j++)
    {
        UniverseData[i][j].x_position = i;
        UniverseData[i][j].y_position = j;
        UniverseData[i][j].s_number = 0;
        UniverseData[i][j].is_snake =false
    }
return UniverseDData;
}
```

```
//function to display the universe map.
void displayUniverseMap(UniverseCoordinde **
                        UniverseData, int width, int height)
{
  cout << " Current universe map "<<endl;
   for (int x = 0 ; x <width ; x++)
   {
    for (int y =0 ; y <height ; y++)
   {
cout << "[ " << Universe Data [x][y].x-position <<
  " , " << UniverseData [x][y].y-postion << " ] ";
if (Universe Data [x] [y].is snake)
{
 cout << " Snake ID " << Universe Data[x][y].s-num
}
else
{ cout << " No snake found " <<endl;
 }
  cout <<endl;
  }
 }
}
```

```
// function to locate a snake at a spe
void locateSnake ( Universe Coordinate
UniverseData, int x, int y, int snake_id , int
int height )
{ if (x>=0 && x<width && y>=0 && y<heigh
    {
    Universe Data [x][y] . s_number = snake_id;
    Universe Data [x][y] . is_snake = true;
    cout << "Snake with ID " << snake_id <<
    "located at (" << x << "," << y << ")" <<
    endl;
}
else
    { cout << "Coordinates are out of bounds" << endl;
}
```

(Please Turn Over --->)

```
// function to expand the universe.
UniverseCoordinate **expandUniverse
(UniverseCoordinate **newUniverseData,
 int &width, int &height, int new_width,
 int new_height)
{  // create a new 2D array. (Larger)
   UniverseCoordinate **new_UniverseData=
        createUniverseData(new_width,
                           new_height);
// copy the element of previous into new universe
for (int i=0; i<width; i++)
  {  for (int j=0;j<height;++j)
     {  new_UniverseData[i][j] = old_UniverseData[i][j];
     }
  }
// deallocate memory of previous array.
for (int i=0; i<width; i++)
  {  delete [] old_UniverseData[i];
  }
  delete [] old_UniverseData;
  // update the width and height.
  width = new_width;
  height = new_height;
}
```

```
int main ()
{
    // asking the user the initial value of the Universe size
    int width, height, num_snakes;
    cout << "Enter initial Universe width (X-axis) size"
         <<endl;
    cin >> width;
    cout << "Enter initial Universe height (Y-axis) size:"
         <<endl;
    cin >> height;
    // creating universe with specified size
    Universe Coordinate** UniverseData =
                    createUniverseData (width, height);
    // asking user to input the number of snake
    cout << "Enter the number of snake to locate"
         <<endl;
    cin >> num_snakes;
    for (int i = 1 ; i <= num_snakes ; i++)
    {
        int x, y;
        cout << "Enter the coordinates for snake"
        << i << "(X Y) : ";
        cin >> x >> y;
        locateSnake (UniverseData, width,
                     height);
    }
```

// Displaying the initial Universe.

displayUniverseMap (UniverseData, width, height);

// asking the user if they want to expand the universe.

char expand;

cout << "Do you want to expand the Universe? (Y/n): ";

cin >> expand; expand ==

if (expand == 'y' || 'Y')

// expanding the universe.

{

int new_width, new_height;

cout << "Enter new universe width:" << endl;

cin >> new_width;

cout << "Enter new universe height:" << endl;

cin >> new_height;

UniverseData = expandUniverse (UniverseData,
width, height, new_width, new_height);

int more_snakes.

// asking the user to add more snake

cout << "How many more snakes you want to add
after expanding Universe" << endl;

cin >> more_snakes;

// code continued.....

for (int i = num_snakes + 1 ; i < num_snakes
+ more_snakes ; i++)

{ int x, y;

cout << "Enter the coordinates for snake" << i <<
"(x y)" << endl;

Pg no. 34

```
    cin >> x >> y;
    locateSnake (UniverseData, x, y, i,
                 width, height);
  }
  // Display the expanded universe.
displayUniverseMap (UniverseData, Width,
                    height);
  // Deallocate Memory.
for (int i=0 ; i<width ; i++)
{
  delete[] UniverseData[i];
}
  delete[]  UniverseData;
return 0;
}
```

— X —