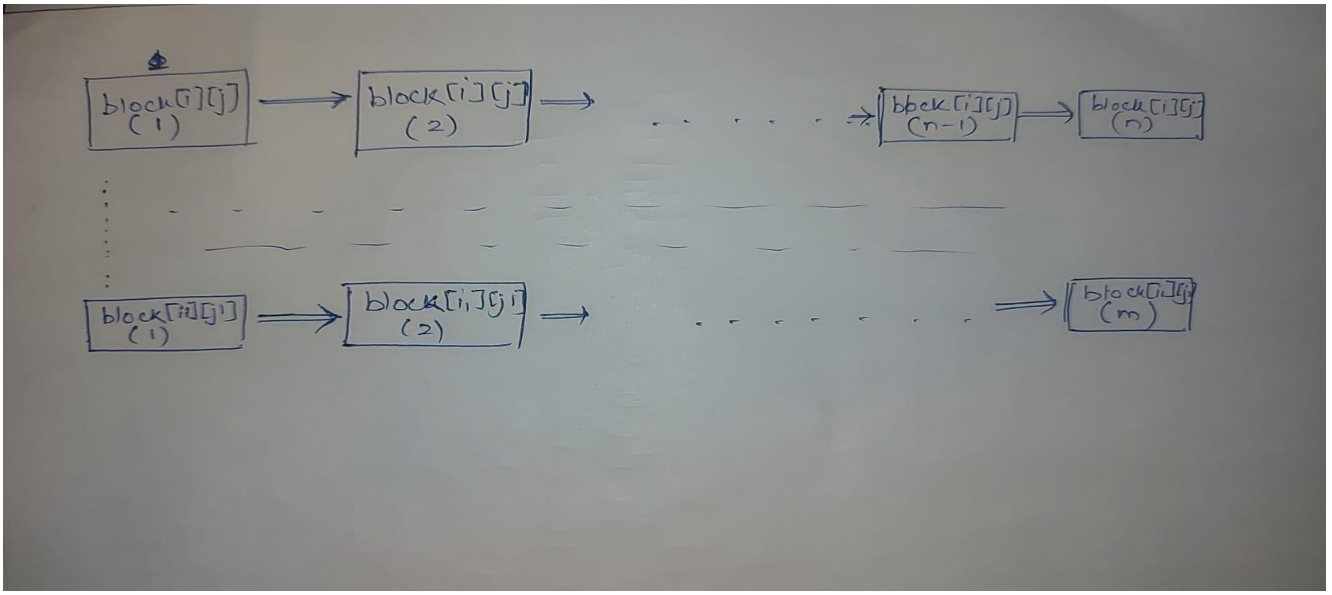
Task graph is a mathematical representation of a set of tasks and dependencies between them, It is usually in the form of Directeg acyclic graph. It is commonly used to solve the problems of scheduling and optimisation.Here each task is represented as a node and dependencies are represented as directed edges.The directed edgesshow the ordering of the tasks, with tasks that must be completed before other tasks being represented as a source node with a directed edge pointing to the target node.

A ---> B ---> C ---> D

In this graph A must be completed before B , B before C and so on.

```
}
#pragma omp task depend(inout:bmatrix[x1*n/m+y1].block)
{
    for(int a=0;a<m;a++)
    {
        for(int c=0;c<m;c++)
        {
            for(int b=0;b<m;b++)
            {
                bmatrix[x1*n/m+y1].block[a*m+b]=min(maxval,Outer(bmatrix[x1*n/m+y1].block[a*m+b],Inner(b1[i]
            }
        }
    }
}
isSet[x1*n/m+y1]=true;
```

In my code I have used depends to multiply the blocks and get the answers. Here the depends clause signifies that the computation for a block is independent from the computation of other blocks but is dependent on the previous computatioon of the same block.Hence the task graph for the program is as follows



Here block[i][j](1) represents the 1st computation of block[i][j]