APPLICATIONS OF DATA STRUCTURES & ALGORITHMS IN SYSTEMS

1. ARRAY

Where Arrays Are Applied

Operating systems, databases, embedded systems, scientific computing, web applications.

Example Applications

Student management systems, image processing software, e-commerce platforms.

Algorithms Used

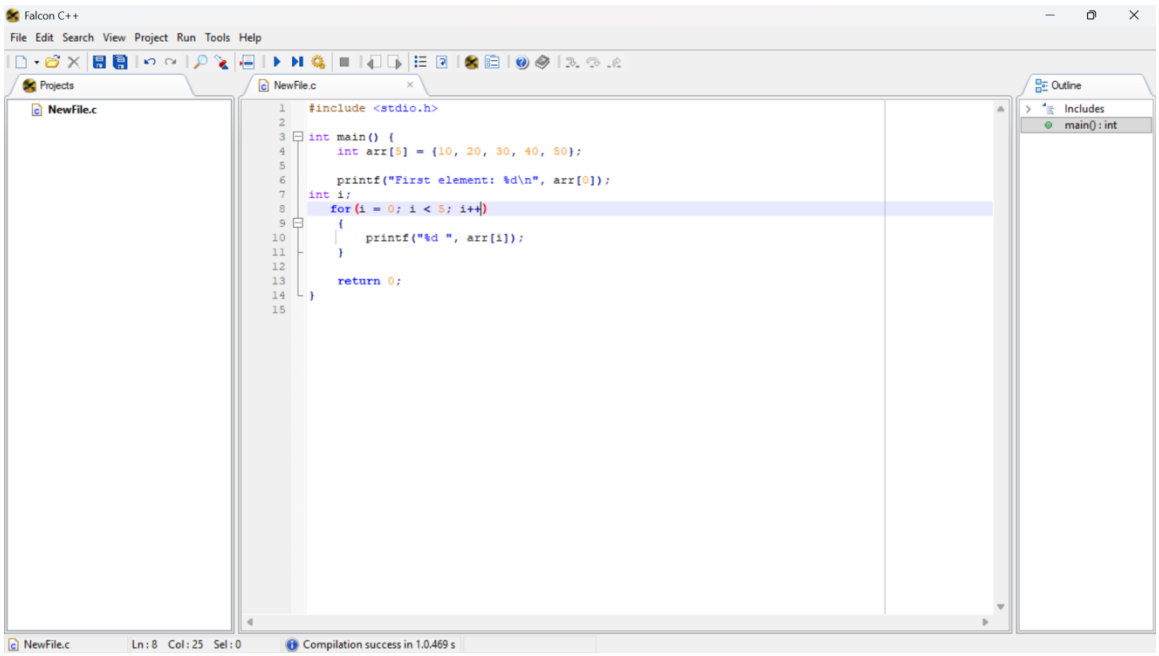
Linear Search, Binary Search, Bubble Sort, Quick Sort.

Reasons for Using Arrays

Fast access (O(1)), simple structure, efficient for fixed-size data.

How Arrays Work Within Systems

Arrays use contiguous memory allocation, allowing fast access and high performance.



2. LINKED LIST

Where Applied

Memory management, dynamic storage, system software, multimedia applications.

Example Applications

Music playlists, undo/redo systems, file navigation.

Algorithms Used

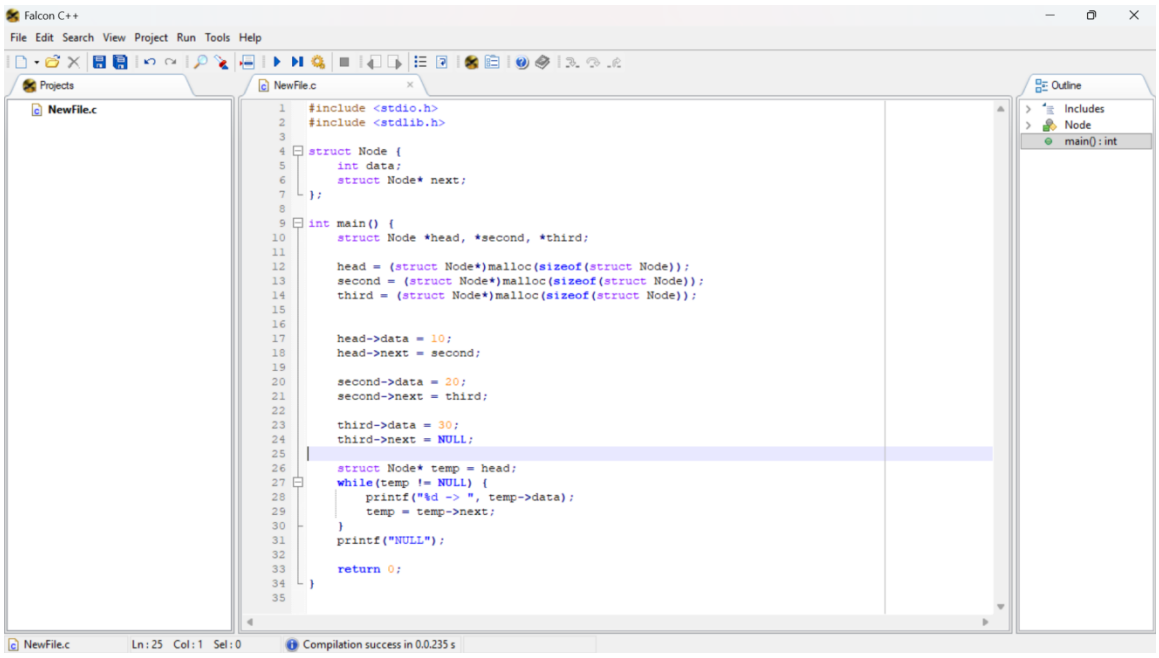
Traversal, insertion, deletion.

Reasons

Dynamic size, efficient insertion and deletion.

How They Work

Nodes stored in non-contiguous memory linked by pointers.



3. STACK

Where Applied

Operating systems, compilers, browsers.

Example Applications

Browser navigation, function calls, undo/redo.

Algorithms Used

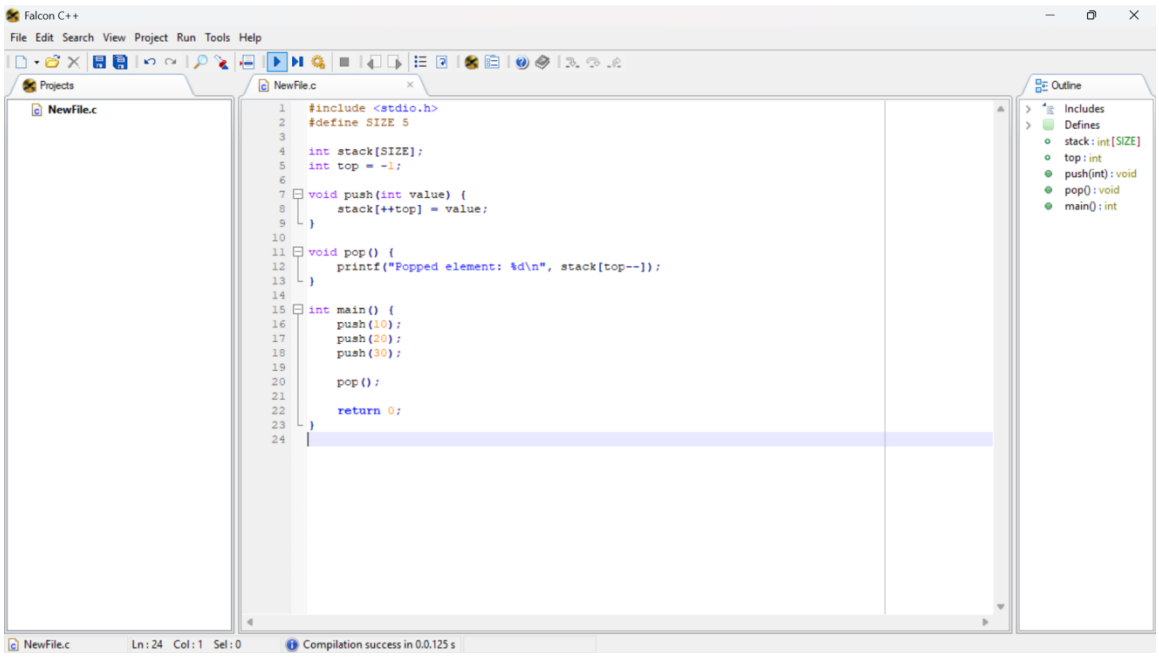
Push, Pop, Peek.

Reasons

LIFO principle, fast operations.

How They Work

Used in call stacks for function execution.



4. QUEUE

Where Applied

Operating systems, networking, real-time systems.

Example Applications

Printer spooling, CPU scheduling.

Algorithms Used

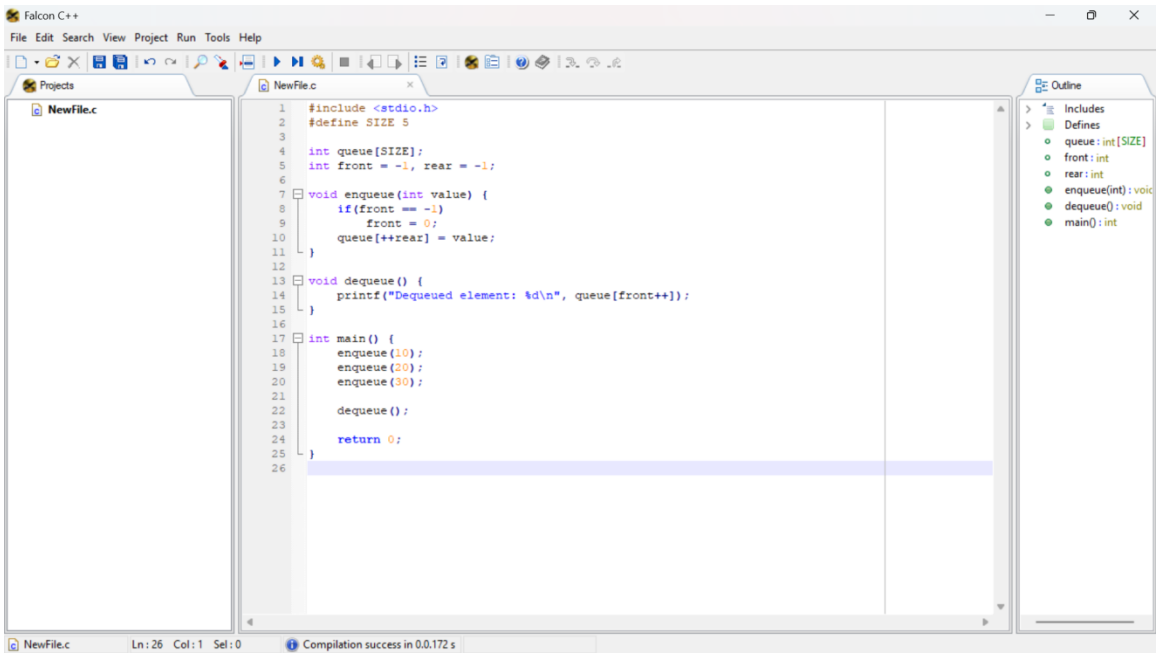
Enqueue, Dequeue.

Reasons

FIFO principle, fairness.

How They Work

Manages waiting processes in order.



5. TREE

Where Applied

Databases, file systems, AI.

Example Applications

Folder structures, database indexing.

Algorithms Used

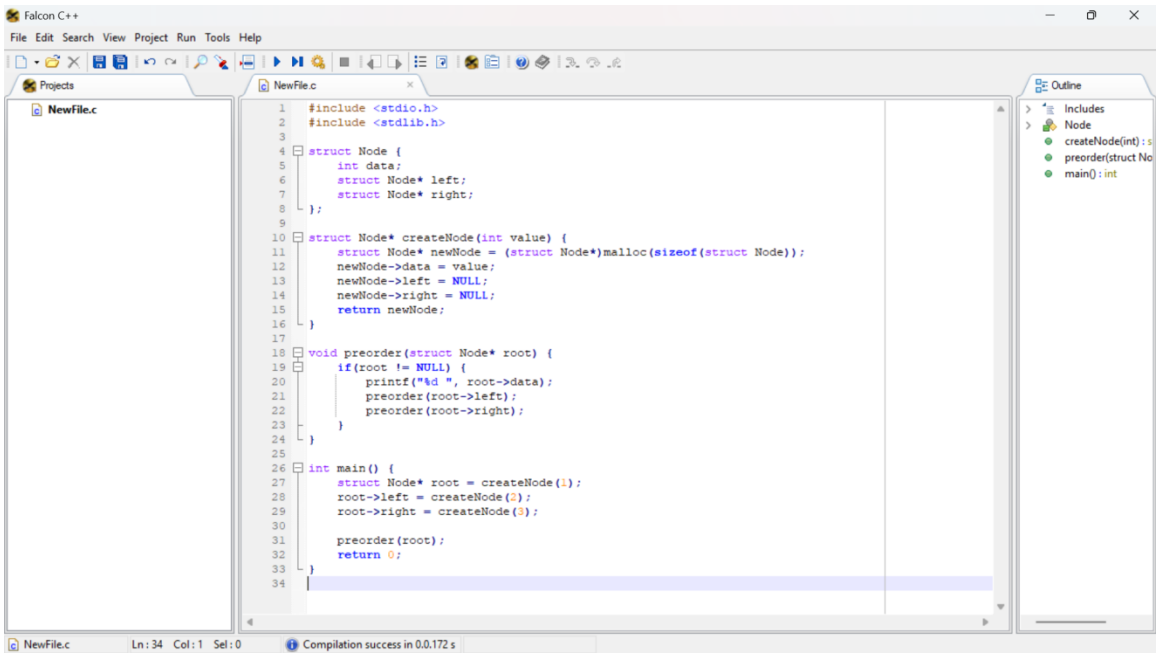
Tree traversal, searching.

Reasons

Hierarchical representation, fast searching.

How They Work

Balanced trees reduce disk access time.



6. GRAPH

Where Applied

Networks, social media, navigation.

Example Applications

Google Maps, social networks.

Algorithms Used

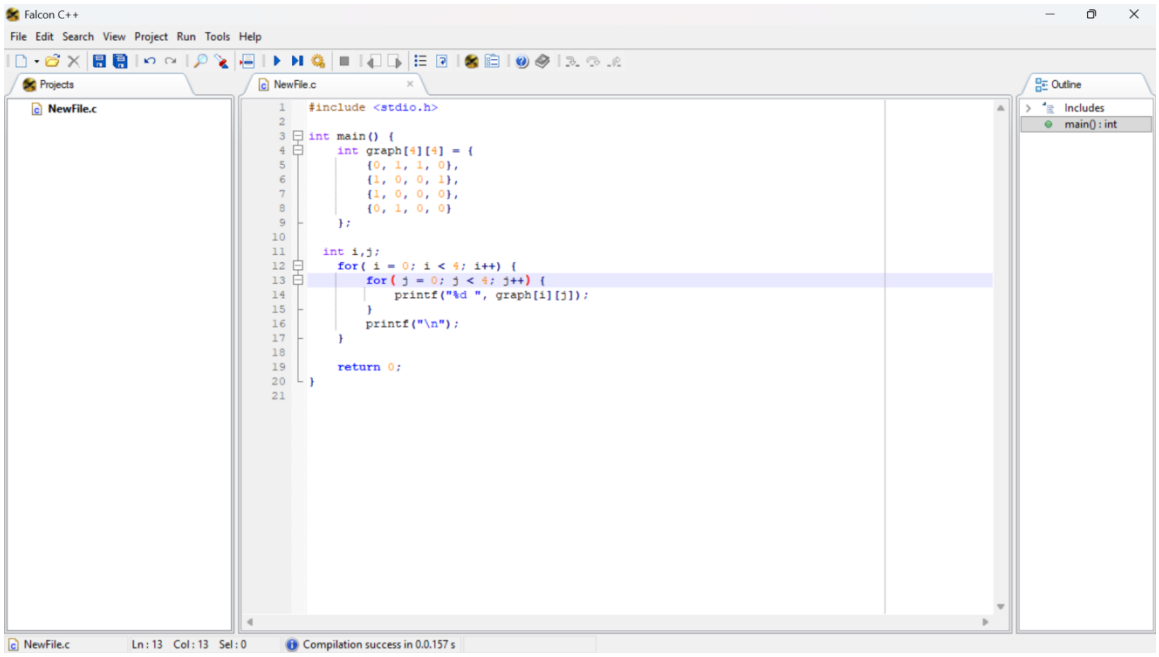
BFS, DFS, Dijkstra.

Reasons

Models relationships, efficient routing.

How They Work

Represents interconnected systems.


Falcon C++
File Edit Search View Project Run Tools Help
Projects
NewFile.c
NewFile.c
#include <stdio.h>
int main() {
int graph[4][4] = {
{0, 1, 1, 0},
{1, 0, 0, 1},
{1, 0, 0, 0},
{0, 1, 0, 0}
};
int i,j;
for(i = 0; i < 4; i++) {
for(j = 0; j < 4; j++) {
printf("%d ", graph[i][j]);
}
printf("\n");
}
return 0;
}
Outline
Includes
main() : int
NewFile.c
Ln : 13 Col : 13 Sel : 0
Compilation success in 0.0.157 s