Aim:

Create a binary tree and print inorder traversal.

Algorithm:

1. Create nodes with data.
2. Link nodes to form a tree.
3. Traverse left subtree, print node, traverse right subtree recursively.

Code:

```
#include <stdio.h>
#include <stdlib.h>

struct Node {
    int data;
    struct Node* left;
    struct Node* right;
};

struct Node* newNode(int data) {
    struct Node* node = (struct
Node*)malloc(sizeof(struct Node));
    node->data = data;
    node->left = node->right = NULL;
    return node;
}
```

```
void inorder(struct Node* root) {
    if (root != NULL) {
        inorder(root->left);
        printf("%d ", root->data);
        inorder(root->right);
    }
}

int main() {
    struct Node* root = newNode(1);
    root->left = newNode(2);
    root->right = newNode(3);
    root->left->left = newNode(4);
    root->left->right = newNode(5);

    printf("Inorder traversal: ");
    inorder(root);
    return 0;
}
```

Input:

Constructed tree:

```
       1
       / \
      2   3
     / \
    4   5
```

Output:

```
Inorder traversal: 4 2 5 1 3
```

Result:

Binary tree created and traversed in inorder successfully.