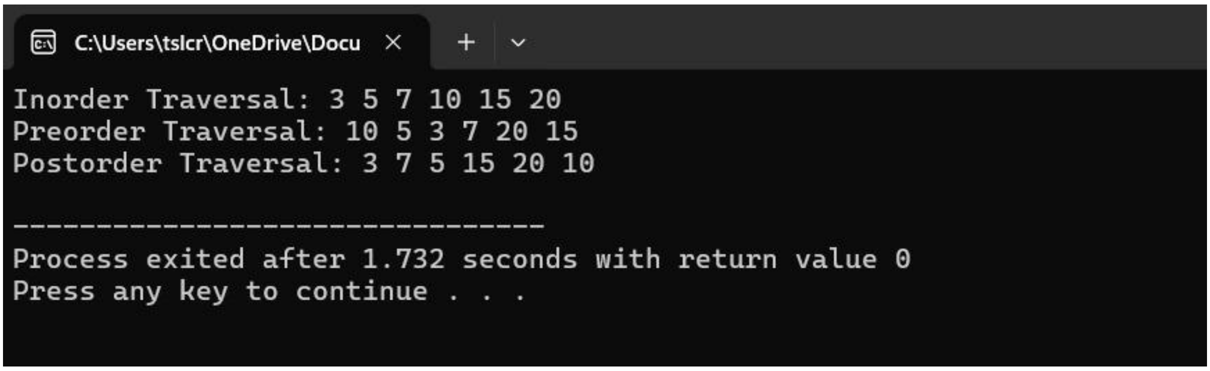
18 . Write a C program to implement the Tree Traversals (Inorder, Preorder, Postorder)

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

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

struct Node* create(int data)
{
  struct Node* temp = (struct Node*)malloc(sizeof(struct Node));
  temp->data = data;
  temp->left = temp->right = NULL;
  return temp;
}
void inorder(struct Node* root) {
  if (root) {
    inorder(root->left);
    printf("%d ", root->data);
    inorder(root->right);
  }
}
void preorder(struct Node* root) {
  if (root) {
    printf("%d ", root->data);
    preorder(root->left);
    preorder(root->right);
  }
}
void postorder(struct Node* root) {
  if (root) {
    postorder(root->left);
    postorder(root->right);
    printf("%d ", root->data);
  }
}
int main() {
  struct Node* root = create(10);
  root->left = create(5);
  root->right = create(20);
  root->left->left = create(3);
  root->left->right = create(7);
  root->right->left = create(15);

  printf("Inorder Traversal: ");
  inorder(root);
  printf("\n");

  printf("Preorder Traversal: ");
  preorder(root);
  printf("\n");

  printf("Postorder Traversal: ");
  postorder(root);
```

```
    printf("\n");


    return 0;
}
```

Output



```
Inorder Traversal: 3 5 7 10 15 20
Preorder Traversal: 10 5 3 7 20 15
Postorder Traversal: 3 7 5 15 20 10

--------------------------------
Process exited after 1.732 seconds with return value 0
Press any key to continue . . .
```