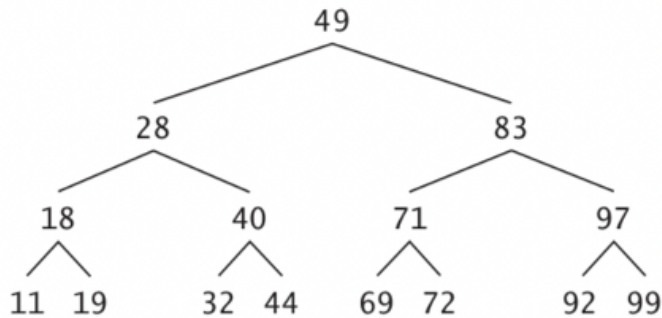


Lesson 16 Lab

Task 1: Provide the in-order, preorder and post traversals of the give binary tree



The inorder traversal is:

11 18 19 28 32 40 44 49 69 71 72 83 92 97 99
--

The preorder traversal is:

49 28 18 11 19 40 32 44 83 71 69 72 97 92 99
--

The postorder traversal is:

11 19 18 32 44 40 28 69 72 71 92 99 97 83 49
--

Task 2: Refer to our version of inOrder() and preOrder(), define your postOrder()

Prototype: `void postOrder(TreeNode* pNode);`

Same set up:

```

typedef struct TreeNode TreeNode;
struct TreeNode {
    int data;
    TreeNode* left;
    TreeNode* right;
};
  
```

<pre> void postOrder(TreeNode* pNode) { if (pNode != NULL) { postOrder(pNode->left); } </pre>
--

```

    postOrder(pNode->right);
    printf(" %d ", pNode->data);
}
}

```

Task 3: To define two programs:

The first **program** opens a file named “salutation.txt”, and then reads one line from user, writes that line into the file, close the file.

```

#include <stdio.h>

int main()
{
    const char filename[] = "salutation.txt";
    char content[100] = "";
    fgets(content, 100, stdin);
    FILE *fp = fopen(filename, "a");
    if (fp == NULL)
        return 1;
    fprintf(fp, "%s", content);
    fclose(fp);
    return 0;
}

```

The second **program** opens the “salutation.txt” reads the text from the file (should be one line) and then prints the text on screen(stdout).

```

#include <stdio.h>

int main()
{
    const char filename[] = "salutation.txt";
    char content[1000] = "";

    FILE *fp = fopen(filename, "r");
    if (fp == NULL)
        return 1;

```

```

while (fgets(content, sizeof(content), fp) != NULL)
{
    printf("%s", content);
}

fclose(fp);
return 0;
}

```

Task 4: To define a **program** writes the three given records into a file named “out1.txt”, using fwrite function

```

typedef struct record record;
struct record{
    char name[20];
    int age;
};

```

In main(), use: `record records[3] = {{"Sam", 25}, {"Tom", 30}, {"Kim", 16}};`

```

#include <stdio.h>

typedef struct record record;
struct record{
    char name[20];
    int age;
};

int main()
{
    record records[3] = {{"Sam", 25}, {"Tom", 30}, {"Kim",
16}};
    const char filename[] = "out1.txt";
    FILE *fp = fopen(filename, "w");
    if (fp == NULL)
        return 1;
    fwrite(records, sizeof(record), 3, fp);
    fclose(fp);
    return 0;
}

```

```
}

```

Task 5: Assume we have the file named “out1.txt” created by your Task4 program. Define a **program** using fwrite, fseek and fread functions changes Sam’s age to 26, and then changes Kim’s name to “Kimmy”

```
typedef struct record record;
struct record{
    char name[20];
    int age;
};
```

In Task 4, we had: `record records[3] = {{"Sam", 25}, {"Tom", 30}, {"Kim", 16}};`

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

typedef struct record record;
struct record{
    char name[20];
    int age;
};

int main()
{
    record records[3] = {0};
    const char filename[] = "out1.txt";
    FILE *fp = fopen(filename, "rb+");
    if (fp == NULL)
        return 1;
    fread(records, sizeof(record), 3, fp);
    records[0].age = 26;
    strcpy(records[2].name, "Kimmy");
    rewind(fp);
    fwrite(records, sizeof(record), 3, fp);
    fclose(fp);
    return 0;
}
```

--