Experiment 4

Write a C program that uses functions to perform the following:

a) Create a binary search tree of characters.

b) Traverse the above Binary search tree recursively in postorder.

Solution:

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

typedef struct NodeType
{
    char data;
    struct NodeType *left;
    struct NodeType *right;
} Node;

Node *create(char item)
{
    Node *node = (Node *)malloc(sizeof(Node *));
    node->data = item;
    node->left = node->right = NULL;
}

Node *insertion(Node *node, char item)
{
    if (node == NULL)
        return create(item);
```

```
    if (item < node->data)
        node->left = insertion(node->left, item);
    else
        node->right = insertion(node->right, item);

    return node;
}

void postorder(Node *node)
{
    if (node == NULL)
    {
        return;
    }

    postorder(node->left);
    postorder(node->right);
    printf("%c, ", node->data);
}

int main()
{
    Node *root = NULL;
    for (char i = 'A'; i <= 'E'; i++)
    {
        root = insertion(root, i);
    }
    postorder(root);
```

```
    return 0;
}
```

Output:

```
PS F:\DSA using C>  & 'c:\Users\Niladitya Sen\.vscode\extensions\ms-vscode.cpptools-1.17.5-win32-x64\debugAdapters\bin\WindowsDebu
gLauncher.exe' '--stdin=Microsoft-MIEngine-In-jvfa53hk.yxa' '--stdout=Microsoft-MIEngine-Out-1fttuct3.hgm' '--stderr=Microsoft-MIE
ngine-Error-vn1w5m4k.edp' '--pid=Microsoft-MIEngine-Pid-siotwiea.kwy' '--dbgExe=E:\C++ Compilers\mingw64\bin\gdb.exe' '--interpret
er=mi'
E, D, C, B, A,
PS F:\DSA using C>
```

Experiment 5

Write a C program that uses functions to perform the following:

a) Create a binary search tree of integers.

b) Traverse the above Binary search tree recursively in inorder.

Solution:

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

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

Node *create(int item)
{
    Node *node = (Node *)malloc(sizeof(Node *));
    node->data = item;
    node->left = node->right = NULL;
}

Node *insertion(Node *node, int item)
{
    if (node == NULL)
        return create(item);
```

```
    if (item < node->data)
        node->left = insertion(node->left, item);
    else
        node->right = insertion(node->right, item);

    return node;
}

void inorder(Node *node)
{
    if (node == NULL)
    {
        return;
    }

    inorder(node->left);
    printf("%d, ", node->data);
    inorder(node->right);
}

int main()
{
    Node *root = NULL;
    for (int i = 1; i <= 5; i++)
    {
        root = insertion(root, i);
    }
    inorder(root);
```

```
    return 0;
}
```

Output:

```
PS F:\DSA using C>  & 'c:\Users\Niladitya Sen\.vscode\extensions\ms-vscode.cpptools-1.17.5-win32-x64\debugAdapters\bin\WindowsDebu
gLauncher.exe' '--stdin=Microsoft-MIEngine-In-par5g4we.45m' '--stdout=Microsoft-MIEngine-Out-5t23aba2.w5x' '--stderr=Microsoft-MIE
ngine-Error-4wshfmf4.cqb' '--pid=Microsoft-MIEngine-Pid-ja2fny2w.ufa' '--dbgExe=E:\C++ Compilers\mingw64\bin\gdb.exe' '--interpret
er=mi'
1, 2, 3, 4, 5,
PS F:\DSA using C>
```

Experiment 6

Write C programs for implementing the following sorting methods to arrange a list of integers in ascending order:

a) Insertion sort

b) Merge sort.

Solution:

a) Insertion Sort

```
#include <stdio.h>
void insertionSort(int arr[], int length) {
    for (int i = 1; i < length; i++) {
        int key = arr[i];
        int j = i - 1;
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j--;
        }
        arr[j + 1] = key;
    }
}
int main() {
    int arr[] = {5, 2, 4, 6, 1, 3};
    int length = sizeof(arr) / sizeof(arr[0]);
    insertionSort(arr, length);
    for (int i = 0; i < length; i++) {
        printf("%d ", arr[i]);
    }
    return 0;
```

```
}
```

Output:

```
PS F:\DSA using C>  & 'c:\Users\Niladitya Sen\.vscode\extensions\ms-vscode.cpptools-1.17.5-win32-x64\debugAdapters\bin\WindowsDebu
gLauncher.exe' '--stdin=Microsoft-MIEngine-In-tpbfuhtm.u0b' '--stdout=Microsoft-MIEngine-Out-g0dg2yjl.xdq' '--stderr=Microsoft-MIE
ngine-Error-y1ik3j1i.tfo' '--pid=Microsoft-MIEngine-Pid-5ljbaers.sju' '--dbgExe=E:\C++ Compilers\mingw64\bin\gdb.exe' '--interpret
er=mi'
1 2 3 4 5 6
```

b) Merge Sort

```
#include <stdio.h>
void printArray(int arr[], int length);
void merge(int arr[], int left, int middle, int right);
void mergeSort(int arr[], int left, int right);
void mergeSort(int arr[], int left, int right) {
    if (left < right) {
        int middle = (left + right) / 2;
        mergeSort(arr, left, middle);
        mergeSort(arr, middle + 1, right);
        merge(arr, left, middle, right);
    }
}
void merge(int arr[], int left, int middle, int right) {
    int leftArrLen = middle - left + 1;
    int rightArrLen = right - middle;
    int leftArr[leftArrLen], rightArr[rightArrLen];
    for (int i = 0; i < leftArrLen; i++) {
        leftArr[i] = arr[left + i];
    }
    for (int i = 0; i < rightArrLen; i++) {
        rightArr[i] = arr[middle + 1 + i];
    }
    int i = 0, j = 0, k = left;
    while (i < leftArrLen && j < rightArrLen) {
        if (leftArr[i] <= rightArr[j]) {
            arr[k] = leftArr[i];
            i++;
```

```
            }
            else {
                arr[k] = rightArr[j];
                j++;
            }
            k++;
        }
        while (i < leftArrLen) {
            arr[k] = leftArr[i];
            i++;
            k++;
        }
        while (j < rightArrLen) {
            arr[k] = rightArr[j];
            j++;
            k++;
        }
    }
    void printArray(int arr[], int length) {
        for (int i = 0; i < length; i++) {
            printf("%d ", arr[i]);
        }
        printf("\n");
    }
    int main() {
        int arr[] = {5, 2, 4, 6, 1, 3};
        int length = sizeof(arr) / sizeof(arr[0]);
        mergeSort(arr, 0, length - 1);
```

```
    printArray(arr, length);
    return 0;
}
```

Output:

```
PS F:\DSA using C>  & 'c:\Users\Niladitya Sen\.vscode\extensions\ms-vscode.cpptools-1.17.5-win32-x64\debugAdapters\bin\WindowsDebu
gLauncher.exe' '--stdin=Microsoft-MIEngine-In-tpbfuhtm.u0b' '--stdout=Microsoft-MIEngine-Out-g0dg2yjl.xdq' '--stderr=Microsoft-MIE
ngine-Error-y1ik3j1i.tfo' '--pid=Microsoft-MIEngine-Pid-5ljbaers.sju' '--dbgExe=E:\C++ Compilers\mingw64\bin\gdb.exe' '--interpret
er=mi'
1 2 3 4 5 6
```

Experiment 7

Write C programs for implementing the following sorting methods to arrange a list of integers in ascending order:

a) Quick sort

b) Selection sort.

Solution:

a) Quick Sort

```
#include <stdio.h>
int partition(int arr[], int left, int right) {
    int pivot = arr[right];
    int i = left - 1;
    for (int j = left; j < right; j++) {
        if (arr[j] < pivot) {
            i++;
            int temp = arr[j];
            arr[j] = arr[i];
            arr[i] = temp;
        }
    }
    int temp = arr[right];
    arr[right] = arr[i + 1];
    arr[i + 1] = temp;
    return i + 1;
}
void quickSort(int arr[], int left, int right) {
    if (left < right) {
        int pivot = partition(arr, left, right);
```

```
        quickSort(arr, left, pivot - 1);
        quickSort(arr, pivot + 1, right);
    }
}
int main() {
    int arr[] = {5, 4, 3, 2, 1};
    int length = sizeof(arr) / sizeof(arr[0]);
    printf("Unsorted array: ");
    for (int i = 0; i < length; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
    quickSort(arr, 0, length - 1);
    printf("Sorted array: ");
    for (int i = 0; i < length; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
    return 0;
}
```

Output:

```
PS F:\DSA using C>  & 'c:\Users\Niladitya Sen\.vscode\extensions\ms-vscode.cpptools-1.17.5-win32-x64\debugAdapters\bin\WindowsDebu
gLauncher.exe' '--stdin=Microsoft-MIEngine-In-jrmzti0a.r4w' '--stdout=Microsoft-MIEngine-Out-sv50j5ke.gw4' '--stderr=Microsoft-MIE
ngine-Error-41cfh52f.vyk' '--pid=Microsoft-MIEngine-Pid-fadfv2v4.moo' '--dbgExe=E:\C++ Compilers\mingw64\bin\gdb.exe' '--interpret
er=mi'
Unsorted array: 5 4 3 2 1
Sorted array: 1 2 3 4 5
PS F:\DSA using C>
```

b) Selection Sort

```
#include <stdio.h>
void selectionSort(int arr[], int length) {
    for (int i = 0; i < length - 1; i++) {
        int minIdx = i;
        for (int j = i + 1; j < length; j++) {
            if (arr[j] < arr[minIdx]) {
                minIdx = j;
            }
        }
        int temp = arr[i];
        arr[i] = arr[minIdx];
        arr[minIdx] = temp;
    }
}
void printArray(int arr[], int length) {
    for (int i = 0; i < length; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}
int main() {
    int arr[] = { 5, 4, 3, 2, 1 };
    int length = sizeof(arr) / sizeof(arr[0]);
    printf("Original array: ");
    printArray(arr, length);
    selectionSort(arr, length);
    printf("Sorted array: ");
```

```
    printArray(arr, length);
    return 0;
}
```

Output:

```
PS F:\DSA using C>  & 'c:\Users\Niladitya Sen\.vscode\extensions\ms-vscode.cpptools-1.17.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-q0ihbrwd.wkc' '--stdout=Microsoft-MIEngine-Out-psaorqzy.sjr' '--stderr=Microsoft-MIEngine-Error-qhbskwit.4gd' '--pid=Microsoft-MIEngine-Pid-tlh10qpo.q3u' '--dbgExe=E:\C++ Compilers\mingw64\bin\gdb.exe' '--interpreter=mi'
Original array: 5 4 3 2 1
Sorted array: 1 2 3 4 5
PS F:\DSA using C>
```