**CSC221: DATA STRUCTURES & ALGORITHMS**

**BSCS 3*B***

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | LAB | | **09** | | Implementation of Recursion for the following problems |
|  |  |
| Objective(s): | Upon completion of this lab session, Student will be able to understand the following concepts |
| |  |  | | --- | --- | | 1 | Recursion meaning | | 2 | Recursive technique in programming | | 3 | How recursion works with stack | |  |  | |  |  | | |



**Submitted By:**

* Ahsan Ghaffar
* Reg: 48411
* BS(CS)\_3B

**Submitted to:**

* Miss Ambreen Akram(AA)

**Submission Date:**

[01/05/2018]

**DEPARTMENT OF COMPUTER SCIENCE**

**BAHRIA UNIVERSITY, KARACHI CAMPUS**

|  |
| --- |
| Lab Task(s): |
|
|  |
| |  |  | | --- | --- | | 1 | Write a program to calculate factorial of any given number by using recursion | | 2 | Write a program to calculate the Fibonacci series of any given number using recursion | | 3 | Implementation of Tower of Hanoi problem using recursion | | 4 | Implement Merge Sort with recursion | | |

**SOURCECODE:**

#include<iostream>

using namespace std;

void merge(int arr[], int l, int m, int r);

void mergeSort(int arr[], int l, int r);

void TOH(int n, char Sour, char Aux, char Des);

int fab(int val);

int fact(int val);

void mergeSort(int arr[], int l, int r) {

if (l < r) {

int m = l + (r - l) / 2;

mergeSort(arr, l, m);

mergeSort(arr, m + 1, r);

merge(arr, l, m, r);

}

}

void merge(int arr[], int l, int m, int r)

{

int i, j, k;

const int n1 = m - l + 1;

const int n2 = r - m;

int \* L = new int[n1];

int \* R = new int[n2];

for (i = 0; i < n1; i++)

L[i] = arr[l + i];

for (j = 0; j < n2; j++)

R[j] = arr[m + 1 + j];

i = 0,j = 0,k = l;

while (i < n1 && j < n2){

if (L[i] <= R[j]){

arr[k] = L[i];

i++;

}

else{

arr[k] = R[j];

j++;

}

k++;

}

while (i < n1){

arr[k] = L[i];

i++;

k++;

}

while (j < n2){

arr[k] = R[j];

j++;

k++;

}

}

//tower of HANOI function implementation

void TOH(int n, char Sour, char Aux, char Des)

{

if (n == 1){

cout << "Move Disk " << n << " from " << Sour << " to " << Des << endl;

return;

}

TOH(n - 1, Sour, Des, Aux);

cout << "Move Disk " << n << " from " << Sour << " to " << Des << endl;

TOH(n - 1, Aux, Sour, Des);

}

int fab(int val) {

if ((val == 1) || (val == 0))

return (val);

else {

return (fab(val - 1) + fab(val - 2));

}

}

int fact(int val)

{

if (val == 0)

return 1;

else{

return fact(val - 1)\*val;

}

}

int main()

{

int choice , val , arr[] = { 18, 3, 55, 14, 2, 22 };

int arr\_size = sizeof(arr) / sizeof(arr[0]);

char c;

do {

cout << "\n===========================" << endl;

cout << "1- MERGE SORT " << endl;

cout << "2- TOWER OF HANOI " << endl;

cout << "3- FIBONACCI SERIES " << endl;

cout << "4- FACTORIAL " << endl;

cout << "5- Exit " << endl;

cout << "===========================\n" << endl;

cout << "choose any one from the above options: ";

cin >> choice;

if (choice == 1) {

cout << "\nGiven array is \n";

for (int i = 0; i < arr\_size; i++)

cout << arr[i] << endl;

//sort array :

mergeSort(arr, 0, arr\_size - 1);

cout << "\nSorted array is \n";

for (int i = 0; i < arr\_size; i++)

cout << arr[i] << endl;

}

else if (choice == 2) {

cout << "\nEnter no. of disks: ";

cin >> val;

// calling the TOH function:

TOH(val, 'A', 'B', 'C');

}

else if (choice == 3) {

cout << "\nEnter Number for fibonacci series: ";

cin >> val;

int i = 0;

while (i < val){

cout << " " << fab(i);

i++;

}

}

else if (choice == 4) {

cout << "Enter Any Number to find its factorial :"l;

cin >> val;

cout << fact(val) << "\n"<< endl;

}

else if (choice == 5) {

exit(0);

}

cout << "\nPress 'y' to continue and ''n' to exit" << endl;

cin >> c;

} while (c == 'Y' || c == 'y');

system("pause");

return 0;

}

**SCREENSHOT:**

