LAB TASK – 3

1. Converting recursive programs to non-recursive programs. Towers of Hanoi Problem example.

a) Recursive Solution

Code :

#include <stdio.h>

void towerOfHanoi(int n, char src, char des, char aux)

{

if (n == 1)

{

printf("\n Move disk 1 from %c to %c",src,des);

return;

}

towerOfHanoi(n-1,src,aux,des);

printf("\n Move disk %d from %c to %c",n,src,des);

towerOfHanoi(n-1, aux,des,src);

}

int main()

{

int n;

printf("Enter the number of disks : ");

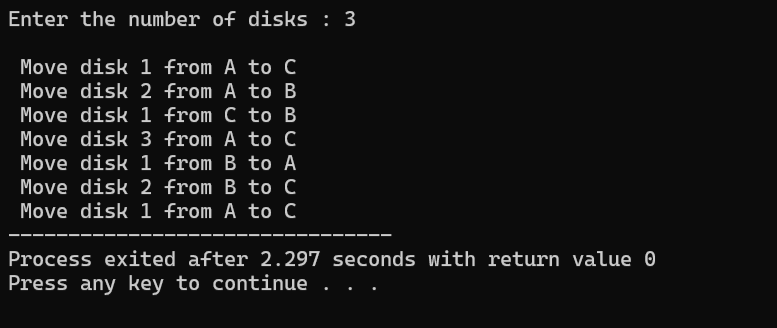
scanf("%d",&n);

towerOfHanoi(n, 'A', 'C', 'B');

return 0;

}

Output :



b) Non-Recursive Solution

Code :

#include <stdio.h>

#include <math.h>

void hanoi(int n, char src, char des, char aux){

int i;

int moves = pow(2, n)-1;

if (n%2==0) {

char temp = des;

des = aux;

aux = temp;

}

for (i=1; i <= moves; i++) {

if (i%3==1)

printf("Move disk between %c and %c \n", src, des);

else if (i%3==2)

printf("Move disk between %c and %c \n", src, aux);

else

printf("Move disk between %c and %c \n", aux, des);

}

}

int main(){

int n;

printf("Enter number of disks: ");

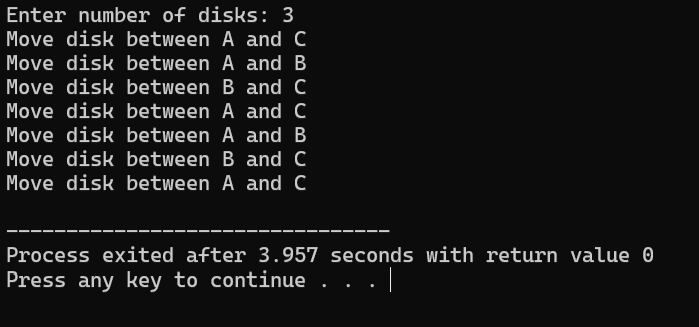
scanf("%d", &n);

hanoi (n, 'A', 'C', 'B');

return 0;

}

Output :



1. Implementation of Stack

Code :

#include <stdio.h>

#define MAX 100

int stack[MAX];

int top = -1;

int isFull() {

return top == MAX - 1;

}

int isEmpty() {

return top == -1;

}

void push(int value) {

if (isFull()) {

printf("Stack Overflow\n");

} else {

stack[++top] = value;

printf("%d pushed\n", value);

}

}

void pop() {

if (isEmpty()) {

printf("Stack Underflow\n");

} else {

printf("%d popped\n", stack[top--]);

}

}

void peek() {

if (isEmpty()) {

printf("Stack is empty\n");

} else {

printf("Top element: %d\n", stack[top]);

}

}

void display() {

int i;

if (isEmpty()) {

printf("Stack is empty\n");

} else {

printf("Stack elements: ");

for (i = top; i >= 0; i--) {

printf("%d ", stack[i]);

}

printf("\n");

}

}

int main() {

printf("Enter 5 elements into the stack");

int a[5],i;

for (i=0;i<5;i++){

scanf("%d",&a[i]);

}

for (i=0;i<5;i++){

push(a[i]);

}

display();

peek();

pop();

display();

peek();

return 0;

}

Output :

