

5) / *Design, Develop and Implement a Program in C for the following Stack Applications

a. Evaluation of postfix expression with single digit operands and operators: +, -, *, /, %, ^ or \$

b. Solving Tower of Hanoi problem with n disks */

```
#include <stdio.h>
#include <math.h>
#include <ctype.h>

float evaluatePostfix(char postfix[]);
float oper(char symb, float op1, float op2);
void push(float x);
float pop();
float stack[50];
int top = -1;
int main() {
char postfix[50];
int choice;
float res;
do {
printf("Enter postfix expression: ");
scanf("%s", postfix);
res = evaluatePostfix(postfix);
printf("Result = %.2f\n", res);
printf("Do you want to enter another expression? (1 for Yes / 0 for No): ");
scanf("%d", &choice);
} while (choice != 0);
return 0;
}
/* Function to evaluate a postfix expression */
float evaluatePostfix(char postfix[]) {
float op1, op2, res;
char ch;
int i = 0;
while ((ch = postfix[i]) != '\0') {
if (isdigit(ch)) {
push(ch - '0'); // Convert character digit to float and push onto stack
} else {
op2 = pop();
op1 = pop();
res = oper(ch, op1, op2);
push(res);
}
i++;
}
return pop(); // Final result will be the only item left on the stack
}
/* Function to perform the operation based on the operator */
float oper(char symb, float op1, float op2) {
switch (symb) {
case '$': // Fall-through
case '^': return pow(op1, op2);
case '*': return op1 * op2;
case '/': return op1 / op2;
```

```

case '+': return op1 + op2;
case '-': return op1 - op2;
default:
printf("Invalid operator\n");
return 0; // Return 0 for invalid operator
}
}
/* Function to push an element onto the stack */
void push(float x) {
if (top < 49) { // Check to prevent stack overflow
stack[++top] = x;
} else {
printf("Stack Overflow\n");
}
}
/* Function to pop an element from the stack */
float pop() {
if (top >= 0) { // Check to prevent stack underflow
return stack[top--];
} else {
printf("Stack Underflow\n");
return 0; // Return 0 or another error indicator if stack is empty
}
}
}

```

```

/*Sample output
Enter postfix expression
623+-382/+*2$3+
Result=52.000000
*/

```

/* Design, Develop and Implement a Program in C for the following Stack Applications
b. Solving Tower of Hanoi problem with n disks

```

*/

```

```

#include<stdio.h>

```

```

void towers(int n, char src, char dest, char aux)
{
if(n==1)
{
printf("MOVE DISK 1 FROM PEG %c TO PEG %c\n",src,dest);
return;
}
towers(n-1,src,aux,dest);
printf("MOVE DISK %d FROM PEG %c TO PEG %c\n",n,src,dest);
towers(n-1,aux,dest,src);
}
}

```

```
void main()
{
    int n;
    printf("ENTER THE NUMBER OF DISKS\n");
    scanf("%d",&n);
    printf("MOVES MADE\n");
    towers(n,'A','C','B');
}
```

/*Sample output

ENTER THE NUMBER OF DISKS

3

MOVES MADE

MOVE DISK 1 FROM PEG A TO PEG C

MOVE DISK 2 FROM PEG A TO PEG B

MOVE DISK 1 FROM PEG C TO PEG B

MOVE DISK 3 FROM PEG A TO PEG C

MOVE DISK 1 FROM PEG B TO PEG A

MOVE DISK 2 FROM PEG B TO PEG C

MOVE DISK 1 FROM PEG A TO PEG C

*/