

5) Develop a Program in C for the following Stack Applications

- a. Evaluation of Suffix expression with single digit operands and operators: +, -, *, /, %, ^
- b. Solving Tower of Hanoi problem with n disks

a) Evaluation of Suffix expression with single digit operands and operators: +, -, *, /, %, ^

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

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

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

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

```
int i, top = -1;
```

```
int op1, op2, res, s[20];
```

```
char postfix[90], symb;
```

```
void push();
```

```
int pop();
```

```
void main() {
```

```
printf("\nEnter a valid postfix expression:\n");
```

```
scanf("%s", postfix);
```

```
for (i = 0; postfix[i] != '\0'; i++) {
```

```
    symb = postfix[i];
```

```
    if (isdigit(symb)) {
```

```
        push(symb - '0');
```

```
    }
```

```
    else {
```

```
        op2 = pop();
```

```
        op1 = pop();
```

```
        switch (symb) {
```

```
            case '+':
```

```
                push(op1 + op2);
```

```
break;
case '-':
push(op1 - op2);
break;
case '*':
push(op1 * op2);
break;
case '/':
push(op1 / op2);
break;
case '%':
push(op1 % op2);
break;
case '$':
case '^':
push(pow(op1, op2));
break;
default:
push(0);
}
}
}
res = pop();
printf("\n Result = %d", res);
}
```

```
void push(int item) {
top = top + 1;
s[top] = item;
}
```

```

int pop() {
int item;

item = s[top];

top = top - 1;

return item;

}

```

b) Solving Tower of Hanoi problem with n disks

```

#include <stdio.h>

#include <math.h>

void tower(int n, int source, int temp, int destination) {
if (n == 0)
return;

tower(n - 1, source, destination, temp);

printf("\nMove disc %d from %c to %c", n, source, destination);

tower(n - 1, temp, source, destination);

}

void main() {
int n;

printf("\nEnter the number of discs: \n");

scanf("%d", & n);

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

printf("\n\nTotal Number of moves are: %d", (int) pow(2, n) - 1);

}

```