

DEPARTMENT : INFORMATION SCIENCE
AND
ENGINEERING

STUDENT NAME : R.G. MANYVITHA

STUDENT VSN : JRN19IS111

SEM : III

SEC : B

DATE : 21/10/2020

TIME : 9.00 AM - 11.00 AM

COURSE NAME : DATA STRUCTURE LABORATORY
WITH

CODE : 18CSL38

BATCH : 6

SIGNATURE : R.G. Manyvitha

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

Evaluation of suffix expression with single digit operands and operators +, -, *, /, %, ^

b) Solving Tower of Hanoi problem with n-disks.

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

#define max 25

int stack[max], top = -1;
void push()
{
    stack[++top] = pterm;
}
int pop()
{
    return (stack[top--]);
}
void main()
{
    char suffix[100];
    int i=0, temp;
    printf("Enter suffix expression = ");
    gets(suffix);
}
```

scans ("10%1", suffix);

for (i = 0; suffix[i] != '10'; i++)

if (isDigit(suffix[i]))

push (suffix[i] - '0');

switch (suffix[i])

{

case '+': push (pop() + pop());

break;

case '-': temp = pop()

push (pop() - temp);

break;

case '*': push (pop() * pop());

break;

case '/': temp = pop();

push (pop() / temp);

break;

case '^': temp = pop();

push (pow (pop(), temp));

break

}

```
printf(" Value of expression = '%d' , pop());
}
```

5b) Tower of Hanoi \Rightarrow

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

```
void tower (int, char, char, char);
int main()
```

```
int n;
```

```
printf(" Enter number of disks : ");
scanf("%d", &n);
```

```
printf(" sequence of moves : [n]");
tower (n, 'A', 'C', 'B');
```

```
getch();
```

```
return 0;
```

```
}
```

```
void tower (int n, char s, char a,
            char d)
```

```
{
```

```
if (n == 1)
```

```
printf(" Move disk 1 from
peg %c to peg %c , s, d);
return; }
```

tower (n - 1, b, a, d);

finishes " in Move disk 'd' from peg 'c'

to peg 'c' (n, b, a, d);

tower (n - 1, a, d, s);

}

OUTPUT 5a: Enter suffix expression =

3 4 5 * 6 / +

Value of the expression is 6

Symbol	Stack
3	3
4	3, 4
5	3, 4, 5
*	3, 20
6	3, 20, 6
/	3, 3
+	6

Enter suffix expression = 6 5 2 3 + 7 * + 2 + *

Value of the expression is 252

Symbol	Stack
6	6
5	6, 5
2	6, 5, 2
3	6, 5, 2, 3
+	6, 5, 5

*	6, 5, 5, 7
*	6, 5, 35
+	6, 40
2	6, 40, 2
+	6, 42
*	252

Enter suffix expression = 231 * + 9 -
Stack

2	2
3	2, 3
1	2, 3, 1
*	2, 3
+	5
9	5, 9
-	- 4

Enter suffix expression = 1234 ^ 2 - 654 * + % * + 4

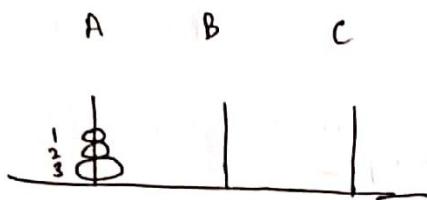
Value of expression = 14

OUTPUT 5 by Enter the number of disks : 3

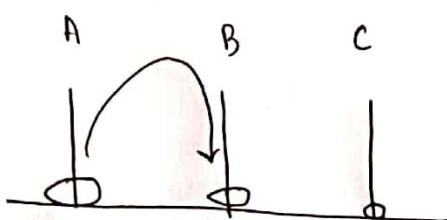
Sequence of moves :

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

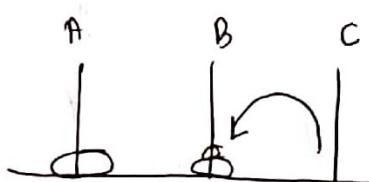
Move disk 1 from pig B to pig A
 Move disk 2 from pig B to pig C
 Move disk 1 from pig A to pig C
 ...



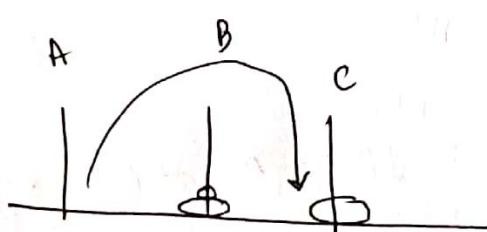
STEP 1



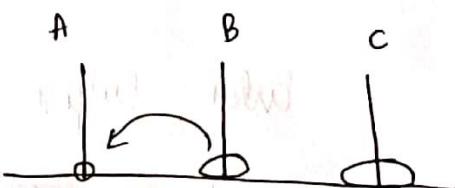
STEP 2



STEP 3



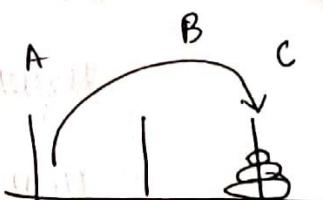
STEP 4



STEP 5



STEP 6

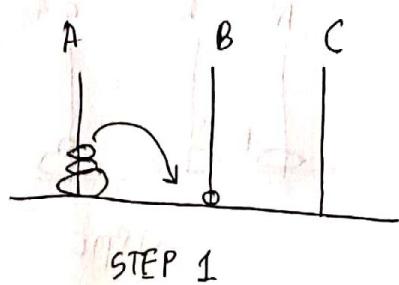
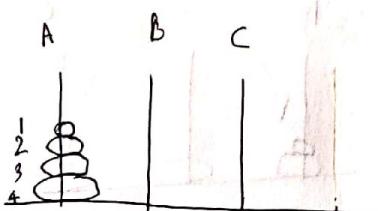


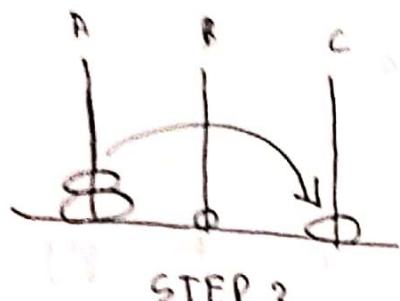
STEP 7

OUTPUT2: Enter the number of disks : 4

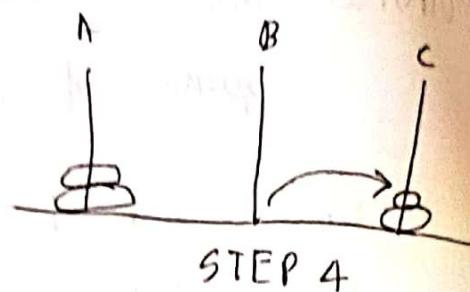
sequence of moves :

Move	disk 1	from	peg A	to	peg B
Move	disk 2	from	peg A	to	peg C
Move	disk 1	from	peg B	to	peg C
Move	disk 3	from	peg A	to	peg B
Move	disk 1	from	peg C	to	peg A
Move	disk 2	from	peg C	to	peg B
Move	disk 1	from	peg A	to	peg B
Move	disk 4	from	peg A	to	peg C
Move	disk 1	from	peg B	to	peg C
Move	disk 2	from	peg B	to	peg A
Move	disk 1	from	peg C	to	peg A
Move	disk 3	from	peg B	to	peg C
Move	disk 1	from	peg A	to	peg B
Move	disk 2	from	peg A	to	peg C
Move	disk 1	from	peg B	to	peg C

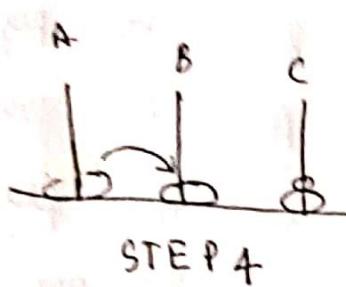




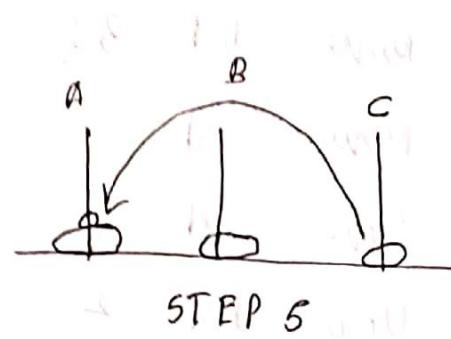
STEP 3



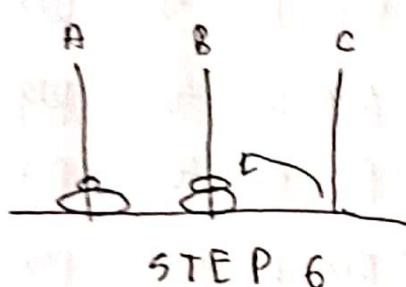
STEP 4



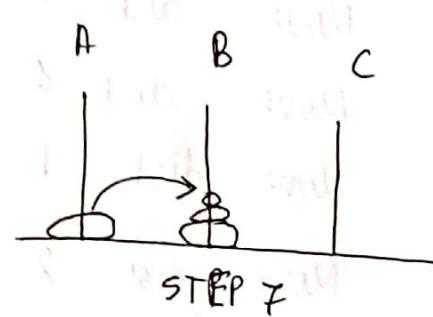
STEP 4



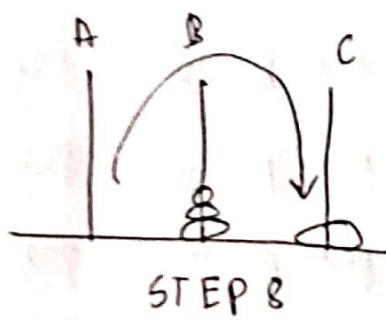
STEP 5



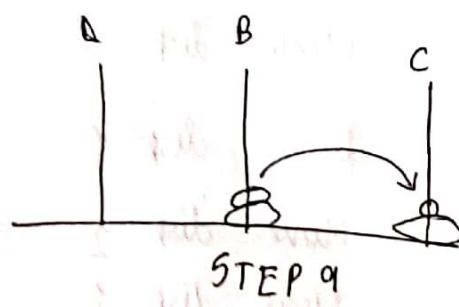
STEP 6



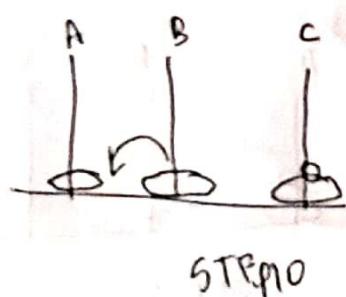
STEP 7



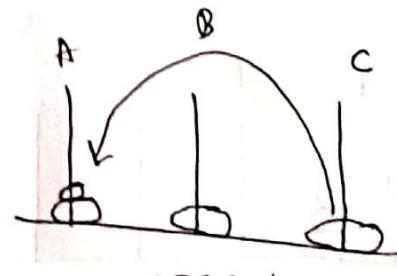
STEP 8



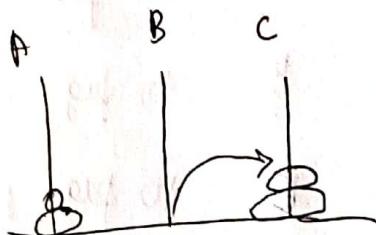
STEP 9



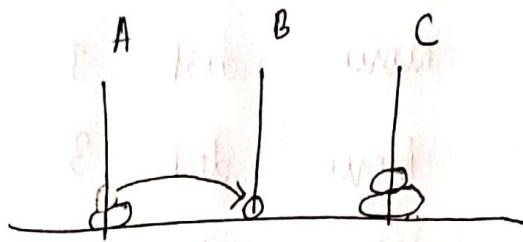
STEP 10



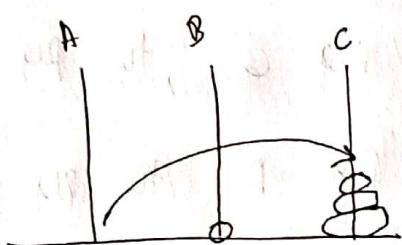
STEP 11



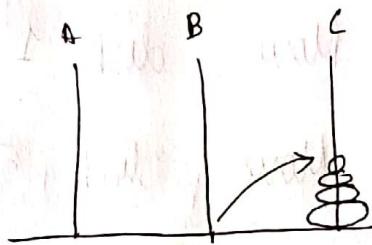
STEP 12



STEP 13



STEP 14



STEP 15

Enter the output number of disks : 5

Sequence of moves :

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
Move	disk 4	from	peg	A	to	peg	B
Move	disk 1	from	peg	C	to	peg	B

- Move disk 2 from peg C to peg A
Move disk 1 from peg B to peg A
Move disk 3 from peg E to peg B
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 5 from peg A to peg C
Move disk 1 from peg AB to peg A
Move disk 2 from peg B to peg C
Move disk 1 from peg A to peg C
Move disk 3 from peg B to peg A
Move disk 1 from peg C to peg B
Move disk 2 from peg C to peg A
Move disk 1 from peg B to peg A
Move disk 4 from peg B to peg C
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 pig A to pig C
 move disk 1 from pig B to pig A
 move disk 2 from pig B to pig C
 move disk 1 from pig A to pig C

Steps for evaluating postfix expression:

- 1 > Scan the symbol from left to right
- 2 > If the scanned - symbol is an operand , push it on the stack .
- 3 > If the scanned - symbol is an operator , pop 2 operands from the stack . The first popped operand acts as operand 2 and second popped operand act as operand 1 . Now perform the indicated operation and Push the result on to the stack .
- 4 > Repeat the above procedure till the end of input is encountered .

```
define max 35
nt stack[max], top = -1;
```

```
void push (int ele)
```

input

```
enter suffix expression =
6523+7*+2+*
```

```
value of the expression is 252
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```

```
include<stdio.h>
include<ctype.h>
include<math.h>
define max 35
int stack[max], top = -1;
```

```
void push (int ele)
```

input

```
enter suffix expression =
345*6/+  
value of the expression is 6
```

...Program finished with exit code 0

Press ENTER to exit console.

input

enter suffix expression =

231*+9-

value of the expression is -4

...Program finished with exit code 0

Press ENTER to exit console.

input

enter suffix expression =

123462-654*+%*+4

value of the expression is 4

... Program finished with exit code 0

Press ENTER to exit console.

```
4 int n;
5 printf("Enter the number of disks : ");
6 scanf("%d", &n);
7 printf(" sequence of moves involved in the Tower of Hanoi : \n");
8 tower(n, 'A', 'C', 'B');
9 getch();
10 return 0;
11 }
```

input

```
main.c:9:1: warning: implicit declaration of function 'getch' [-W
implicit-function-declaration]
Enter the number of disks : 3
sequence of moves involved in the Tower of Hanoi :

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

... Program finished with exit code 0
Press ENTER to exit console.
```

```
4 int n;
5 printf("Enter the number of disks : ");
6 scanf("%d", &n);
7 printf(" sequence of moves involved in the Tower of Hanoi :\n");
8 tower(n, 'A', 'C', 'B');
9 getch();
10 return 0;
11 }
```

```
input
main.c:9:1: warning: implicit declaration of function 'getch' [-Wimplicit-function-declaration]
Enter the number of disks : 4
sequence of moves involved in the Tower of Hanoi :

Move disk 1 from peg A to peg B
Move disk 2 from peg A to peg C
Move disk 1 from peg B to peg C
Move disk 3 from peg A to peg B
Move disk 1 from peg C to peg A
Move disk 2 from peg C to peg B
Move disk 1 from peg A to peg B
Move disk 4 from peg A to peg C
Move disk 1 from peg B to peg C
Move disk 2 from peg B to peg A
Move disk 1 from peg C to peg A
Move disk 3 from peg B to peg C
Move disk 1 from peg A to peg B
Move disk 2 from peg A to peg C
Move disk 1 from peg B to peg C

...Program finished with exit code 0
Press ENTER to exit console.
```

Run Debug Stop Share Save Beautify

Language C

input

main.c:9:1: warning: implicit declaration of function 'getch' [-Wimplicit-function-declaration]

Enter the number of disks : 5

sequence of moves involved in the Tower of Hanoi :

```
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
Move disk 4 from peg A to peg B
Move disk 1 from peg C to peg B
Move disk 2 from peg C to peg A
Move disk 1 from peg B to peg A
Move disk 3 from peg C to peg B
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 5 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
Move disk 3 from peg B to peg A
Move disk 1 from peg C to peg B
Move disk 2 from peg C to peg A
Move disk 1 from peg B to peg A
Move disk 4 from peg B to peg C
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 C to peg A
Move disk 1 from peg B to peg A
Move disk 4 from peg B to peg C
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 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
Move disk 4 from peg A to peg B
Move disk 1 from peg C to peg B
Move disk 2 from peg C to peg A
Move disk 1 from peg B to peg A
Move disk 3 from peg C to peg B
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 5 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
Move disk 3 from peg B to peg A
Move disk 1 from peg C to peg B
Move disk 2 from peg C to peg A
Move disk 1 from peg B to peg A
Move disk 4 from peg B to peg C
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
```

...Program finished with exit code 0

Press ENTER to exit console.

