```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXSIZE 100

typedef union
{
    float fData;
    char  cData;
} Item;


typedef struct
{
    Item items[MAXSIZE];
    int top;
}Stack;

int Priority (char *c)
{
    if(c=='^')return 3;
    else  if (c=='*' || c=='/')return 2;
    else  if (c=='+' || c=='-')return 1;
    else return 0;
}
Stack* initialize()
{
    Stack *s = malloc(sizeof (Stack));
    s->top=0;
    return s;
}


Item pop (Stack *s)
{
return s->items[--s->top];
}


void push(Stack *s,Item value)
{
    s->items[s->top++]=value;
}


Item top (Stack *s)
{
    return s->items[--s->top];
}


int isEmpty(Stack *s)
{
    if(s->top==0) return 1;
    else return 0;
}

void infixTopostfix(char *infix, char* postfix)
{
    int i,j=0;
    Item x;
    Stack *p = initialize();
    *postfix="";
    for(i=0;i<strlen(infix);i++)
    {
        if(infix[i]==' ')continue ;
    if(isdigit(infix[i]) || isalpha(infix[i])){
     strncat(postfix,infix[i],1);
    }
    else if (infix[i]=='(')
    {
      x.cData=infix[i];
      push(p,x);
    }
     else if (infix[i]==')')
     {
       while (top(p).cData!='(')
         strncat(postfix ,  pop(p).cData,1);
    pop(p);
     }

     else
     {
       while (!isEmpty(p) && (Priority(infix[i]) <= Priority(top(p).cData)))
         strncat(postfix, pop(p).cData,1);

           x.cData = infix[i];
           push(p,x);

     }

  }

}
int main()
{
Stack *s = initialize();
char *infix = (char*)malloc(100);
strcpy(infix,"1 + 2 + 3 * 4");
 char *postfix = (char*)malloc(100);
 infixTopostfix(infix,postfix);
 printf("%s",postfix);
 return 0;
 }
```