```
function createnode(char chh)
   struct ss* neww=NULL
   neww->c=chh;
   neww->nxt=NULL
   if(chh=='*' or chh=='/')
           neww->val=2

   else if(chh=='+' or chh=='-')
           neww->val=1

   else if(chh=='(')
          neww->val=3

   else if(chh==')')
          neww->val=0

   return neww




function topostfix(char ar[10000],int n)

   char w[10000],z[10000]
   struct ss *stack1=NULL,*zz=NULL
   struct lll *a=NULL

   a->top=NULL
```

```
int j=0
for(int i=0 to n)
   if(isalpha(ar[i])){
      w[j]=ar[i]
      j++
   }
   else {
      zz=createnode(ar[i])
      while(1){
      if(a->top!=NULL){

         if(a->top->val<zz->val){
            zz->nxt=a->top
            a->top=zz
            if(zz->c=='('){
               zz->val=-1

            }
            break
         }
         else{
            w[j]=a->top->c
            a->top=a->top->nxt
            j++
         }

      }
      else{
         a->top=zz
          if(zz->c=='('){
```

```
                zz->val=-1


              }
            break
        }


        }
        if(a->top->c==')'){
            a->top=a->top->nxt->nxt
        }


      }
    }
    while(a->top!=NULL){
      w[j]=a->top->c
      j++
      a->top=a->top->nxt
    }


    char *re=w
    return re




function createnode1(char ch,struct s* a,struct s* b)
    struct s *sel=NULL
    sel->key=ch
    sel->lchild=a
    sel->rchild=b
    sel->nxt=NULL
```

```
    return sel


function push(struct ll *a,struct s *node)

        node->nxt=a->top
        a->top=node









function maketree(char ar[],int n)



    a->top=NULL
    for(int i=0 to n){
        if(isalpha(ar[i])){
            node=createnode1(ar[i],NULL,NULL)
            push(a,node)
        }
        else{
            struct s* sele=NULL


            sele->rchild=a->top
            a->top=a->top->nxt
            sele->lchild=a->top
            a->top=a->top->nxt
            node=createnode1(ar[i],sele->lchild,sele->rchild)
```

```
            push(a,node)
        }

    }
    return a->top




function postorder(struct s* node)
    if(node==NULL){
        return
    }
    postorder(node->lchild)
    postorder(node->rchild)
    print(node->key)



function main()
   char ar[10000],br[10000]
   read ar
   int n=strlen(ar)
    strncpy(br,ar+2,n-3)
    char *ar2=topostfix(br,n-3)
    struct s *troot=NULL,*kingroot=NULL,*left=NULL
    troot=maketree(ar,n)
    left=createnode1(ar[0],NULL,NULL)
    kingroot=createnode1(ar[1],left,troot)
    postorder(kingroot)
```