```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int ret[100];
static int pos=0;
static int sc=0;
void nfa(int st,int p,char *s)
{
    int i,sp,fs[15],fsc=0;
    sp=st;pos=p;sc=st;
    while(*s != NULL)
    {if(isalpha(*s))
        {ret[pos++]=sp;
        ret[pos++]=*s;
        ret[pos++]=++sc;}
    if(*s=='.')
        {sp=sc;
         ret[pos++]=sc;
         ret[pos++]=238;
         ret[pos++]=++sc;
         sp=sc;}
    if(*s=='+')
        {sp=st;
         fs[fsc++]=sc;}
    if(*s=='*')
        {ret[pos++]=sc;
         ret[pos++]=238;
         ret[pos++]=sp;
         ret[pos++]=sp;
         ret[pos++]=238;
         ret[pos++]=sc;
         }
     if (*s=='(')
        {char ps[50];
         int i=0,flag=1;
         s++;
           while(flag!=0)
            {ps[i++]=*s;
              if (*s=='(')
                  flag++;
              if (*s==')')
                  flag--;
```

```c
            s++;}
            ps[--i]='\0';
            nfa(sc,pos,ps);
            s--;
        }
     s++;
    }
    sc++;
      for(i=0;i<fsc;i++)
        {ret[pos++]=fs[i];
         ret[pos++]=238;
         ret[pos++]=sc;
        }
        ret[pos++]=sc-1;
        ret[pos++]=238;
        ret[pos++]=sc;
}
int main()
{    int i;
    char *inp;
    printf("Enter the regular expression : ");
    gets(inp);
    nfa(1,0,inp);

    printf("\nCurrent State  Input  Next State\n\n");
    for(i=0;i<pos;i=i+3)
        printf("\t%d      --%c-->   \t%d\n",ret[i],ret[i+1],ret[i+2]);
    printf("\n");

    return 0;

}
```

```
Enter the regular expression : a.b

Current State   Input   Next State

      1        --a-->      2
      2        --ε-->      3
      3        --b-->      4
      4        --ε-->      5
```

Enter the regular expression : a.b.(a+b)*

Current State   Input   Next State

        1   --a-->   2
        2   --ε-->   3
        3   --b-->   4
        4   --ε-->   5
        5   --a-->   6
        5   --b-->   7
        6   --ε-->   8
        7   --ε-->   8
        8   --ε-->   5
        5   --ε-->   8
        8   --ε-->   9

Enter the regular expression : (a+b)*a*b.b

Current State   Input   Next State

Current State	Input	Next State
1	--a-->	2
1	--b-->	3
2	--ε-->	4
3	--ε-->	4
4	--ε-->	1
1	--ε-->	4
1	--a-->	5
5	--ε-->	1
1	--ε-->	5
1	--b-->	6
6	--ε-->	7
7	--b-->	8
8	--ε-->	9