

26. Write a yacc program to check the acceptance of the language $a^n.b^n$ where $n \geq 0$.

```
//lex
%{
#include "y.tab.h"
%}

%%
[aA] {return A;}
[bB] {return B;}
\n {return NL;}
. {return yytext[0];}
%%

//yacc
%{
#include<stdio.h>
#include<stdlib.h>
%}

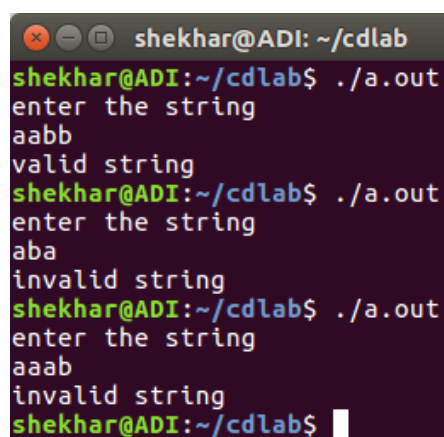
%token A B NL

%%
stmt: S NL {printf("valid string\n");
           exit(0);}
;
S: A S B |
;
%%

int yyerror(char *msg)
{
printf("invalid string\n");
exit(0);
}

main()
{
printf("enter the string\n");
yyparse();
}
```

OUTPUT:



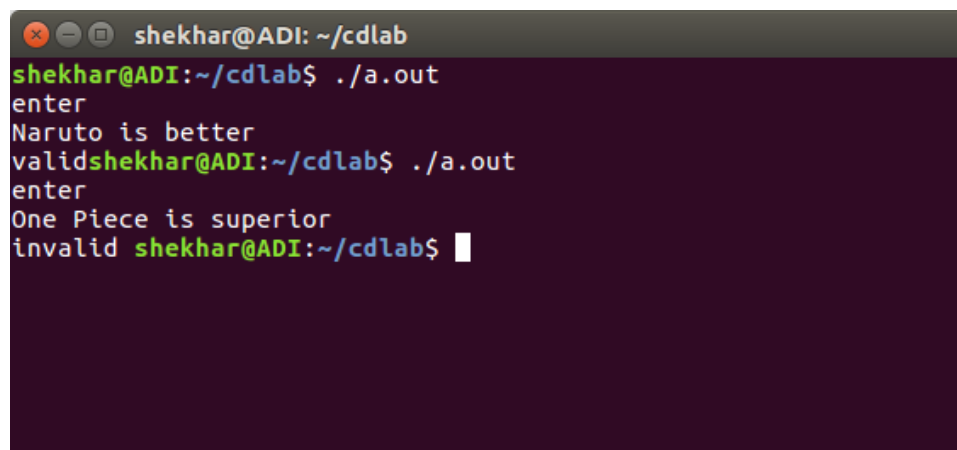
```
shekhar@ADI: ~/cdlab
shekhar@ADI:~/cdlab$ ./a.out
enter the string
aabb
valid string
shekhar@ADI:~/cdlab$ ./a.out
enter the string
aba
invalid string
shekhar@ADI:~/cdlab$ ./a.out
enter the string
aaab
invalid string
shekhar@ADI:~/cdlab$
```

27. Implement a sentence parser using yacc for recognizing English sentences.

```
//lex
%{
#include "y.tab.h"
%}
%%
Naruto|DBZ|DeathNote {return E;}
is {return V;}
better|superior {return O; }
[ \t] {;}
\n {return 0;}
. {return yytext[0];}
%%

//yacc
%{
#include<stdio.h>
%}
%token E V O
%%
stmt: S
    ;
S: E S V O
    |
    ;
%%
void main()
{
printf("enter \n");
yyparse();
printf("valid");
exit(0);
}
void yyerror()
{
printf("invalid ");
exit(0);
}
```

OUTPUT:



```
shekhar@ADI: ~/cdlab
shekhar@ADI:~/cdlab$ ./a.out
enter
Naruto is better
validshekhar@ADI:~/cdlab$ ./a.out
enter
One Piece is superior
invalid shekhar@ADI:~/cdlab$
```

28. Write a program in yacc to implement binary to decimal conversion.

```
//lex
%{
#include<stdio.h>
#include<stdlib.h>
#include"y.tab.h"
extern int yylval;
%}
%%

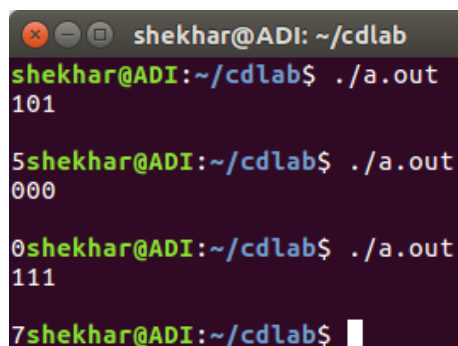
[ \t] {;}
\n return 0;
. return yytext[0];
%%

//yacc
%{
#include<stdio.h>
#include<stdlib.h>
void yyerror(char *s);
%}
%token ZERO ONE
%%

N: L {printf("\n%d", $$);}
L: L B { $$=$1*2+$2;}
| B { $$=$1;}
B:ZERO { $$=$1;}
|ONE { $$=$1;}
%%

int main()
{
while(yyparse());
}
yyerror(char *s)
{
fprintf(stdout, "\n%s", s);
}
```

OUTPUT:



A terminal window with a dark purple background. The title bar shows 'shekhar@ADI: ~/cdlab'. The prompt is 'shekhar@ADI:~/cdlab\$'. The first command is './a.out' followed by the output '101'. The second command is '5shekhar@ADI:~/cdlab\$./a.out' followed by the output '000'. The third command is '0shekhar@ADI:~/cdlab\$./a.out' followed by the output '111'. The fourth command is '7shekhar@ADI:~/cdlab\$' followed by a cursor.

```
shekhar@ADI: ~/cdlab
shekhar@ADI:~/cdlab$ ./a.out
101

5shekhar@ADI:~/cdlab$ ./a.out
000

0shekhar@ADI:~/cdlab$ ./a.out
111

7shekhar@ADI:~/cdlab$
```