

CD Assignment 3

Name: Yash Oswal

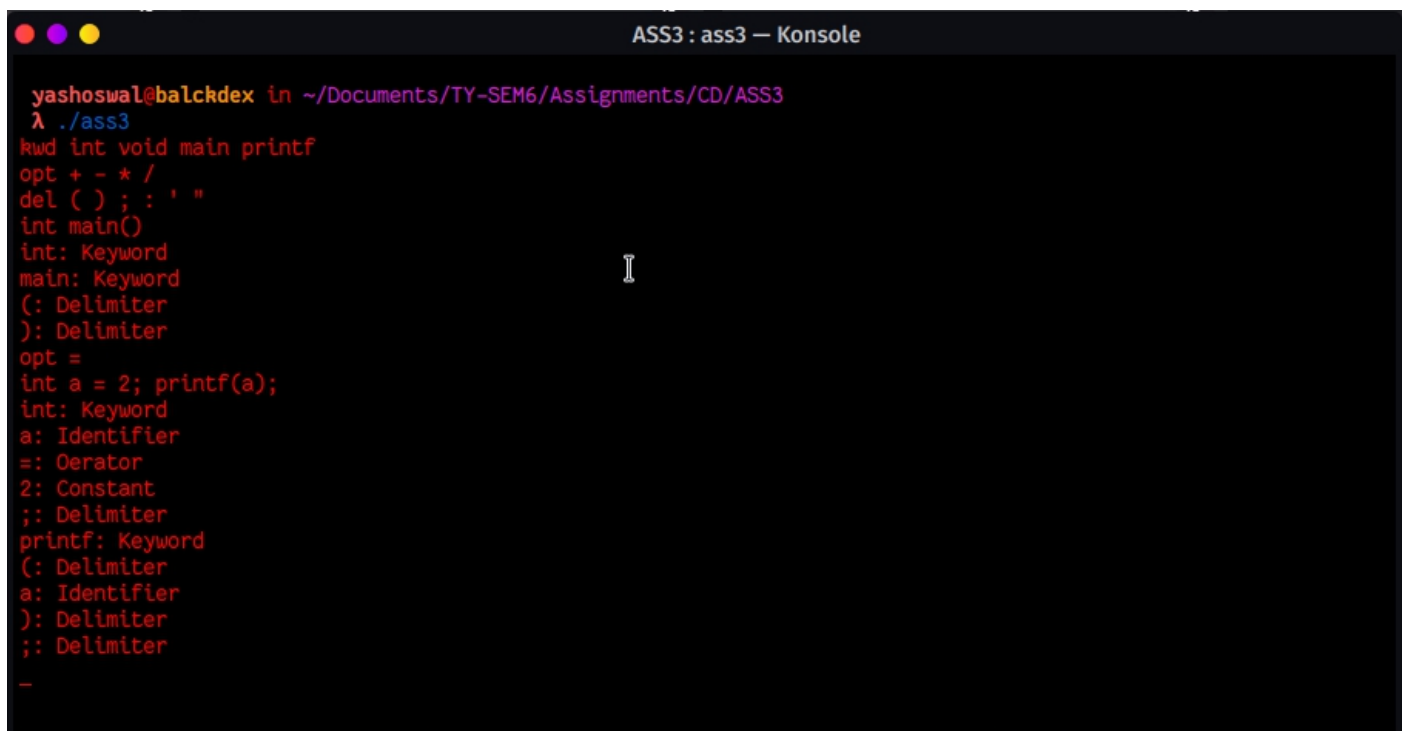
Div: B Roll no.: 38

SRN: 201901226

Input:

```
kwd void main int float printf
int a = 5; a = a + 2; printf(a);
```

Output:



```
ASS3 : ass3 — Konsole
yashoswal@balckdex in ~/Documents/TY-SEM6/Assignments/CD/ASS3
λ ./ass3
kwd int void main printf
opt + - * /
del ( ) ; : ' "
int main()
int: Keyword
main: Keyword
(: Delimiter
): Delimiter
opt =
int a = 2; printf(a);
int: Keyword
a: Identifier
=: Operator
2: Constant
;; Delimiter
printf: Keyword
(: Delimiter
a: Identifier
): Delimiter
;; Delimiter
-
```

Source Code:

```
%{
enum {
    LOOKUP = 0, /* default - looking rather than defining. */
    KWD,
    IDE,
    OPT,
    DEL
};
int state;
int add_word(int type, char *word);
int lookup_word(char *word);
```

```
%}
```

```
%%
```

```
\n    { state = LOOKUP; }  
^kwd  { state = KWD; }  
^ide   { state = IDE; }  
^opt   { state = OPT; }  
^del   { state = DEL; }  
[0-9]  { printf("%s: Constant\n", yytext); }
```

```
[+ - / * = %] {  
    if(state != LOOKUP) {  
        add_word(state, yytext);  
    } else {  
        switch(lookup_word(yytext)) {  
            case OPT:  
                printf("%s: Operator\n", yytext);  
                break;  
            default:  
                printf("%s: Identifier\n", yytext);  
                break;  
        }  
    }  
}
```

```
[; ' " ) ( : ] {  
    if(state != LOOKUP) {  
        add_word(state, yytext);  
    } else {  
        switch(lookup_word(yytext)) {  
            case DEL:  
                printf("%s: Delimiter\n", yytext);  
                break;  
            default:  
                printf("%s: Identifier\n", yytext);  
                break;  
        }  
    }  
}
```

```
[a-zA-Z]+ {  
    if(state != LOOKUP) {  
        add_word(state, yytext);  
    } else {  
        switch(lookup_word(yytext)) {  
            case KWD:  
                printf("%s: Keyword\n", yytext);  
                break;  
            case IDE:
```

```

                printf("%s: Identifier\n", yytext);
                break;
            default:
                printf("%s: Identifier\n", yytext);
                break;
        }
    }
}

.    /* ignore anything else */ ;

%%
int main()
{
    yylex();
}
struct word {
    char *word_name;
    int word_type;
    struct word *next;
};
struct word *word_list;
extern void *malloc() ;
int add_word(int type, char *word)
{
    struct word *wp;
    if(lookup_word(word) != LOOKUP)
    {
        printf("!!! warning: word %s already defined \n", word);
        return 0;
    }
    wp = (struct word *) malloc(sizeof(struct word));
    wp->next = word_list;
    wp->word_name = (char *) malloc(strlen(word)+1);
    strcpy(wp->word_name, word);
    wp->word_type = type;
    word_list = wp;
    return 1;
}
int lookup_word(char *word)
{
    struct word *wp = word_list;
    for(; wp; wp = wp->next) {
        if(strcmp(wp->word_name, word) == 0)
            return wp->word_type;
    }
    return LOOKUP;
}

```