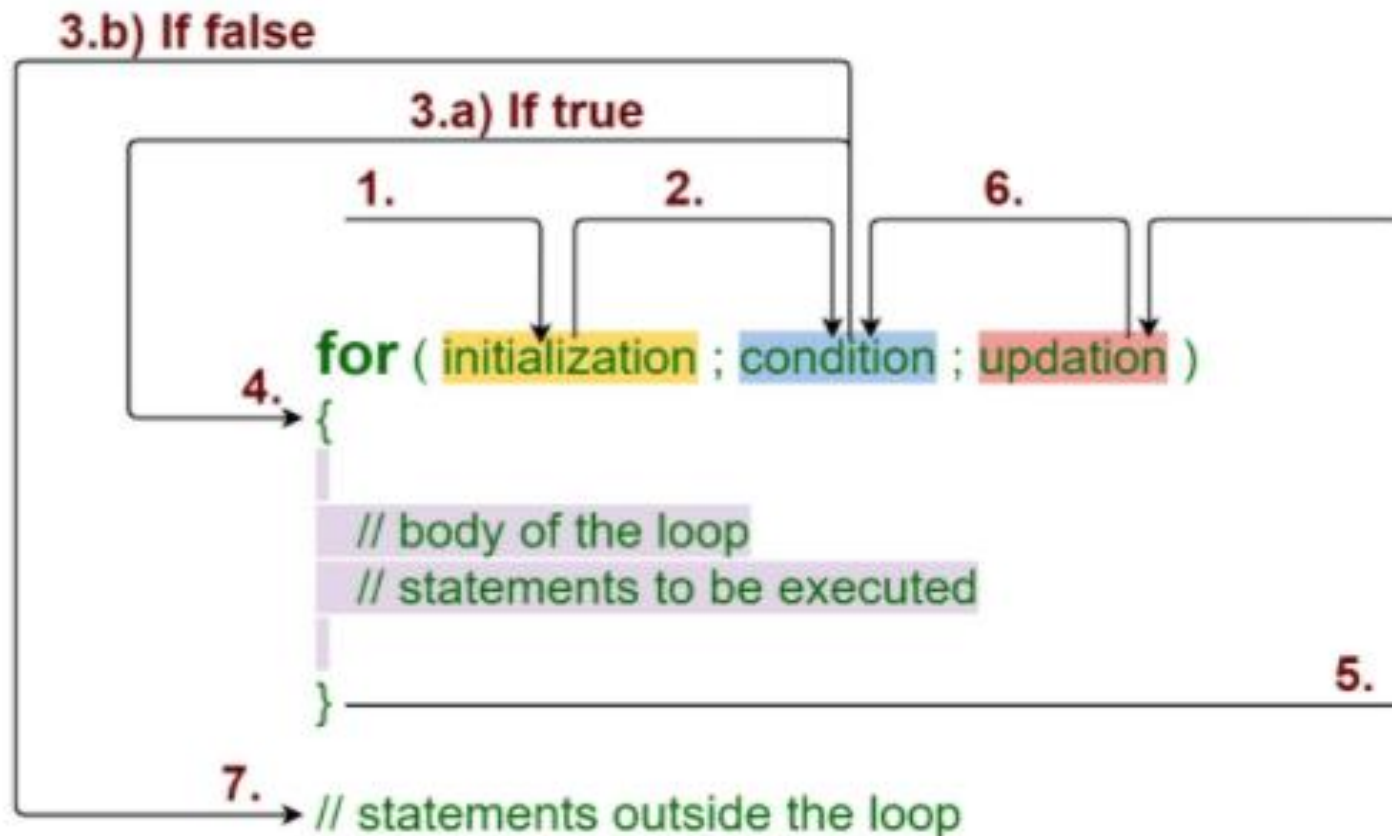


Lab7 – for loop

Parsing FOR statement

For Loop



Parsing FOR statement

CORRECT:

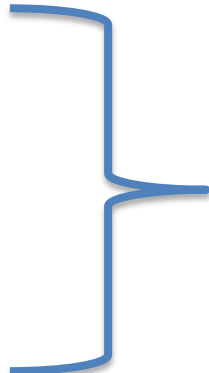
```
for(int i=1;i<10;i++)  
{  
    f15=99;  
    m3=v77;  
}
```



Parsed FOR statement

UNCORRECT:

```
for(i=1 int ; i++; i<10)  
{  
    f15=99;  
    m3=v77;  
}
```



Error parsed FOR statement

Parsing FOR statement

TEXT file1:(for1.t)

```
for(int i=10;i<15;i--)  
{  
    f15=99;  
    m3=v77;  
}
```



Parsed FOR statement

Parsing FOR statement

יש לבנות תכנית שמקבלת כקלט קוד של לולאת
FOR

```
for(int i=10;i<15;i--)  
{  
    f15=99;  
    m3=v77;  
}
```

Parsing FOR statement

LEX file1:(for1.l)

%%

for return FOR;

int return INT;

[a-z]+[a-z0-9]* return ID;

[0-9]+ return NUM;

[\t] printf(" ");

. return yytext[0];

.\n ;

%%

Parsing FOR statement

YACC file1:(for1.y)

```
%token FOR,INT,ID,NUM
```

```
%%
```

```
S: ST { printf("Parsed FOR statement\n"); }
```

```
ST: FOR '(' INT ASS1 COND ';' ASS2 ')' '{' BLOCK '}';
```

```
ASS1: ID '=' NUM ';' | ID '=' ID ';';
```

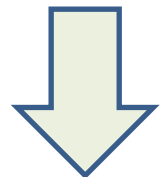
```
ASS2: ID '+' '+' | ID '-' '-' ;
```

```
COND: ID '<' ID | ID '>' ID | ID '<' NUM | ID '>' NUM;
```

```
BLOCK: BLOCK ASS1 | ASS1;
```

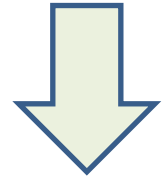
```
%%
```

```
for(int i=1  
{  
    f15=99  
    m3=v7  
}
```



Parsing FOR statement

YACC file1:(for1.y)



```
#include "lex.yy.c"
main()
{
    return yyparse();
}
int yyerror()
{
    printf("Error FOR statement parsing\n");
    return 0;
}
```


Parsing FOR statement

nano for1.t

בניית קובץ טקסט לבדיקת תחבירית

nano for1.l

בניית קובץ LEX

nano for1.y

בניית קובץ YACC

lex for1.l

LEX קומפילציה

yacc for1.y

YACC קומפילציה

cc -o for4 y.tab.c -ll -Ly

C קומפילציה

./for1<for1.t

הרצת הקובץ text2

Parsing FOR statement

TEXT file2:(for2.t)

```
for(int i3=10;i3<100;i++)  
{  
    f15=99;  
    m3=v77;  
    for(int j1 = 50;j1>0; j1--)  
    {  
        s33=15;  
        m5=v77;  
    }  
    k15=a99;  
}
```



Nested FOR statements

Parsing FOR statement

יש לבנות תכנית שמקבלת כקלט קוד של לולאת
FOR

```
for(int i3=10;i3<100;i++) -  
{  
    f15=99;  
    m3=v77;  
    for(int j1 = 50;j1>0;j1--)  
    {  
        s33=15;  
        m5=v77;  
    }  
    k15=a99;  
}
```

Parsing FOR statement

YACC file2:(for2.y)

```
%token FOR,INT,ID,NUM
```

```
%%
```

```
S: ST { printf("Parsed FOR statement\n"); }
```

```
ST: FOR '(' INT ASS1 COND ';' ASS2 ')' '{' BLOCK '}';
```

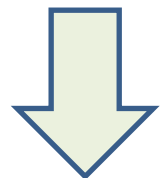
```
ASS1: ID '=' NUM ';' | ID '=' ID ';';
```

```
ASS2: ID '+' '+' | ID '-' '-' ;
```

```
COND: ID '<' ID | ID '>' ID | ID '<' NUM | ID '>' NUM;
```

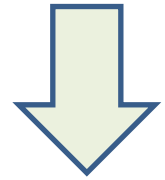
```
BLOCK: BLOCK ASS1 | ASS1 | BLOCK ST | ST;
```

```
%%
```



Parsing FOR statement

YACC file2:(for2.y)



```
#include "lex.yy.c"
main()
{
    return yyparse();
}
int yyerror()
{
    printf("ERROR FOR STATEMENT PARSING\n");
    return 0;
}
```

Parsing FOR statement

nano for2.t

בניית קובץ טקסט לבדיקת תחבירית

nano for2.l

בניית קובץ LEX

nano for2.y

בניית קובץ YACC

lex for2.l

LEX קומפילציה

yacc for2.y

YACC קומפילציה

cc -o for2 y.tab.c -ll -Ly

C קומפילציה

./for2<for2.t

הרצת הקובץ text2