CUT-COPY-COMPILE

CS251 PROJECT

Abinash Acharya-11840050
Asad Abidi-11840220
Thummala Milind Kesar-11841160

Files Submitted:

1) Lexer.l
2) lex.yy.c
3) Parser.y
4) Parser.tab.h
5) Parser.tab.c
6) correct(Sample Program)
7) wrong(Sample Program)
8) README.md

Note:- The file 'correct' contains a sample bash program written with correct syntax.
It contains the following program :

```
#!/bin/bash
# Basic function
print_something () {
echo "Hello I am a function"
}

counter=0
echo "Enter your lucky number"
read n

if [ "$n" -lt 50 ];
then
   until [ "$counter" -gt 5 ]
   do
   echo "Counter: $counter"
   counter=counter+1
done
elif [[ "$n" -gt 70 ]];
then
echo "number entered is greater than 70"
else
echo "calling function :"
print_something
fi
```

The file ‘wrong’ contains a sample bash program written with the wrong syntax.
It contains the following program :

```
#!/bin/bash
# Basic function
print_something () {
echo "Hello I am a function"
}

for (( a = 0 ; a <= 10 ; a++ ));
   if [[ $a -lt 5 ]]
   then
   echo "a is less than 5"
   elif [[ $a -gt 5 && $a -lt 8 ]]
   then
   echo "a is between 5 and 7"
   else
   echo "a is greater than 8"
   fi
done
echo "calling function :"
print_something
```

Reason - ‘do’ is missing in the for loop.

Instructions to run:

$ bison -d Parser.y
$ flex Lexer.l
$ gcc Parser.tab.c lex.yy.c -lfl
$ ./a.out <filename>
> Input the Bash code to check for syntax errors
> Ctrl + d -to stop taking input

Sample Working programs

1) Hello World

```
#!/usr/bin/env bash

NAME="PersonX"
echo "Hello $NAME!"
```

2) Comment example

```
#!/bin/bash

# Add two numeric value
sum=25+35

#Print the result
#System.out.println("Hello World");
echo $sum
```

3) Switch-case example

```
case $INPUT_STRING in
hello)
        {echo "Hello yourself!"}
        ;;
bye)
        {echo "See you again!"}
        ;;
*)
        {echo "Sorry, I don't understand"}
        ;;
 esac
```

4) Switch-case example

```
#!/bin/bash

echo "Enter your lucky number"
read n
case $n in
101)
echo "You got 1st prize" ;;
510)
echo "You got 2nd prize" ;;
999)
```

```
echo "You got 3rd prize" ;;
*)
echo "Sorry, try for the next time" ;;
esac
```

5) If-Elif-Else example

```
#!/bin/bash

echo "Enter username"
read username
echo "Enter password"
read password

if [[ ( $username == "admin" && $password == "secret" ) ]]; then
echo "valid user"
else
echo "invalid user"
fi
```

6) If-Elif-Else example

```
#!/bin/bash

echo "Enter your lucky number"
read n

if [[ $n -eq 101 ]];
then
echo "You got 1st prize"
elif [[ $n -eq 510 ]];
then
echo "You got 2nd prize"
elif [[ $n -eq 999 ]];
then
echo "You got 3rd prize"

else
echo "Sorry, try for the next time"
fi
```

7) Function eg

```
#!/bin/bash
myfunc() {
   myresult='some value'
   echo $myresult
   if  [[ a<b ]]  ; then echo $myresult else echo $myresult fi
}
```

8) Function eg

```
#!/bin/bash
function greeting() {

str="Hello, $name"
echo "$str"

}

echo "Enter your name"
read name

val=$greeting
echo "Return value of the function is $val"
```

9) While loop

```
#!/bin/bash
valid=true
count=1
while [ $valid ]
do
echo $count
if [[ $count -eq 5 ]];
then
break
fi
count=count+1
done
```

10) For loop

```
#!/bin/bash
for (( counter=10; counter>0; counter-- ))
do
echo -n "$counter "
done
printf( "\n")
```

11) Arrays

```
#! /bin/bash

# To declare static Array
arr=(abinash randomX 1 milind randomY asad)

# To print all elements of array
echo "${arr[@]}"
echo "${arr[*]}"
echo "${arr[@]:0}"
echo "${arr[*]:0}"
```