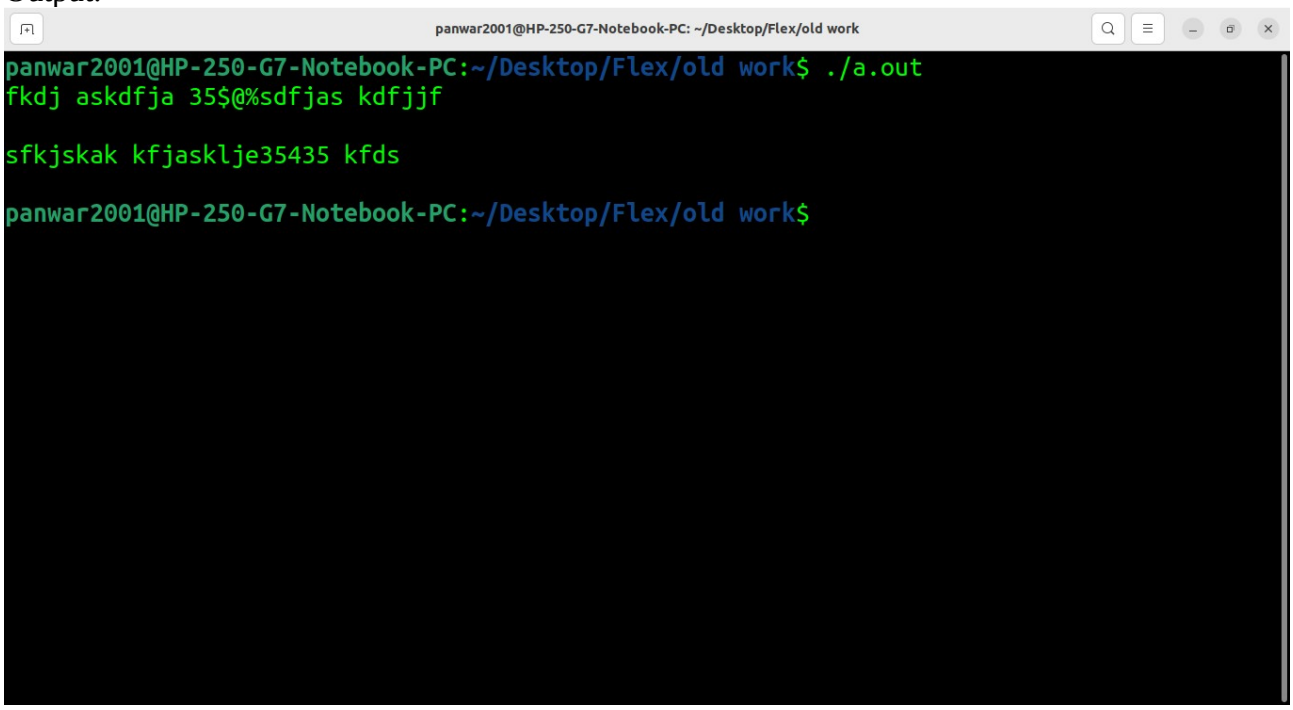
Student-Name: Ayush panwar, Section A, class Roll No. 34

1. Design a lex code to count the number of words in the input pattern.
Code:

```
%{
#include<stdio.h>
int wordCount=0;
%}

%%
[ ]+ ;
[^ \t\n]+ {wordCount++;}
%%
int yywrap(){};
int main(){
  yylex();
  printf("Number of words are %d",wordCount);
}
```

Output:



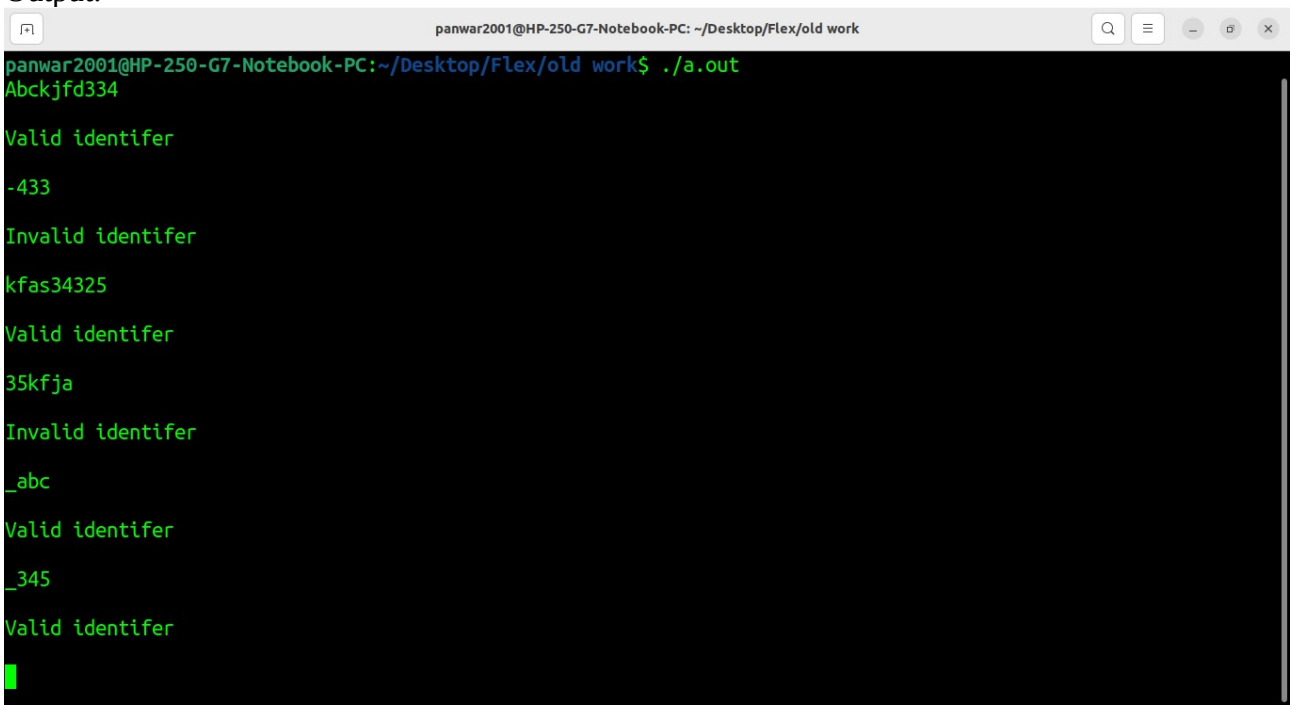
2. WAP in Lex to find a valid C identifier.

Code:

```
%{
#include<stdio.h>
%}

%%
^[a-zA-Z_][a-zA-Z0-9_]* { printf("\nValid identifer\n");}
.* { printf("\nInvalid identifer\n");}
%%
int yywrap(){};
int main(){
  yylex();
}
```

Output:



3. WAP in Lex to calculate number of alphabets, special characters and digits in a given stream of data.

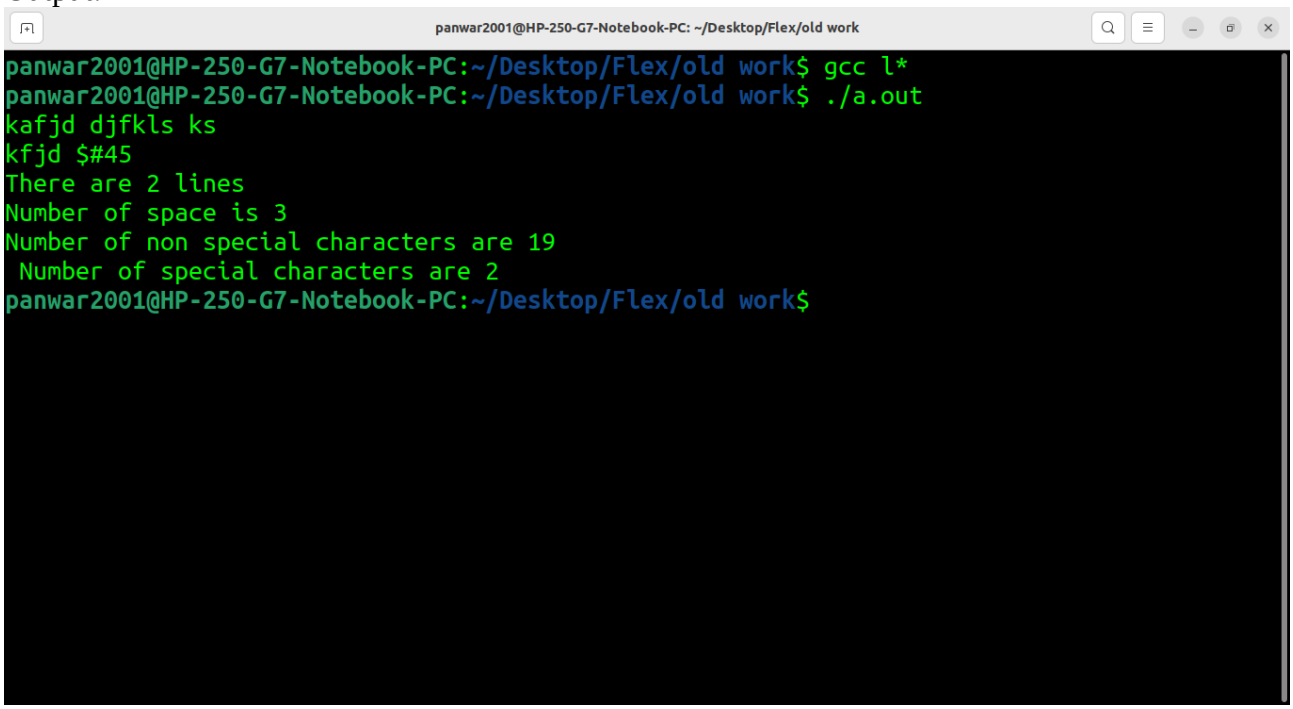
Code:

```
%{
#include<stdio.h>
int lineCount=0,spaceCount=0,nonSpecialCharacters=0,specialCharacters=0;
%}

%%
[\n] {lineCount++; }
[ ]+ {spaceCount++;}
[A-Za-z0-9] {nonSpecialCharacters++;}
[^A-Za-z0-9] {specialCharacters++;}
%%

int yywrap(){};
int main(){
  yylex();
  printf("There are %d lines \nNumber of space is %d \nNumber of non special characters are %d \n
Number of special characters are %d \
n",lineCount,spaceCount,nonSpecialCharacters,specialCharacters);
}
```

Output:

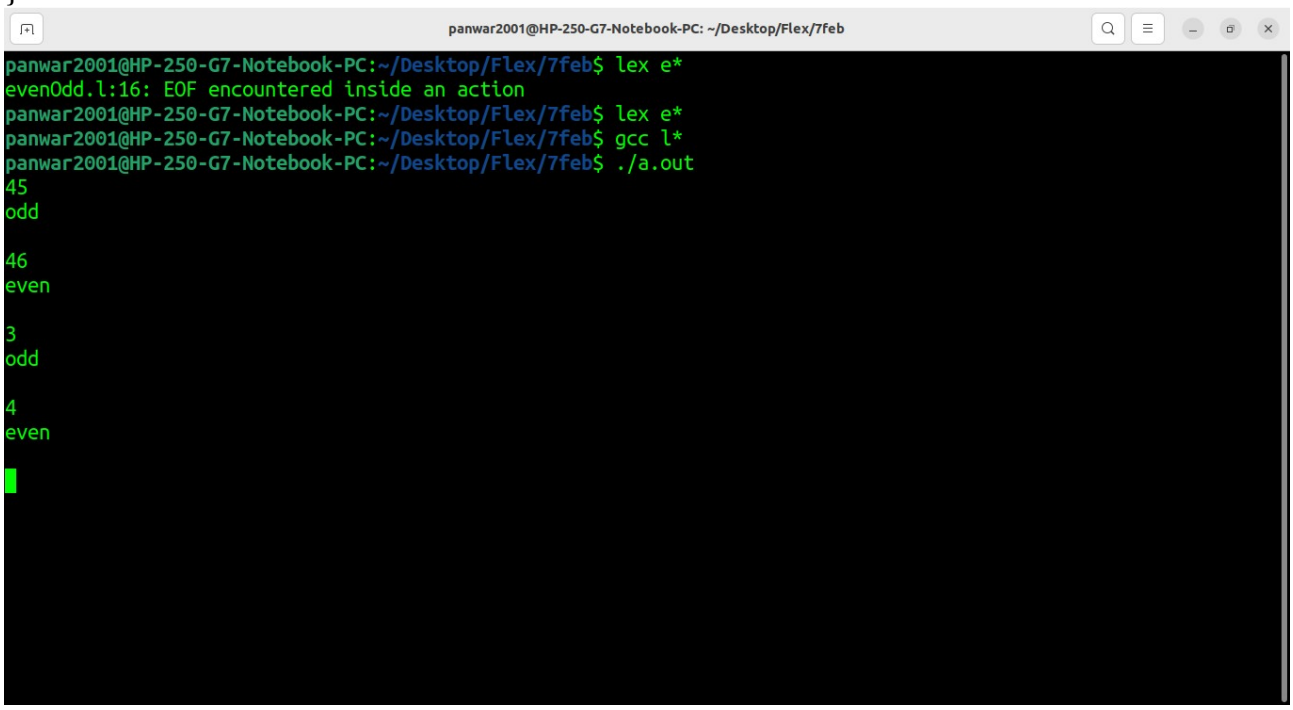


4. WAP in Lex to find if a number is even or odd.
Code:

```
%{
#include<stdio.h>
int longestStringLength=0;
int num;
%}

%%
^[+-]?[0-9]+ {num=atoi(yytext);num%2?printf("odd\n"):printf("even\n");}
%%

int yywrap(){};
int main(){
  yylex();
}
```

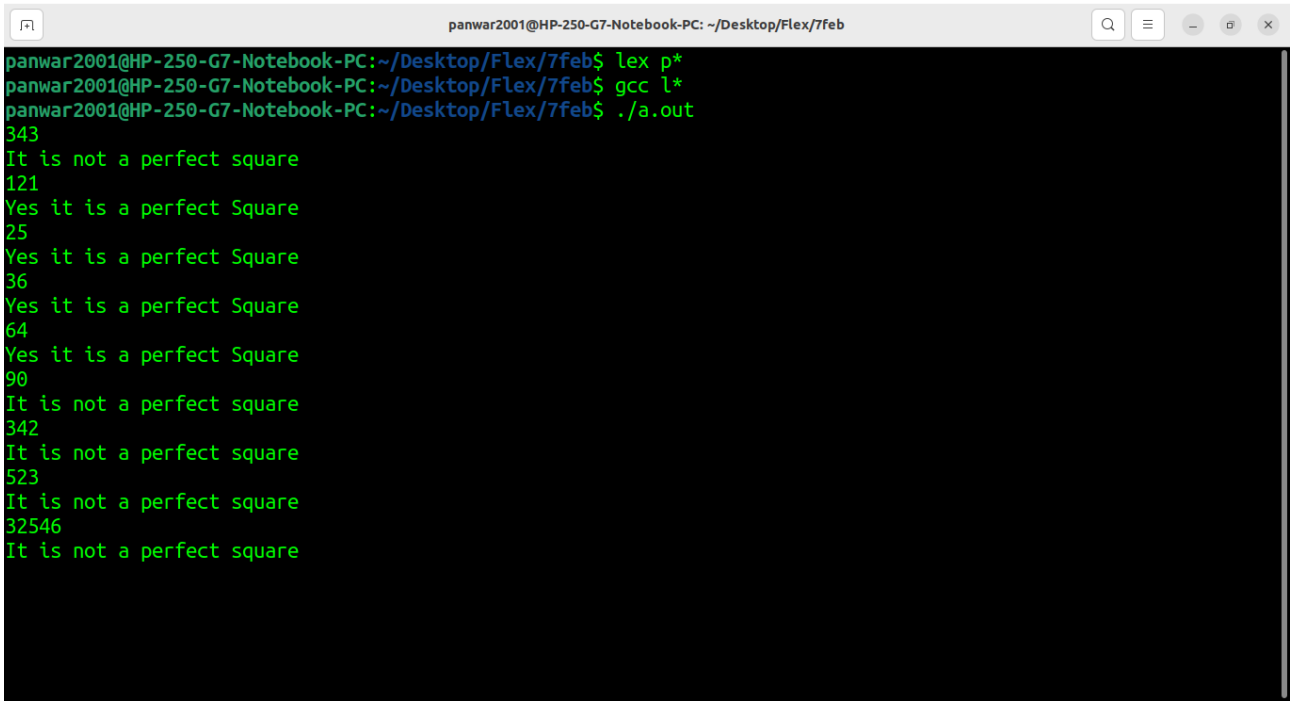


5. WAP in Lex to find if a number is perfect Square or not.
Code:

```
%{
#include<stdio.h>
int longestStringLength=0;
int num,isPerfectSquare=0;
%}

%%
[0-9]+ {num=atoi(yytext);
        for(int i=0;i<=num/2;i++){
         if(i*i==num){
           isPerfectSquare=1;
           break;
         }
        }
        if(isPerfectSquare==1){
          printf("Yes it is a perfect Square");
        }else{
          printf("It is not a perfect square");
        }
        isPerfectSquare=0;
       }
%%
int yywrap(){};
int main(){
  yylex();
}
```

Ouput:



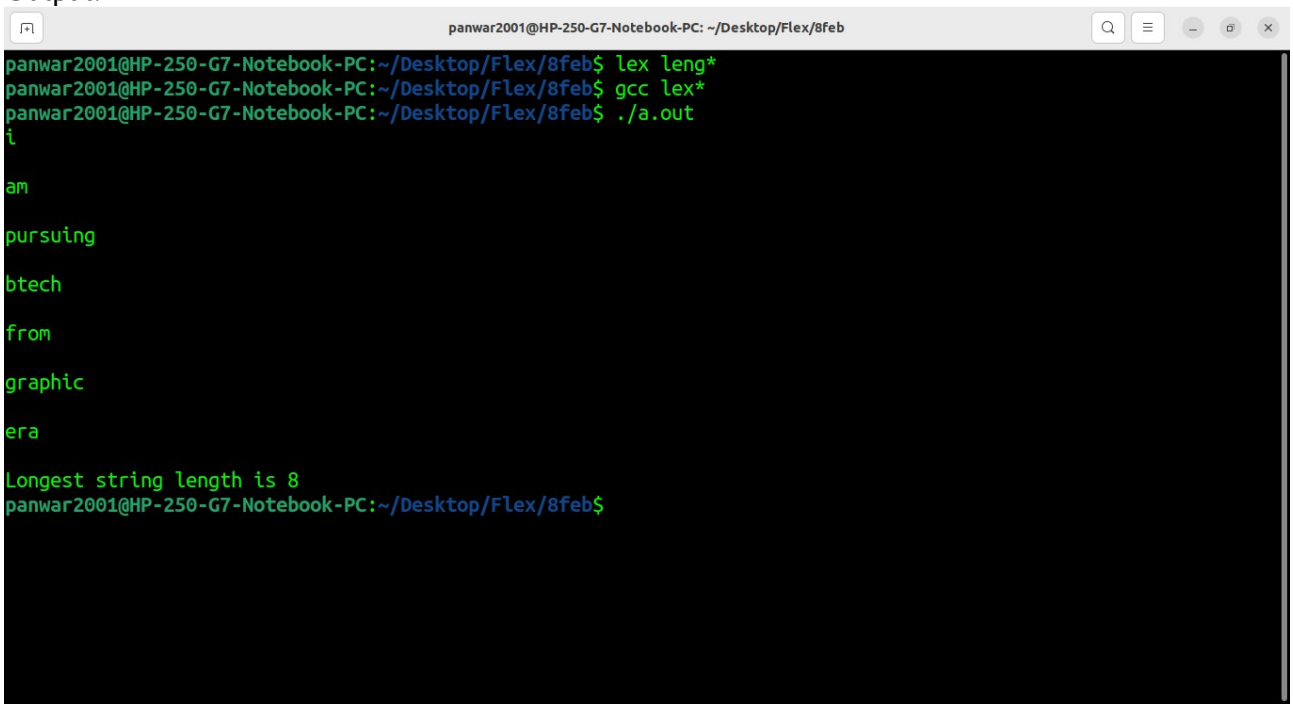
6. WAP in Lex to find the length of longest string in a given stream of data.
Code:

```
%{
#include<stdio.h>
int longestStringLength=0;
%}

%%
.* {longestStringLength=yyleng>longestStringLength?yyleng:longestStringLength;}
%%

int yywrap(){return 1;d};
int main(){
  yylex();
  printf("Longest string length is %d\n",longestStringLength);
}
```

Output:



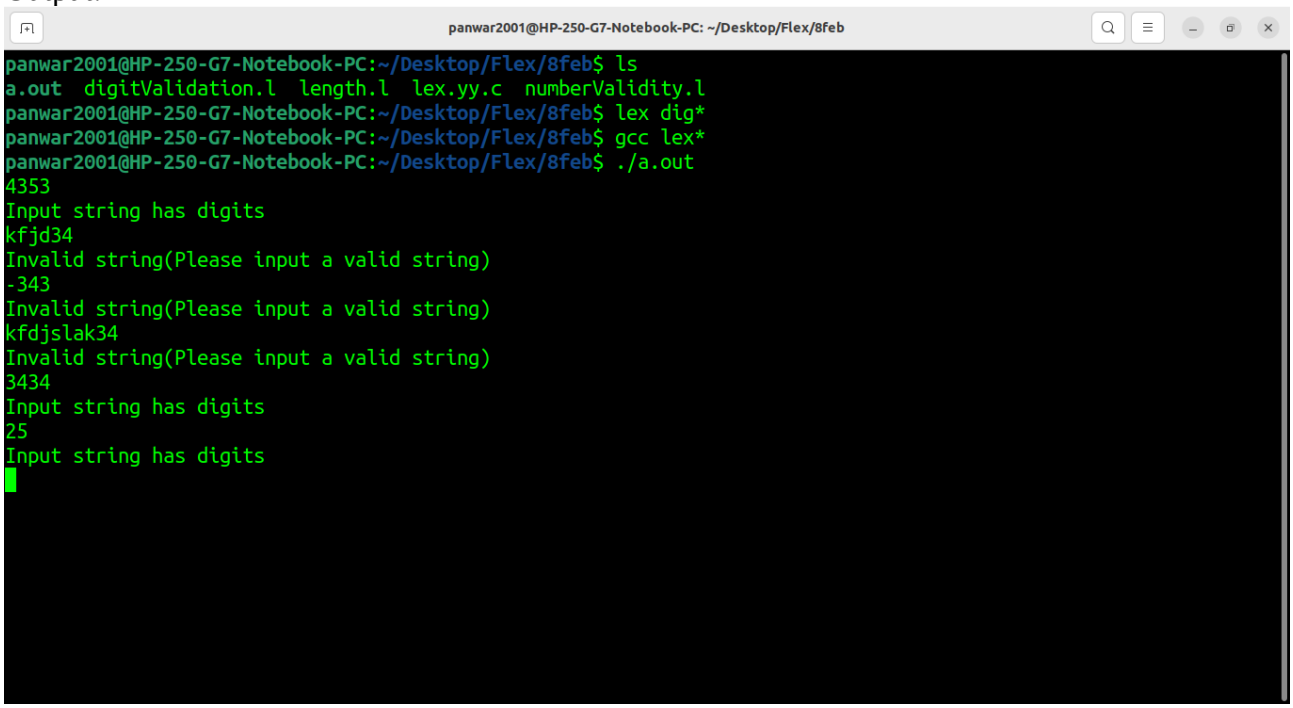
7. WAP in Lex to check if the given input string has digits or not.
Code:

```
%{
#include<stdio.h>
int longestStringLength=0;
%}

%%
[0-9]+ {printf("Input string has digits");}
.* {printf("Invalid string(Please input a valid string)");}
%%

int yywrap(){};
int main(){
  yylex();
}
```

Output:



8. WAP in Lex to check if the given input string is an Integer or a Floating point number.
Code:

```
%{
#include<stdio.h>
%}

%%
^[-+]?[0-9]+$ {printf("valid integer");}
^[-+]?[0-9]*[.][0-9]+$ {printf("Valid Float Number");}
.* {printf("Not a valid Number.Please enter a number0");}
%%

int yywrap(){return 1;}
int main(){
  yylex();
}
```

Output: