COMPILER DESIGN LAB (CSL5404)

Name: Lakhan Kumawat

Roll: 1906055

Program: B.Tech CSE
(5th Sem JUL-DEC 2021)

Compiler Lab Test

Question 1:

Q.1) Write a LEX program which takes your roll number as input and convert it into:

(i) octal if it is odd,
(ii) (ii)hexadecimal if it is even.

> Program Code

```
%{
#include<stdio.h>
int i;
int num, r, digit=0, count, pcount=0;
char a[20],arr[20];
%}

%%

[0-9]+   {

       num=atoi(yytext);
        if(num%2==0){
              printf("Even Roll \nHexadecimal :");

        while(num!=0)
        {

            r=num%16;
            digit='0'+r;
            if(digit>'9')
            digit+=7;
            a[count++]=digit;
            num=num/16;
```

```
        }

        for(i=count-1;i>=pcount;--i)
                printf("%c", a[i]);
                pcount=count;

        }

        else{
        printf("Odd Roll \n");
        while(num!=0)
            {

                r=num%8;
                digit='0'+r;
                arr[count++]=digit;
                num=num/8;

            }
            printf("octal :\n");
            for(i=count-1;i>=0;--i){
                printf("%c", arr[i]);
            }
        }

        }

%%

int yywrap(){}


int main()
{
    printf("Enter your roll: \n");
    yylex();
```

```
    return 0;
}
```

```
F:\Compiler Design\Lab\LakhanKumawat>"f:\Compiler Design\Lab\LakhanKumawat\a.exe"
Enter your roll:
1906055
Odd Roll
octal :
7212607
1906066
Even Roll
Hexadecimal :1D1592
```

Question 2:

Q.2) Given a text file, write a LEX program to search an input word in the file.Every time you encounter the input word then check:

a)If the first letter is a consonant, move it to the end of the word and then add your name to it.

b)If the first letter is a vowel, then just add your name to the end of the word. All non-letters are copied intact to the output.

> Program Code

```
%{
#include<stdio.h>
#include<string.h>
char replace [100] ;
int count = 0;
```

```
char temp_string[100] = "";
%}

%%

[a-zA-Z]+ {

  if(strcmp(yytext, replace)==0)
  {

    if(!(yytext[0] == 'a' || yytext[0] == 'e' || yytext[0] ==
'i' || yytext[0] == 'o' || yytext[0] == 'u' ||
    yytext[0] == 'A' || yytext[0] == 'E' || yytext[0] == 'I' ||
yytext[0] == 'O' || yytext[0] == 'U' ))
    {
        strcpy(temp_string,yytext);
        temp_string[yyleng] = yytext[0];
        temp_string[yyleng+1] = '\0';

        strcat(temp_string,"Ayush");
        fprintf(yyout, "%s", temp_string+1);

    }
    else
      {
        strcpy(temp_string,yytext);
        strcat(temp_string,"Lakhan");
        fprintf(yyout, "%s", temp_string);
      }
  }

  else{
    fprintf(yyout, "%s", yytext);
  }

}
```

```
.|\n fprintf(yyout, "%s", yytext);
%%

int yywrap()
{
    return 1;
}

int main()
{
    extern FILE *yyin, *yyout;
    printf("Enter word to search in the input file : ");
    scanf("%s",replace);
    yyin=fopen("input.txt", "r");
    yyout=fopen("output.txt", "w");

    yylex();

    printf("Output saved to file output.txt\n");
}
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
PS F:\Compiler Design\Lab\LakhanKumawat> flex main.lex
PS F:\Compiler Design\Lab\LakhanKumawat> gcc lex.yy.c
PS F:\Compiler Design\Lab\LakhanKumawat> & 'f:\Compiler Design\Lab\LakhanKumawat\a.exe'
Enter word to search in the input file : Lakhan
Output saved to file output.txt
PS F:\Compiler Design\Lab\LakhanKumawat> & 'f:\Compiler Design\Lab\LakhanKumawat\a.exe'
Enter word to search in the input file : Andy
Output saved to file output.txt
PS F:\Compiler Design\Lab\LakhanKumawat>
```

Input.txt: I am Lakhan Kumawat . Student of NIT PATNA . I love brewing my coffee . My rollno is 1906055 . My best friend's name is Andy Parker.

1. Output.txt: I am akhanLakhan Kumawat . Student of NIT PATNA . I love brewing my coffee . My rollno is 1906055 . My best friend's name is Andy Parker.
2. Output.txt: I am Lakhan Kumawat . Student of NIT PATNA . I love brewing my coffee . My rollno is 1906055 . My best friend's name is AndyLakhan Parker.

Question 3:

Q.3) Given a CFG, write a program in C to implement a top-down parser with back tracking. Check whether the string “cad” and “cabd” is accepted or rejected.

S →cAd
A →a/ ab

> Program Code

```
#include <string.h>

void check(void);
void set_value_backtracking(void);
void get_value_backtracking(void);
void display_output_string(void);
```

```
int iptr = 0, optr = 0, current_optr = 0;
char output_string[20], current_output_string[20],
input_string[20], temp_string[20];

int main()
{
    printf("\nEnter the string to check: ");
    scanf("%s", input_string);
    check();
    return 0;
}

void check(void)
{

    int flag = 1, rule2_index = 1;
    strcpy(output_string, "S");

    printf("\nThe output string in different stages are:\n");

    while (iptr <= strlen(input_string))
    {

        if (strcmp(output_string, temp_string) != 0)
        {
            display_output_string();
        }

        if ((iptr != strlen(input_string)) || (optr !=
strlen(output_string)))
        {
            if (input_string[iptr] == output_string[optr])
            {
                iptr = iptr + 1;
                optr = optr + 1;
```

```
            }

            else
            {
                if (output_string[optr] == 'S')
                {
                    memset(output_string, 0,
strlen(output_string));
                    strcpy(output_string, "cAd");
                }

                else if (output_string[optr] == 'A')
                {
                    set_value_backtracking();

                    if (rule2_index == 1)
                    {
                        memset(output_string, 0,
strlen(output_string));
                        strcpy(output_string, "cabd");
                    }

                    else
                    {
                        memset(output_string, 0,
strlen(output_string));
                        strcpy(output_string, "cad");
                    }
                }

                else if (output_string[optr] == 'b' &&
input_string[iptr] == 'd')
                {
                    rule2_index = 2;
                    get_value_backtracking();
                    iptr = iptr - 1;
```

```
                }

                else
                {
                    printf("\nThe given string, '%s' is
invalid.\n\n", input_string);
                    break;
                }
            }
        }

        else
        {
            printf("\nThe given string, '%s' is valid.\n\n",
input_string);
            break;
        }
    }
}

void set_value_backtracking(void)
{ //setting values for backtracking
    current_optr = optr;
    strcpy(current_output_string, output_string);
    return;
}

void get_value_backtracking(void)
{ //backtracking and obtaining previous values
    optr = current_optr;
    memset(output_string, 0, strlen(output_string));
    strcpy(output_string, current_output_string);
    return;
}

void display_output_string(void)
```

```
{
    printf("%s\n", output_string);
    memset(temp_string, 0, strlen(temp_string));
    strcpy(temp_string, output_string);
    return;
}
```

```
F:\Compiler Design\Lab\LakhanKumawat>"f:\Compiler Design\Lab\LakhanKumawat\main.exe"

Enter the string to check: cad

The output string in different stages are:
S
cAd
cabd
cAd
cad

The given string, 'cad' is valid.


F:\Compiler Design\Lab\LakhanKumawat>"f:\Compiler Design\Lab\LakhanKumawat\main.exe"

Enter the string to check: cabd

The output string in different stages are:
S
cAd
cabd

The given string, 'cabd' is valid.


F:\Compiler Design\Lab\LakhanKumawat>
```

Name : Lakhan Kumawat Roll No : 1906055 Course : CSL5404

End Of Assignment