SPCC Practical Questions 2023

	Questions	Marks
A	Write a program to create your own ‘C’ library using macros that can find the area of geometrical shapes (any 4)	10 Marks

- **Area.h [sq, rect, tri, circle(macros)]**

```
//Name:-Saloni Mhadgut  Roll no. 63 TECMPN-B4
   #define areasq(l1) (l1*l1)
   #define arearect(l2,b1) (l2*b1)
   #define areatri(b2,h) (0.5*b2*h)
   #define areacir(r) (3.14*r*r)
```

Area.c:

```
//Name:-Saloni Mhadgut  Roll no. 63 TECMPN-B4
   #include<stdio.h>
   #include<conio.h>
   #include"area.h"

   int main()
   {
       int l1,l2,b1,b2,h,r,area,opt;
       while(1)
       {
       printf("\n\n\tAREA OF VARIOUS SHAPES: ");
       printf("\n\n1. Square\t\t2. Rectangle\n3. Triangle\t\t4. Circle\n5. Exit");
       printf("\n\nEnter you choice: ");
       scanf("%d",&opt);

       switch(opt)
       {
               case 1: printf("\nEnter Length of the side of Square: ");
                       scanf("%d",&l1);
                       area=areasq(l1);
                        printf("\nThe Area of the Square is %d sq. cm",area);
                       break;

               case 2: printf("\nEnter Length and Breadth of the side of
   Rectangle: ");
                       scanf("%d",&l2);
                       scanf("%d",&b1);
                       area=arearect(l1,b1);
                        printf("\nThe Area of the Rectangle is %d sq. cm",area);
                       break;
```

```
                                  case 3: printf("\nEnter Breadth and Height of the side of the
            Triangle: ");
                                         scanf("%d",&b2);
                                         scanf("%d",&h);
                                         area=areatri(b2,h);
                                          printf("\nThe Area of the Triangle is %d sq. cm",area);
                                         break;

                                case 4: printf("\nEnter the radius of the circle: ");
                                         scanf("%d",&r);
                                         area=areacir(r);
                                          printf("\nThe Area of the Circle is %d sq. cm",area);
                                         break;

                                case 5: printf("\nExited the Program Successfully");
                                         exit(0);
                         }
                      }
                      return 0;
}
```

B

Write a LEX program to count and identify Vowels and consonants with output

5 Marks

```
%{
#include<stdio.h>
int vowel=0;
int con=0;
%}
%%
[aeiouAEIOU] {printf("Vowel\t\n");vowel++;}
[a-zA-Z] {printf("Consonant\t\n");con++;}
\n    {printf("Vowel=%d  and consonant=%d",vowel,con);}
%%
int main()
{
printf("Enter String: \n");
yylex();
}
int yywrap()
{
return 1;
}
```

	Questions	Marks
A	Write a program to convert the given computation into three address code. x = (a+b) * (c-d)	10 Marks

```
#include <iostream>
#include <string>
#include <stack>
#include <map>
#include <vector>
using namespace std;

int main(){
    string equation,postfix;
    stack<char> stack;
    map <char,int> precedence={{'/',4},{'*',3},{'+',2},{'-',1},{'(',0}};
    cout<<"Enter the equation : ";
    cin>>equation;
    for (int i=0;i<equation.length();i++){
        if (isalpha(equation[i])){
            postfix=postfix+equation[i];
        }
        else{
            if (stack.empty() || equation[i]=='('){
                stack.push(equation[i]);
            }
            else if(equation[i]==')'){
                postfix=postfix+string(1, stack.top());
                stack.pop();
                stack.pop();
            }
            else{
                auto pc=precedence.find(equation[i]);
                auto tc=precedence.find(stack.top());
                while (!stack.empty() && pc-> second <= tc->second){
                    postfix=postfix+string(1, stack.top());
                    stack.pop();
                }
                stack.push(equation[i]);
            }
        }
    }
    while (!stack.empty()){
        postfix=postfix+string(1, stack.top());
        stack.pop();
    }
    int count=0;
    vector<string> var;
    vector<string> tac;
    int i=0,n=postfix.length();
    while(postfix.length()>1){
        if (!isalnum(postfix[i])){
            var.push_back("t"+to_string(count));
            string opr1=string(1,postfix[i-2]);
            string opr2=string(1,postfix[i-1]);
            if (isdigit(postfix[i-2])){
                opr1=var.at(stoi(string(1,postfix[i-2])));
            }
            if (isdigit(postfix[i-1])){
                opr2=var.at(stoi(string(1,postfix[i-1])));
            }
            tac.push_back(var.at(count)+" = "+opr1+postfix[i]+opr2);
            postfix.replace(i-2,3,to_string(count));
            i=0;
            count+=1;
            n=postfix.length();
            continue;
        }
        i++;
    }
    for (auto elem : tac) {
        cout<<elem<<endl;
    }
    return 0;
}
```

B Write a LEX program to count and identify uppercase and lowercase letter with output — 5 Marks

```
%{
#include<stdio.h>
int upper=0;
int lower=0;
%}
%%
[A-Z] {printf("Upper Case\t\n");upper++;}
[a-z] {printf("Lower Case\t\n");lower++;}
\n    {printf("UpperCase=%d  and lowercase=%d",upper,lower);}
%%
int main()
{
printf("Enter String\n");
yylex();
}
int yywrap()
{
return 1;
}
```

	Questions	Marks
A	Write a program to create your own ‘C’ library using macros for conversions. (metre ⇔ feet, litre ⇔ cubic feet, °C ⇔ °F)	10 Marks

- **Convert.h (all macros) [C □□ F, metre □□ feet, litre □□ cubic feet]**

```
//Name:-Saloni Mhadgut  Roll no. 63 TECMPN-B4
 #define CtoF(C1) ((C1*1.8)+32)
 #define FtoC(F1) ((F1-32)*0.55556)
 #define MtoF(M1) (M1*3.28084)
 #define FtoM(F2) (F2*0.3048)
 #define LtoCF(L) (L*0.035315)
 #define CFtoL(CF) (CF*28.31685)
```

Covert.c

```
//Name:-Saloni Mhadgut  Roll no. 63 TECMPN-B4
 #include<stdio.h>
 #include<conio.h>
 #include"Convert.h"

 int main()
 {
     int c1,f1,m1,l,f2,cf, opt;
     float conv;
     while(1)
     {
     printf("\n\n\tCONVERSION: ");
     printf("\n\n1. Celsius to Fahrenheit\t2. Fahrenheit to Celsius\n3. Metre to
 Feet\t\t4. Feet to
    Metre\n5. Litre to Cubic Feet\t\t6. Cubic Feet to Litre\n7. Exit");
     printf("\n\nEnter you choice: ");
     scanf("%d",&opt);

     switch(opt)
     {
             case 1: printf("\nEnter Celsius: ");
```

```
                    scanf("%d",&c1);
                    conv=CtoF(c1);
                     printf("\n%d C = %f F ",c1,conv);
                    break;

            case 2: printf("\nEnter Fahrenheit: ");
                    scanf("%d",&f1);
                    conv=FtoC(f1);
                     printf("\n%d F = %f C ",f1,conv);
                    break;

           case 3: printf("\nEnter Metre: ");
                    scanf("%d",&m1);
                    conv=MtoF(m1);
                     printf("\n%d M = %f Ft ",m1,conv);
                    break;

            case 4: printf("\nEnter Feet: ");
                    scanf("%d",&f2);
                    conv=FtoM(f2);
                    printf("\n%d Ft = %f M ",f2,conv);
                    break;

           case 5: printf("\nEnter Litre: ");
                    scanf("%d",&l);
                    conv=LtoCF(l);
                     printf("\n%d L = %f Cu.ft ",l,conv);
                    break;

            case 6: printf("\nEnter Cubic Feet: ");
                    scanf("%d",&cf);
                    conv=CFtoL(cf);
                     printf("\n%d Cu.ft = %f L ",cf,conv);
                    break;


            case 7: printf("\nExited the Program Successfully");
                    exit(0);
      }
    }
    return 0;
  }
```

B	Write a LEX program to count the number of characters, words, sentences, lines, tabs, numbers and blank spaces present in input %option noyywrap	5 Marks

```
%{
        #include<stdio.h>
        int lines=1;
        int sentences=0;
        int words=0;
        int spaces=0;
        int chars=0;
        int num=0;
%}
%%
[A-Za-z]        {chars++;}
([1-9][0-9]*)   {num++;}
\n              {lines++;words++;sentences++;}
" "|"    "      {spaces++;words++;}
"$"             {printf("\ntotal lines: %d\n total words: %d\n total spaces: %d\n total
characters: %d\n total sentences: %d\n total numbers:
%d",lines,words,spaces,chars,sentences,num);}
%%
int main()
{
        printf("ENTER THE STRING:-  \n");
        yylex();
        return 1;
}
```

	Questions	**Marks**
A	Write a program to convert the given computation into three address code. x = a+ b*c -d and Display Quadruples and Triples	10 Marks

```
%{
int c=0;
```

B	Write a LEX program to count and identify tokens with output
	
	%{
	int c=0;
		5 Marks

```
%}
%%
"while"|"if"|"else"|"int"|"float"  {c++;printf("keywords : %s\n",yytext);}
[a-zA-Z_][a-zA-Z0-9_]*           {c++;printf("identifier : %s\n",yytext);}
"=="|"="|"++"|"+"|"*"|"-"          {c++;printf("operator : %s\n",yytext);}
[(){}|,;]                       {c++;printf("separator : %s\n", yytext);}
[0-9]*"."[0-9]+                 {c++;printf("float : %s\n", yytext);}
[0-9]+                          {c++;printf("integer : %s\n", yytext);}
.                     ;
\n                    {return 0;}
%%
int yywrap(){ return 0;}
int main(){
yylex();
printf("TOTAL NUMBER OF TOKEN = %d\n",c);
return 0;
}
```

	Questions	**Marks**
A	Write a program to create your own 'C' library using macros for conversions. (binary ⇔ decimal, binary ⇔ hexadecimal)	10 Marks

```
.c
#include<stdio.h>
#include "bin_hex_dec.h"
void main(){

int n;
printf("Enter binary number:");
scanf("%d",&n);

int choice;
printf("\n1.Decimal\n2.Hexadecimal");
printf("\nEnter choice:");
scanf("%d",&choice);
switch(choice){
   case 1: printf("\nDecimal Equivalent:%d",bin_dec_hex(n));
         break;
   case 2: printf("\nHexadecimal
Equivalent:%lX",bin_dec_hex(n));
         break;
   default:printf("\nWrong choice");
         break;
}
}

.h
#include<stdio.h>
#include<math.h>

#define bin_dec_hex(bin)({\
int dec=0,rem,i=0;\
while(bin!=0){ \
   rem=bin%10;\
   dec+=rem*pow(2,i);\
   i++;\
   bin=bin/10;\
}\
dec;\
})
```

B Write a LEX program to recognize valid arithmetic expressions — 5 Marks

```
%{
#include<stdio.h>
int v=0,op=0,id=0;
%}
%%
[0-9][0-9]* {id++;printf("\nIdentifier:");ECHO;}
[\+\-\*\/\=] {op++;printf("\nOperartor:");ECHO;}
"("                 {v++;}
")"                 {v--;}
.|\n      {return 0;}
%%
int main()
{
        printf("Enter the expression:\n");
        yylex();
        if((op+1) ==id && v==0)
        {
        printf("\n\nIdentifiers are:%d\nOperators are:%d\n",id,op);
        printf("\nExpression is Valid\n");
        }
        else
        printf("\nExpression is Invalid\n");
        return 1;
}
int yywrap()
```

```
{
        return 1;
}
```

	Questions	Marks
A	Write a program to create your own 'C' library using macros to generate series. (Factorial, prime numbers, leap years)	10 Marks

```
.c
#include<stdio.h>
#include"prime_fact_leap.h"

void main(){

int n,choice;
printf("1.Prime\n2.Factorial\n3.Leap year");
printf("\nEnter choice:");
scanf("%d",&choice);
switch(choice){
   case 1:printf("Enter number:");
          scanf("%d",&n);
          prime(n);
          break;
   case 2:printf("Enter number:");
         scanf("%d",&n);
         fact(n);
         break;
   case 3:printf("Enter year:");
          scanf("%d",&n);
          leap(n);
          break;
}
}

.h
#include<stdio.h>

#define prime(n)({\
int flag=0,i=0;\
if(n==0 || n==1){\
   flag =1;\
}else{\
    for(i=0;i<=n/2;i++){\
        if(n%i==0){\
            flag=1;\
            break;\
        }\
    }\
}\
if(flag==1){\
    printf("%d is not prime",n);\
}else{\
    printf("%d is prime",n);\
}\
})

#define fact(n)({\
int i=1,fact=1;\
if(n==0){ printf("Factorial of %d is %d",n,fact);}\
else{\
    while(i<=n){\
        fact*=i;\
        i++;\
    }\
printf("Factorial of %d is %d",n,fact);\
}\
}\
)

#define leap(n)({\
if(n%400==0){\
    printf("%d is Leap year",n);\
}else if(n%100==0){\
    printf("%d is not Leap year",n);\
}else if(n%4==0){\
    printf("%d is Leap year",n);\
}else{\
    printf("%d is not  Leap year",n);\
}})
```

B

Write a YACC program for Calculator performing four basic operations (+ , -, * and /)

5 Marks

Calculator.l

```
%option noyywrap
%{
        extern int yylval;
        #include<stdio.h>
        #include "y.tab.h"
%}

%%
[0-9]+ {yylval=atoi(yytext);return num;}
[+|\-|\*|\/] {return yytext[0];}
.|\n {return 0;}
%%
```

Calculator.y

```
%{
int yylex();
int yyerror();
%}
%{
        #include<stdio.h>
        #include<stdlib.h>
%}
%token num
%left '+''-'
%left '*''/'
%%
E:expr {printf("%d\n",$$);exit(0);}
expr:expr'+'expr {$$=$1+$3;}
|expr'-'expr {$$=$1-$3;}
|expr'*'expr {$$=$1*$3;}


|expr'/'expr {if ($3==0) {printf("Division by zero error\n");exit(0);} else $$=$1/$3;}
|'('expr')' {$$=$2;}
|num {$$=$1;}
;
%%
int yyerror()
{
        printf("Error");
        exit(0);
}
int main()
{
   printf("Enter an expression:");
        yyparse();
}
```


	Questions	Marks
A	Write a program to create your own 'C' library using macros to generate series. (Fibonacci Series, prime numbers, leap years)	10 Marks

.c
```
#include<stdio.h>
#include"prime_fib_leap.h"

void main(){

int n,choice;
printf("1.Prime\n2.Fibonacci\n3.Leap year");
printf("\nEnter choice:");
scanf("%d",&choice);
switch(choice){
   case 1:printf("Enter number:");
         scanf("%d",&n);
         prime(n);
         break;
   case 2:printf("Enter range:");
        scanf("%d",&n);
        fib(n);
        break;
   case 3:printf("Enter year:");
         scanf("%d",&n);
         leap(n);
         break;
}
}
```

.h
```
#include<stdio.h>

#define prime(n)({\
int flag=0,i=0;\
if(n==0 || n==1){\
   flag =1;\
}else{\
   for(i=0;i<=n/2;i++){\
      if(n%i==0){\
         flag=1;\
            break;\
        }\
    }\
}\
if(flag==1){\
    printf("%d is not prime",n);\
}else{\
    printf("%d is prime",n);\
}\
})

#define fib(n)({\
int a=0,b=1,c,i;\
printf("%d %d ",a,b);\
for(i=2;i<n;i++){\
    c=a+b;\
    printf("%d ",c);\
    a=b;\
    b=c;\
}\
})

#define leap(n)({\
if(n%400==0){\
    printf("%d is Leap year",n);\
}else if(n%100==0){\
    printf("%d is not Leap year",n);\
}else if(n%4==0){\
    printf("%d is Leap year",n);\
}else{\
    printf("%d is not  Leap year",n);\
}})
```

B	Write a YACC program that accepts all the strings ending with b preceded by any number of a's (a^nb)	5 Marks

```
.l
%{
#include "precededanb.tab.h"
%}
%%
[aA] {return A;}
[bB] {return B;}
\n {return NL;}
. {return yytext[0];}
```

```
%%
int yywrap()
{ return 1; }



.y
%{
#include<stdio.h>
#include<stdlib.h>
%}

%token A B NL

%%
stmt: A S B NL {printf("valid string\n"); exit(0);}
    ;

S: A S
|
    ;
%%

int yyerror(char *msg)
{
  printf("invalid string\n");
  exit(0);
}

int main() {
  printf("Enter the string:\n");
  yyparse();
  return 0;
}
```

	Questions	**Marks**

A	Write a program to convert the given computation into three address code and Display Quadruples and Triples
	*x = a*b/c+d;*
	`#include <iostream>`
	`#include <vector>`
	`#include <string>`
		10 Marks

```
using namespace std;
void qQuadruple(vector<string> expression) {
cout << "op\targ1\targ2\tresult" << endl;
for (int i = 0; i < expression.size(); i++) {
string expR = expression[i];
char op = expR[3];
char arg1 = expR[2];
char arg2 = expR[4];
char result = expR[0];
cout << op << "\t" << arg1 << "\t" << arg2 << "\t" << result << endl;
}
}
void tTriples(vector<string> expression) {
cout << "#\top\targ1\targ2" << endl;
int c = 0;
for (int i = 0; i < expression.size(); i++) {
string expR = expression[i];
char op = expR[3];
char arg1 = expR[2];
char arg2 = expR[4];
cout << i+c << "\t" << op << "\t" << arg1 << "\t" << arg2 << endl;
if (expR[0] != NULL) {
++c;
cout << i+c << "\t" << expR[1] << "\t" << expR[0] << "\t" << i+c-1 <<
endl;
}
}
}
int main() {
vector<string> exp;
int n;
string input;
cout << "Enter the number of expressions: ";
cin >> n;
cin.ignore(); // To consume the newline character after the integer input
cout << "Enter the expressions: " << endl;
for (int i = 0; i < n; i++) {
getline(cin, input);
exp.push_back(input);
}
cout << "Quadruple:" << endl << endl;
qQuadruple(exp);
cout << endl << "Triple:" << endl << endl;
tTriples(exp);
return 0;
}
```

	/*
	number of expression is 3
	f=c+d
	e=a-f
	g=b*e
	*/	

B	Write a YACC program that accepts all the strings ending with b preceded by any number of a’s	5 Marks

```
(a^n b^n)

.l
%{
#include "precededanbn.tab.h"
%}
%%
[aA] {return A;}
[bB] {return B;}
\n {return NL;}
. {return yytext[0];}
%%
int yywrap()
{ return 1; }


.y
%{
#include<stdio.h>
#include<stdlib.h>
%}

%token A B NL

%%
stmt: AB NL {printf("valid string\n"); exit(0);}
     ;

AB: A AB B
 |
     ;
%%
int yyerror(char *msg)
```

```
{
   printf("invalid string\n");
   exit(0);
}

int main() {
   printf("Enter the string:\n");
   yyparse();
   return 0;
}
```

	Questions	**Marks**
A	Write a program to create your own 'C' library using macros to find the properties of a given number n – factorial of n, sum of natural numbers till n	10 Marks

```
.c
#include<stdio.h>
#include"natural_fact.h"

void main(){

int n,choice;
printf("1.Natural\n2.Factorial");
printf("\nEnter choice:");
scanf("%d",&choice);
switch(choice){
   case 1:printf("Enter value of n:");
           scanf("%d",&n);
           natural(n);
           break;
   case 2:printf("Enter value of n:");
          scanf("%d",&n);
          fact(n);
          break;
}
}


.h
#include<stdio.h>

#define natural(n)({\
int s=n*(n+1)/2;\
printf("Sum of natural numbers till %d is %d",n,s);\
})

#define fact(n)({\
int i=1,fact=1;\
if(n==0){ printf("Factorial of %d is %d",n,fact);}\
else{\
   while(i<=n){\
      fact*=i;\
      i++;\
   }\
printf("Factorial of %d is %d",n,fact);\
}\
}\
)
```

B

Write a YACC program that accepts all the strings ending with b preceded by any number of a’s (a^nb^{n+1})

5 Marks

```
.l
%{
#include "precededanbn1.tab.h"
%}
%%
[aA] {return A;}
[bB] {return B;}
\n {return NL;}
. {return yytext[0];}
%%
int yywrap()
{ return 1; }


.y
%{
#include<stdio.h>
#include<stdlib.h>
%}

%token A B NL

%%
stmt: AB NL {printf("valid string\n"); exit(0);}
    ;
```

```
AB: A AB B
 | B
    ;
%%
int yyerror(char *msg)
{
   printf("invalid string\n");
   exit(0);
}

int main() {
   printf("Enter the string:\n");
   yyparse();
   return 0;
}
```

	Questions	Marks
A	Consider the following program, Display the Pass-1 of the Program START 501 A DS 1 B DS 1 C DS 1 READ A READ B MOVER AREG, A ADD AREG, B MOVEM AREG, C PRINT C END	10 Marks

```
#include <iostream>
#include <cstdlib>
#include <fstream>
#include <string>
using namespace std;

int count=0,instr=0,st=0;
string symtb[50][2];
string passes[50][50][2];
int start;
```

```
void checker(string arg){
  string
instructions[22][2]={{"STOP","00"},{"ADD","01"},{"SUB","02"},{"MULTI","03"},{"
MOVER","04"},{"MOVEM","05"},{"COMP","06"},{"BC","07"},{"DIV","08"},{"REA
D","09"},{"PRINT","10"},
  {"START","01"},{"END","02"},{"ORIGIN","03"},{"EQU","04"},{"LTORG","05"},
  {"DS","01"},{"DC","02"},
  {"AREG","01"},{"BREG","02"},{"CREG","03"},{"DREG","04"}};
  string symbols[4]={"A","B","C","D"};
  int rows;
  bool found;

  rows=end(symbols)-begin(symbols);
  for(int x=0; x<rows; x++){
    if (symbols[x]==arg){
      passes[count][instr][0]="S";
      for (int y=0; y<50; y++){
        if (symtb[y][0]==""){
          break;
        }
        if (symtb[y][0]==arg){
          passes[count][instr][1]=to_string(y);
          cout<<passes[count][instr][0]<<","<<passes[count][instr][1];
          return;
        }
      }
      symtb[st][0]=arg;
      symtb[st][1]=to_string(start);
      start+=1;
      passes[count][instr][1]=to_string(st);
      cout<<passes[count][instr][0]<<","<<passes[count][instr][1];
      st++;
      instr++;
      return;
    }
  }

  found=false;
  int x=0,cols=2;
  rows=sizeof(instructions)/sizeof(instructions[0]);
  for(x ; x < rows; x++){
    for(int y = 0; y < cols; y++){
      if(instructions[x][y]==arg){
```

```
                found = true;
                break;
            }
        }
        if (found==true){
            break;
        }
    }
    if (found){
        if (x>=0 && x<=10){
            passes[count][instr][0]="IS";
            passes[count][instr][1]=instructions[x][1];
            cout<<passes[count][instr][0]<<","<<passes[count][instr][1];
            instr++;
            return;
        }
        if (x>=11 && x<=15){
            passes[count][instr][0]="AD";
            passes[count][instr][1]=instructions[x][1];
            cout<<passes[count][instr][0]<<","<<passes[count][instr][1];
            instr++;
            return;
        }
        if (x>=16 && x<=17){
            passes[count][instr][0]="DL";
            passes[count][instr][1]=instructions[x][1];
            cout<<passes[count][instr][0]<<","<<passes[count][instr][1];
            instr++;
            return;
        }
        if (x>=18 && x<=21){
            passes[count][instr][0]="RG";
            passes[count][instr][1]=instructions[x][1];
            cout<<passes[count][instr][0]<<","<<passes[count][instr][1];
            instr++;
            return;
        }
    }
    else{
        if (! start){
            start=stoi(arg);
        }
        passes[count][instr][0]="C";
        passes[count][instr][1]=arg;
        cout<<passes[count][instr][0]<<","<<passes[count][instr][1];
        instr++;
```

```
  }

}

int main(){
  string
opcd[22][2]={{"STOP","00"},{"ADD","01"},{"SUB","02"},{"MULTI","03"},{"MOVE
R","04"},{"MOVEM","05"},{"COMP","06"},{"BC","07"},{"DIV","08"},{"READ","09
"},{"PRINT","10"},
  {"START","01"},{"END","02"},{"ORIGIN","03"},{"EQU","04"},{"LTORG","05"},
  {"DS","01"},{"DC","02"},
  {"AREG","01"},{"BREG","02"},{"CREG","03"},{"DREG","04"}};
  string line,res,word="";

  cout<<"\nPass 1 Result: "<<endl;
  ifstream file("srcprg.txt");
  if (file.is_open()){
     while (file.peek() != EOF){
        getline(file, line, '\n');
        for (auto x : line){
           if (x == ' '){
              cout<<"(";
              checker(word);
              cout<<") ";
              word = "";
           }
           else {
              if (x != ','){
                 word = word + x;
              }
           }
        }
        cout<<"(";
        checker(word);
        cout<<") ";
        cout<<endl;
        count+=1;
        instr=0;
        word="";

     }
  }
  else{
     cout << "Couldn't open the file\n";
  }
  cout<<"\nSymbol table : "<<endl;
```

	for(int i=0;i<50;i++){
	    if(symtb[i][0]!=""){
	        cout<<i<<" "<<symtb[i][0]<<" "<<symtb[i][1]<<endl;
	    }
	 }
	 return 0;
	}	

B	Write a YACC program that accepts all the strings ending with b preceded by any number of a's ($a^{2n}b^n$)
	
	.l
	%{
	#include "precededa2nbn.tab.h"
	%}
	%%
	[aA] {return A;}
	[bB] {return B;}
	\n {return NL;}
	. {return yytext[0];}
	%%
	int yywrap()
	{ return 1; }
	
	.y
	%{
	#include<stdio.h>
	#include<stdlib.h>
	%}
	
	%token A B NL
	
	%%
	stmt: AAB NL {printf("valid string\n"); exit(0);}
	    ;
	
	AAB: A A AAB B
	 |
	    ;
	%%
	
	int yyerror(char *msg)
	{
	   printf("invalid string\n");
		5 Marks

```
    exit(0);
}

int main() {
    printf("Enter the string:\n");
    yyparse();
    return 0;
}
```

	Questions	**Marks**
A	For the given program, Display the Pass-2 by taking intermediate code as an input Assembly program LC **Intermediate code (PASS-1)**	10 Marks

START 501 **(AD,01) (c,501)**
A DS 1 501 **(S,0) (DL,0) (c,1)**
B DS 1 502 **(S,1) (DL,0) (c,1)**
C DS 1 503 **(S,2) (DL,0) (c,1)**
READ A 504 **(IS,09) (S,0)**
READ B 505 **(IS,09) (S,1)**
MOVER AREG, A 506 **(IS,04) (RG,01) (S,0)**
ADD AREG, B 507 **(IS,01) (RG,01) (S,1)**
MOVEM AREG, C 508 **(IS,05) (RG,01) (S,2)**
PRINT C 509 **(IS,10) (S,2)**
END 510 **(AD,02)**

```
#include <iostream>
#include <cstdlib>
#include <fstream>
#include <string>
using namespace std;

int count=0,instr=0,st=0;
string symtb[50][2];
string passes[50][50][2];
int start;



void checker(string arg){
  string
instructions[22][2]={{"STOP","00"},{"ADD","01"},{"SUB","02"},{"MULTI","03"},{"
MOVER","04"},{"MOVEM","05"},{"COMP","06"},{"BC","07"},{"DIV","08"},{"REA
D","09"},{"PRINT","10"},
{"START","01"},{"END","02"},{"ORIGIN","03"},{"EQU","04"},{"LTORG","05"},
{"DS","01"},{"DC","02"},
{"AREG","01"},{"BREG","02"},{"CREG","03"},{"DREG","04"}};
string symbols[4]={"A","B","C","D"};
int rows;
bool found;

rows=end(symbols)-begin(symbols);
for(int x=0; x<rows; x++){
    if (symbols[x]==arg){
        passes[count][instr][0]="S";
        for (int y=0; y<50; y++){
            if (symtb[y][0]==""){
                break;
            }
            if (symtb[y][0]==arg){
                passes[count][instr][1]=to_string(y);
                return;
            }
        }
        symtb[st][0]=arg;
        symtb[st][1]=to_string(start);
        start+=1;
        passes[count][instr][1]=to_string(st);
        st++;
        instr++;
        return;
    }
}

found=false;
int x=0,cols=2;
rows=sizeof(instructions)/sizeof(instructions[0]);
for(x ; x < rows; x++){
    for(int y = 0; y < cols; y++){
        if(instructions[x][y]==arg){
            found = true;
            break;
        }
    }
    if (found==true){
        break;
    }
}
if (found){
    if (x>=0 && x<=10){
            passes[count][instr][0]="IS";
            passes[count][instr][1]=instructions[x][1];
            instr++;
            return;
        }
        if (x>=11 && x<=15){
            passes[count][instr][0]="AD";
            passes[count][instr][1]=instructions[x][1];
            instr++;
            return;
        }
        if (x>=16 && x<=17){
            passes[count][instr][0]="DL";
            passes[count][instr][1]=instructions[x][1];
            instr++;
            return;
        }
        if (x>=18 && x<=21){
            passes[count][instr][0]="RG";
            passes[count][instr][1]=instructions[x][1];
            instr++;
            return;
        }
    }
    else{
        if (! start){
            start=stoi(arg);
        }
        passes[count][instr][0]="C";
        passes[count][instr][1]=arg;
        instr++;
    }

}

int main(){
    string
opcd[22][2]={{"STOP","00"},{"ADD","01"},{"SUB","02"},{"MULTI","03"},{"MOVE
R","04"},{"MOVEM","05"},{"COMP","06"},{"BC","07"},{"DIV","08"},{"READ","09
"},{"PRINT","10"},
    {"START","01"},{"END","02"},{"ORIGIN","03"},{"EQU","04"},{"LTORG","05"},
    {"DS","01"},{"DC","02"},
    {"AREG","01"},{"BREG","02"},{"CREG","03"},{"DREG","04"}};
    string line,res,word="";

    ifstream file("srcprg.txt");
```

```
if (file.is_open()){
    while (file.peek() != EOF){
        getline(file, line, '\n');
        for (auto x : line){
            if (x == ' '){
                checker(word);
                word = "";
            }
            else {
                if (x != ','){
                    word = word + x;
                }
            }
        }
        checker(word);
        count+=1;
        instr=0;
        word="";

    }
}
else{
    cout << "Couldn't open the file\n";
}
cout<<"Pass 2: "<<endl;
for(int i=0;i<50;i++){
    if (passes[i][0][0]==""){
        break;
    }
    if (passes[i][0][0]=="IS"){
        for(int j=0;j<50;j++){
            if (passes[i][j][0]==""){
                break;
            }
            if (j==1 && passes[i][j][0]=="S"){
                cout<<"00 "<<symtb[stoi(passes[i][j][1])][1]<<endl;
            }
            else{
                if (passes[i][j][0]=="S"){
                    cout<<symtb[stoi(passes[i][j][1])][1]<<endl;
                }
                else{
                    cout<<passes[i][j][1]<<" ";
                }
            }
        }
        }
    }
    cout<<"\nSymbol table : "<<endl;
    for(int i=0;i<50;i++){
        if(symtb[i][0]!=""){
            cout<<i<<" "<<symtb[i][0]<<" "<<symtb[i][1]<<endl;
        }
    }
    return 0;
}
```

B Write a LEX program to count number of lines, numbers and blank spaces. 5 Marks

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

int line_count = 0;
int num_count = 0;
int space_count = 0;
%}

%%

\n          { line_count++; }
[0-9]+        { num_count++; }
[ \t]       { space_count++; }
.          { /* ignore all other characters */ }

%%

int main()
{
    printf("Enter input:\n");
    yylex();
    printf("Line count: %d\n", line_count);
    printf("Number count: %d\n", num_count);
    printf("Blank space count: %d\n", space_count);
    return 0;
}

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

	Questions	Marks

A	Consider the following Three address code as Input and display Triples and Quadruples f=c+d
	e=a-f
	g=b*e
	
	#include <iostream>
	#include <vector>
	#include <string>
	using namespace std;
	void qQuadruple(vector<string> expression) {
	cout << "op\targ1\targ2\tresult" << endl;
	for (int i = 0; i < expression.size(); i++) {
	string expR = expression[i];
	char op = expR[3];
	char arg1 = expR[2];
	char arg2 = expR[4];
	char result = expR[0];
	cout << op << "\t" << arg1 << "\t" << arg2 << "\t" << result << endl;
	}
	}
	void tTriples(vector<string> expression) {
	cout << "#\top\targ1\targ2" << endl;
	int c = 0;
	for (int i = 0; i < expression.size(); i++) {
	string expR = expression[i];
	char op = expR[3];
	char arg1 = expR[2];
	char arg2 = expR[4];
	cout << i+c << "\t" << op << "\t" << arg1 << "\t" << arg2 << endl;
	if (expR[0] != NULL) {
	++c;
	cout << i+c << "\t" << expR[1] << "\t" << expR[0] << "\t" << i+c-1 <<
	endl;
	}
	}
	}
	int main() {
	vector<string> exp;
	int n;
	string input;
	cout << "Enter the number of expressions: ";
	cin >> n;
	cin.ignore(); // To consume the newline character after the integer input
		10 Marks

```
cout << "Enter the expressions: " << endl;
for (int i = 0; i < n; i++) {
getline(cin, input);
exp.push_back(input);
}
cout << "Quadruple:" << endl << endl;
qQuadruple(exp);
cout << endl << "Triple:" << endl << endl;
tTriples(exp);
return 0;
}

/*
number of expression is 3
f=c+d
e=a-f
g=b*e
*/
```

B

Write a YACC program that accepts all the strings ending with b preceded by any number of a’s ($a^nb^nc^n$)

5 Marks

```
.l
%{
#include "precededanbncn.tab.h"
%}
%%
[aA] {return A;}
[bB] {return B;}
[cC] {return C;}
\n {return NL;}
. {return yytext[0];}
%%
int yywrap()
{ return 1; }


.y
%{
#include<stdio.h>
#include<stdlib.h>
%}

%token A B C NL
```

```
%%
stmt: ABC NL {printf("valid string\n"); exit(0);}
    ;

ABC: A AB B BC C
AB: A AB B
BC: B BC C
|
    ;
%%
int yyerror(char *msg)
{
   printf("invalid string\n");
   exit(0);
}

int main() {
   printf("Enter the string:\n");
   yyparse();
   return 0;
}
```

	Questions	**Marks**

A	Write a program to optimize the given three address code.
	T1= 5*3+10 // Constant folding
	T3=T1 //variable propagation
	T2=T1+T3
	T5=4*T2 // common sub-expression elimination
	T6=4*T2+100
		10 Marks

```
import java.util.*;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class Main
{
HashMap < String, String > statements = new HashMap <> ();
List < String > result = new ArrayList < String > (Arrays.asList ("T1", "T3",
"T2", "T5", "T6"));
List < String > operators =
new ArrayList < String > (Arrays.asList ("+", "*"));
public static void main (String[]args)
{
Main obj = new Main();
obj.getStatements ();
System.out.println ("Initaially statements are: ");
obj.putStatements ();
obj.constantFolding ();
System.out.println ("After constant folding: ");
obj.putStatements ();
obj.variablePropagation ();
System.out.println ("After variable propagation: ");
obj.putStatements ();
obj.commonSubexpElim ();
System.out.println ("After Common Sub-expression Elimination: ");
obj.putStatements ();
}
public void getStatements ()
{
this.statements.put ("T1", "5*3+10");
this.statements.put ("T3", "T1");
this.statements.put ("T2", "T1+T3");


this.statements.put ("T5", "4*T2");
this.statements.put ("T6", "4*T2+100");
}
public void putStatements ()
{
for (Map.Entry mapElement:this.statements.entrySet ())
{
String key = (String) mapElement.getKey ();
String value = (String) mapElement.getValue ();
System.out.println (key + " : " + value);
}
System.out.println ("----------------------------------------");
}
public int evaluate (String str)
{
String[]arr = str.split ("\\+");
for (int i = 0; i < arr.length; i++)
{
int result = 1;
if (arr[i].contains ("*"))
{
String[]num = arr[i].split ("\\*");
for (int j = 0; j < num.length; j++)
{
result *= Integer.parseInt (num[j]);
}
arr[i] = String.valueOf (result);
}
}
int len = arr.length;
int sum = 0;
for(int i = 0; i < len; i++)
{
sum += Integer.parseInt (arr[i]);
}
return sum;
}
public void constantFolding ()
{
for (int i = 0; i < this.result.size (); i++)
{
```

```
String lhs = this.result.get (i);
String rhs = this.statements.get (lhs);
Pattern p = Pattern.compile ("[\\d]+([+*][\\d]+)+");

Matcher m = p.matcher (rhs);
while (m.find ())
{
String subexpr = m.group ();
int result = this.evaluate (subexpr);
String res = String.valueOf (result);
rhs = rhs.replace (rhs.substring (m.start (), m.end ()), res);
m = p.matcher (rhs);
}
this.statements.put (lhs, rhs);
}
}
public void variablePropagation ()
{
for (int i = 0; i < this.result.size (); i++)
{
String lhs1 = this.result.get (i);
String rhs1 = this.statements.get (lhs1);
if (rhs1.length () == 1 && this.result.contains (rhs1))
{
for (int j = 0; j < this.result.size (); j++)
{
String lhs2 = this.result.get (j);
String rhs2 = this.statements.get (lhs2);
if (rhs2.contains (lhs1))
{
rhs2 = rhs2.replace (lhs1, rhs1);
this.statements.put (lhs2, rhs2);
}
}
this.result.remove (lhs1);
this.statements.remove (lhs1, rhs1);
}
}
}
public void commonSubexpElim ()
{
for (int i = 0; i < this.result.size (); i++)
{
String lhs1 = this.result.get (i);
String rhs1 = this.statements.get (lhs1);
for (int j = 0; j < this.result.size (); j++)
{
String lhs2 = this.result.get (j);

String rhs2 = this.statements.get (lhs2);
if (lhs1 == lhs2)
{
continue;
}
if (rhs1.contains (rhs2))
{
int start = rhs1.indexOf (rhs2);
int len = rhs2.length ();
rhs1 = rhs1.replace(rhs1.substring(start,len),lhs2);
this.statements.put(lhs1,rhs1);
}
}
}
}
}
```

	/* To check if javac is installed javac --version To run the prog in cmd javac Filename.java java Filename */	

B	Write a LEX program to count the number of tokens with uppercase characters.
	%{
	#include<stdio.h>
	int Upper=0;
	int Lower=0;
	%}
	%%
	[A-Z] {printf("Uppercase\t\n");Upper++;}
	[a-z] {printf("Lowercase\t\n");Lower++;}
	\n {printf("Uppercase=%d and Lowercase=%d",
	Upper,Lower);}
	%%
	int main()
	{
	printf("Enter a string\n");
	yylex();
	}
	int yywrap()
	{ return 1; }
		5 Marks

	Questions	**Marks**

A	Write a program to generate the three address code of
	pi = 3.145;
	x = a * pi * 180 + b * pi * 2;
	
	#include <iostream>
	#include <string>
	#include <stack>
	#include <map>
	#include <vector>
	using namespace std;
		10 Marks

```
int main(){
    string equation,postfix;
    stack<char> stack;
    map <char,int> precedence={{'/',4},{'*',3},{'+',2},{'-',1},{'(',0}};
    cout<<"Enter the equation : ";
    cin>>equation;
    for (int i=0;i<equation.length();i++){
        if (isalpha(equation[i])){
            postfix=postfix+equation[i];
        }
        else{
            if (stack.empty() || equation[i]=='('){
                stack.push(equation[i]);
            }
            else if(equation[i]==')'){
                postfix=postfix+string(1, stack.top());
                stack.pop();
                stack.pop();
            }
            else{
                auto pc=precedence.find(equation[i]);
                auto tc=precedence.find(stack.top());
                while (!stack.empty() && pc-> second <= tc->second){
                    postfix=postfix+string(1, stack.top());
                    stack.pop();
                }
                stack.push(equation[i]);
            }
        }
    }
    while (!stack.empty()){
        postfix=postfix+string(1, stack.top());
        stack.pop();
    }
    int count=0;
    vector<string> var;
    vector<string> tac;
    int i=0,n=postfix.length();
    while(postfix.length()>1){
        if (!isalnum(postfix[i])){
            var.push_back("t"+to_string(count));
            string opr1=string(1,postfix[i-2]);
            string opr2=string(1,postfix[i-1]);
            if (isdigit(postfix[i-2])){
                opr1=var.at(stoi(string(1,postfix[i-2])));
```

```
          }
          if (isdigit(postfix[i-1])){
             opr2=var.at(stoi(string(1,postfix[i-1])));
          }
          tac.push_back(var.at(count)+" = "+opr1+postfix[i]+opr2);
          postfix.replace(i-2,3,to_string(count));
          i=0;
          count+=1;
          n=postfix.length();
          continue;
       }
       i++;
    }
    for (auto elem : tac) {
       cout<<elem<<endl;
    }
    return 0;
}
```

B

Write a LEX program to check valid Mobile Number (10 digit)

5 Marks

```
/* Lex Program to check valid Mobile Number */

%{
   /* Definition section */
%}

/* Rule Section */
%%

[1-9][0-9]{9} {printf("\nMobile Number Valid\n");}

.+ {printf("\nMobile Number Invalid\n");}

%%

int yywrap() {
   return 1;
}

// driver code
int main()
{
   printf("\nEnter Mobile Number : ");
   yylex();
   printf("\n");
   return 0;
```

	}	

	Questions	Marks
A	Write a C/ C++/ Java program to to design lexical analyzer for a language whose grammar is known. LINE □If PHRASE then ACTION. LINE / ∈ PHRASE□NOUN VERB NOUN NOUN□(a-z) * VERB□hate / like ACTION□they NOUN **Input:** “If dogs hate cats then they chase. $” **Output**:(k) (n,1) (v) (n,2) (k) (a) (n,3) (op) Identify and count the number of tokens	10 Marks

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>

#define MAX_SIZE 1000

int is_keyword(char buffer[]) {
char keywords[][10] = {"If", "then", "else"}; int i, flag = 0;
for(i = 0; i < 3; ++i) { if(strcmp(keywords[i], buffer) == 0) {
flag = 1; break;
}
}

return flag;
}
int is_verb(char buffer[]) {
char keywords[][10] = {"hate", "like"}; int i, flag = 0;

for(i = 0; i < 2; ++i) { if(strcmp(keywords[i], buffer) == 0) {
flag = 1; break;
}}

return flag;
}
int is_action(char buffer[]) {
char keywords[][10] = {"they"}; int i, flag = 0;
```

```
for(i = 0; i < 1; ++i) { if(strcmp(keywords[i], buffer) == 0) {
flag = 1; break;

}}
return flag;
}

int main() {
char input[MAX_SIZE] = "If dogs hate cats then they chase . If cats like milk then
they drink . $"; char c, buffer[MAX_SIZE], nouns[MAX_SIZE][MAX_SIZE];
int i, j=0, k=0, n=0, token_count=0; printf("\nInput: %s\n", input); printf("\nTokens:
");
for(i = 0; i < strlen(input); ++i) { c = input[i];
if(isalnum(c)) { buffer[j++] = c;
} else if((c == ' ' || c == '\n' || c == '\t') && (j != 0)) { buffer[j] = '\0';

if(is_keyword(buffer)) { printf("(keyword) ");
} else if(is_verb(buffer)){ printf("(verb)");
}else if(is_action(buffer)){ printf("(a)");
}
else {
int found = 0;
for (int l = 0; l < n; ++l) {
if (strcmp(nouns[l], buffer) == 0) { printf("(noun,%d) ", l+1);
found = 1; break;
}
}
if (!found) { strcpy(nouns[n], buffer); printf("(noun,%d) ", n+1); n++;
}
}
token_count++; j = 0;
} else if(c == '.') {
printf("(op) "); token_count++;
} else if(c == '$') { printf("<eof>\n"); token_count++;
}
}
printf("\nSymbol table:\n\n");
for(i = 0; i < n; ++i) { printf("%s\t", nouns[i]);
}
printf("\n");
```

```
    for(i = 0; i < n; ++i) { printf("[%d]\t", i+1);
    }
    printf("\n\nNumber of tokens: %d\n", token_count);
    return 0;
    }
```

B

Lex program to take check whether the given number is even or odd

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

DIGIT [0-9]
%%
{DIGIT}+ {
  int num = atoi(yytext);
  if(num % 2 == 0) {
    printf("%d is even\n", num);
  }
  else {
    printf("%d is odd\n", num);
  }
}
.|\n {}

%%
int yywrap() {
  return 0;
}

int main() {
  yylex();
  return 0;
}
```

5 Marks