Name: Muhammad Huzaifa
Roll Number : 19B-139-Cs
Section : B
CS412-CompilerLab:01
Date:27/10/2022
Lab Instructor : Sir Faisal

Lab : 03

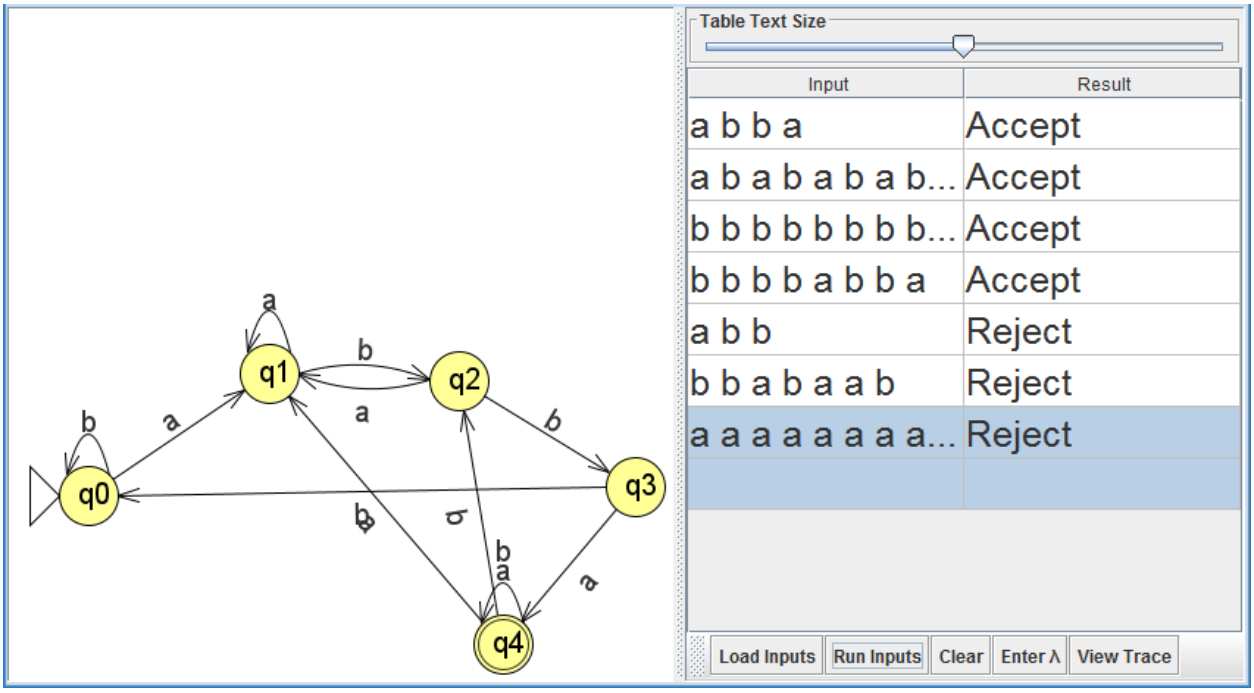
Task 1

Construct the automata machines for any five of the following given conditions of regular expressions:

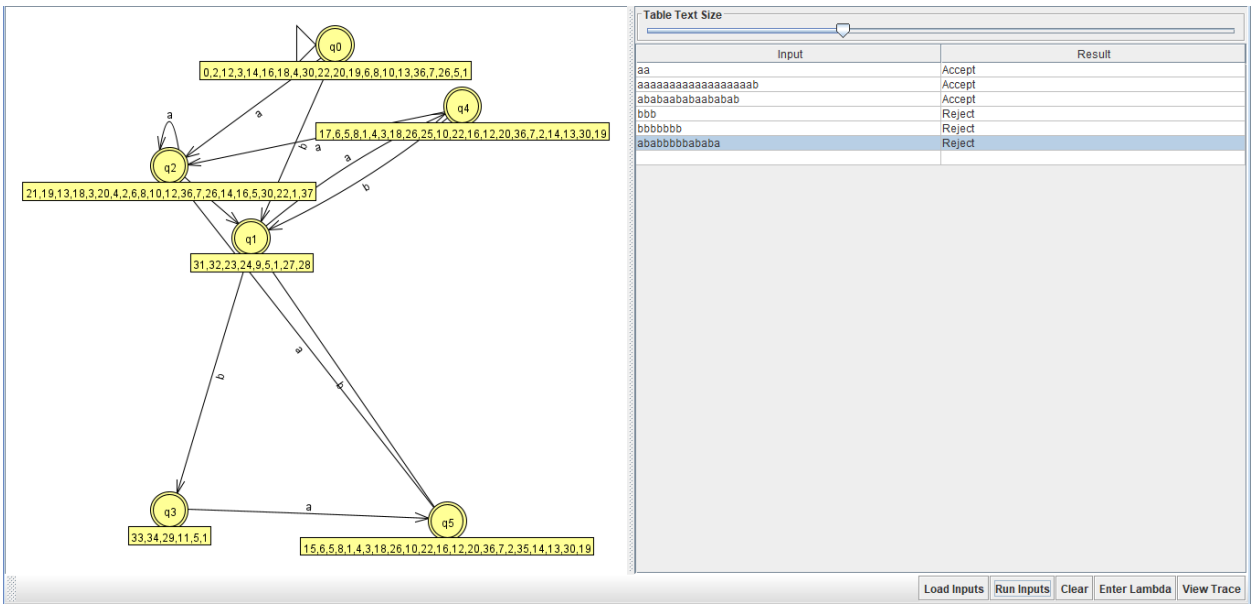
regular language	*regular expression*	*in the language*	*not in the language*
binary alphabet			
fifth-to-last symbol is a	(a|b)*a(a|b)(a|b)(a|b)(a|b)	aaaaa bbbabbbb bbbbbbababababa	a bbbbbbbba aaaaaaaaaaabaaaa
contains the substring abba	(a|b)*abba(a|b)*	abba aababbabbababbba bbbbbbbbabbabbbbb	abb bbabaab aaaaaaaaaaaaaaa
does not contain the substring bbb	(bba|ba|a*)*(a*|b|bb)	aa abababbbaba aaaaaaaaaaaaab	bbb ababbbbabab bbbbbbbbbbbbb
number of b symbols is a multiple of 3	a*|(a*ba*ba*ba*)*	bbb aaa bbbaababbaa	b baaaaaaab baabbbaaaaab
decimal digits			
positive integer divisible by 5	5|(1|2|...|9)(0|1|...|9)*(0|5)	5 200 9836786785	1 0005 3452345234
positive ternary number	(1|2)(0|1|2)*	11 2210221	011 19 9836786785
lowercase letters			
contains the trigraph spb	(a|b|c|...|z)*spb(a|b|c|...|z)	raspberry crispbread	subspace subspecies
uses only the top row of the keyboard	(q|w|e|r|t|y|u|i|o|p)*	typewriter reporter	alfalfa paratrooper
genetic code			
fragile X syndrome pattern	GCG(CGG|AGG)*CTG	GCGCTG GCGCGGCTG GCGCGGAGGCTG	GCGCGG CGGCGGCGGCTG GCGCAGGCTG

1) Contain the substring (a+b)*abba(a+b)*:

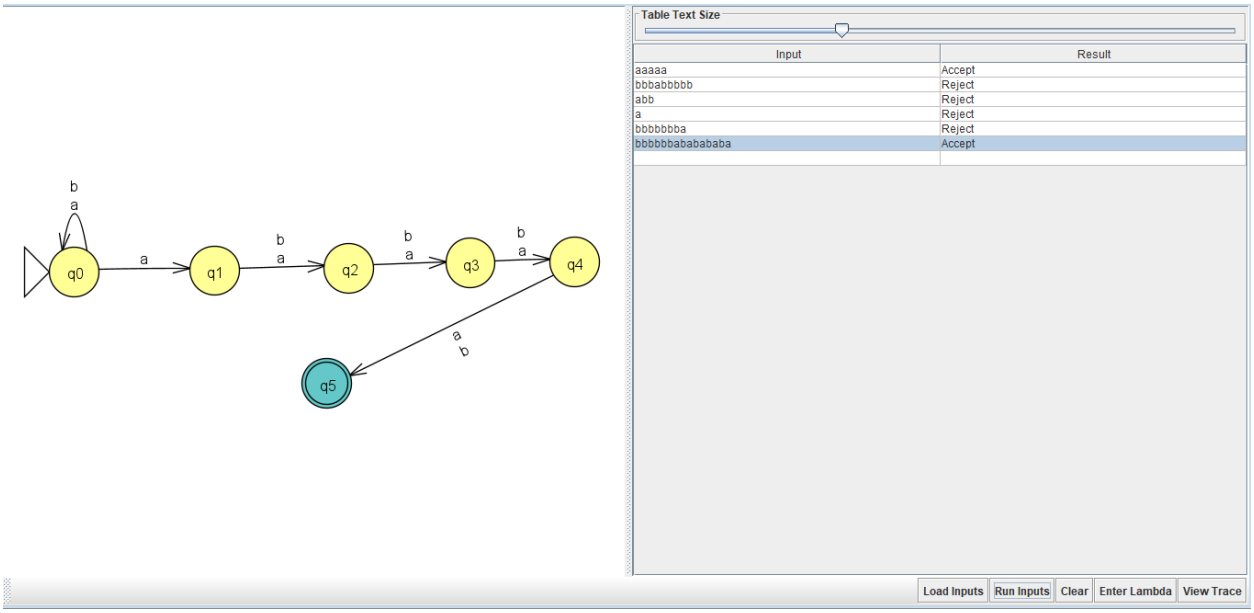
R.E=(a+b)*abba(a+b)*



2) <u>Doesn't conatin the substring R.E=(bba|ba|a*)*(a*|b|bb):</u>



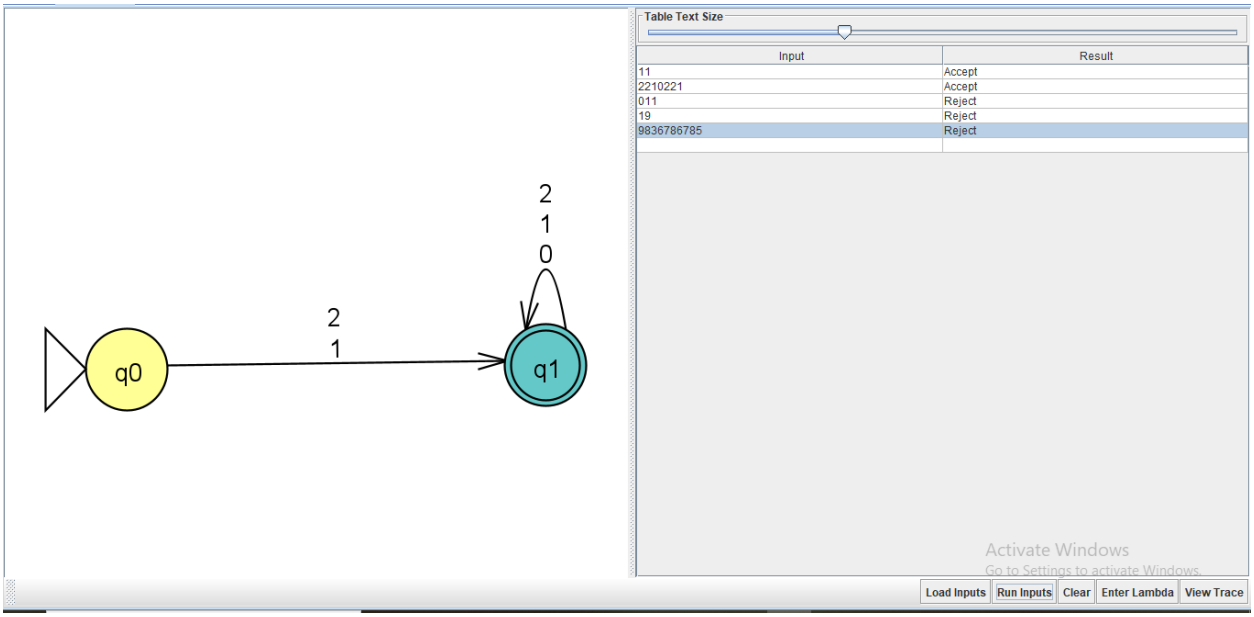
3) Fifth-to-last symbol is a:



4) Decimal Digits:

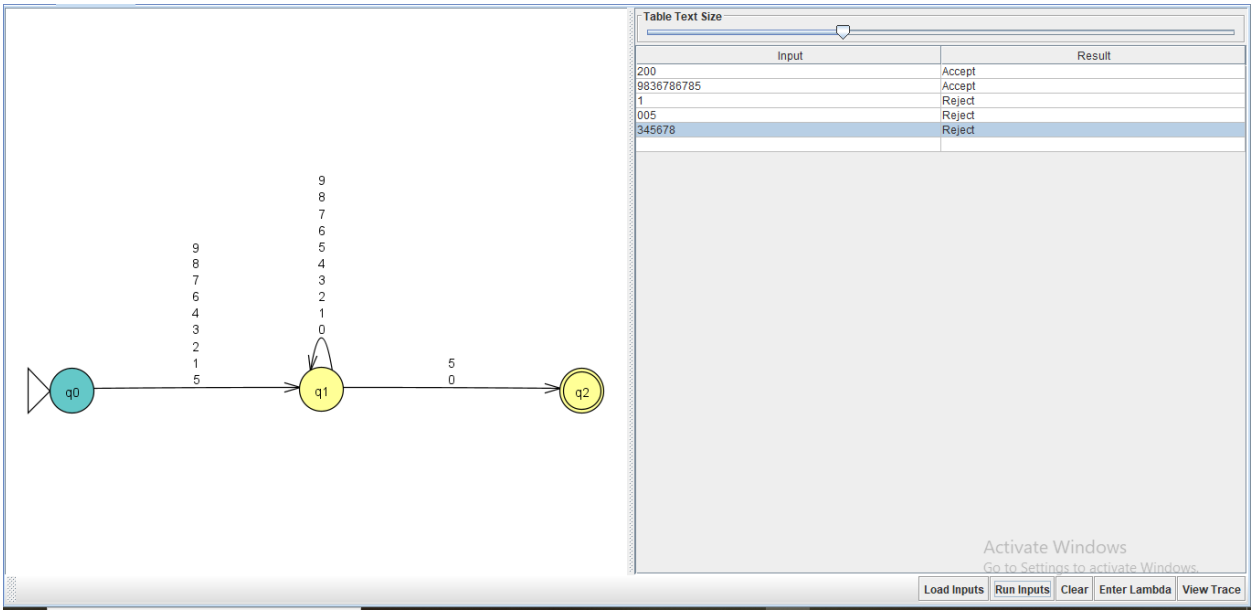
Positive ternary Number:

R.E= (1|2)(0|1|2)*



5) Positive Integer Divisible By 5:

R.E: (5|1-9)(0-9)*(0|5)



4. Homework Tasks:

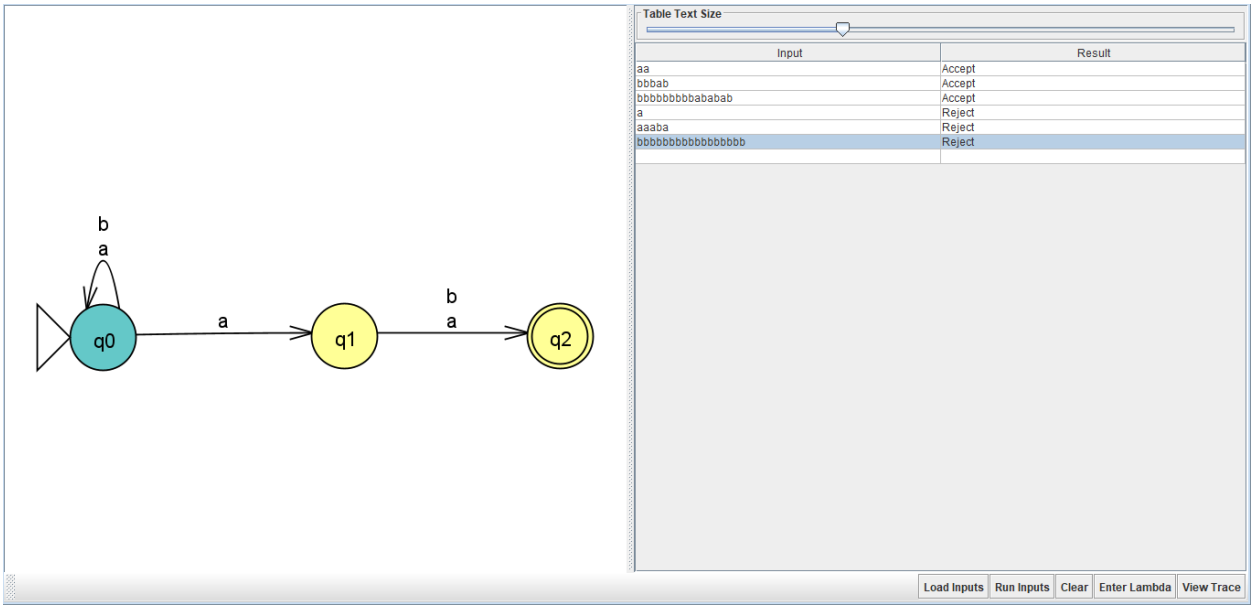
2. Use JFLAP to prove that the above regular expressions are valid.
3. Write a code to check any four regular expressions that are defined in question 1. Generate the right and wrong answer or validate the correct or incorrect input

Combine 2 and 3 :

	in the language	*not in the language*
second-to-last symbol is a	aa bbbab bbbbbbbbbababab	a aaaba bbbbbbbbbbbbb
equal numbers of as and bs	ba bbaaba aaaabbbbbbbaaaba	a bbbaa abababababababa
palindromes	a aba abaabaabaaba	ab bbbba abababababababab
contain the pattern abba	abba abaababbabbababbba bbbbbbbbbbabbabbbbb	abb bbabaab aaaaaaaaaaaaaaaa
number of bs is divisible by 3	bbb baaaaabaaaab bbbabbaaabaaababababaaa	bb abababab aaaaaaaaaaaaaaaab

1) Second-last symbol is a :

R.E is = (a+b)*a(a+b)



Accepted string is =aa,bbbab,bbbbbbbbbbbbabb

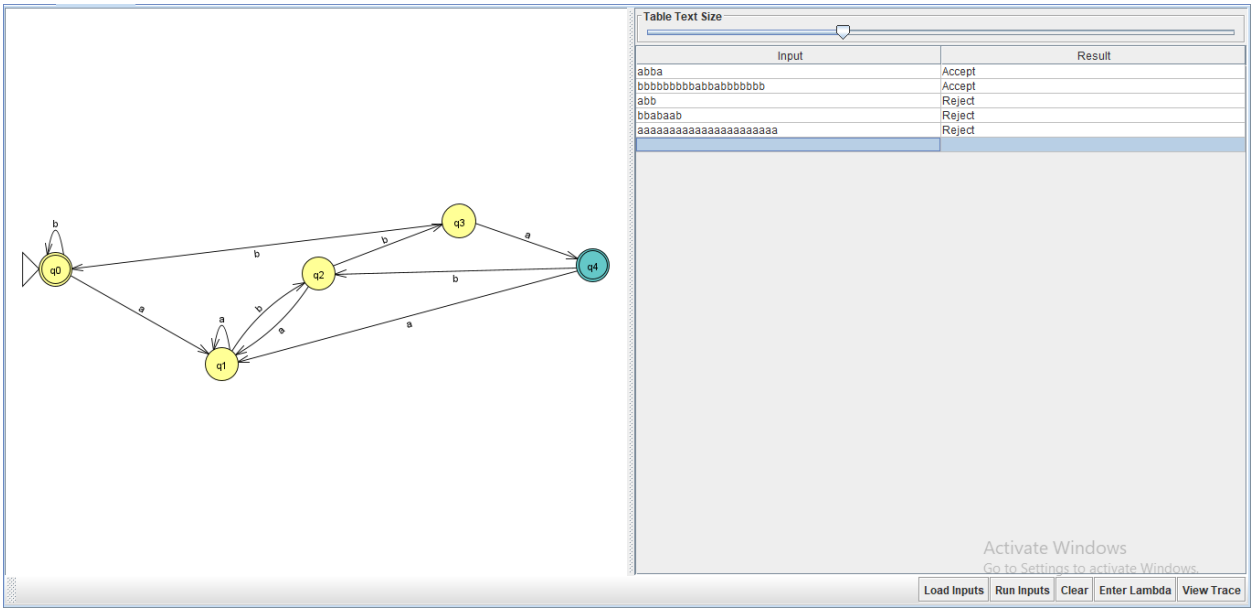
Rejected string is = a,aaaba, bbbbbbbbbbb

2) Equal Number of a's and b's:

R.E = (ab)+

3) Containing the Pattern abba:

R.E = (a+b)*abba

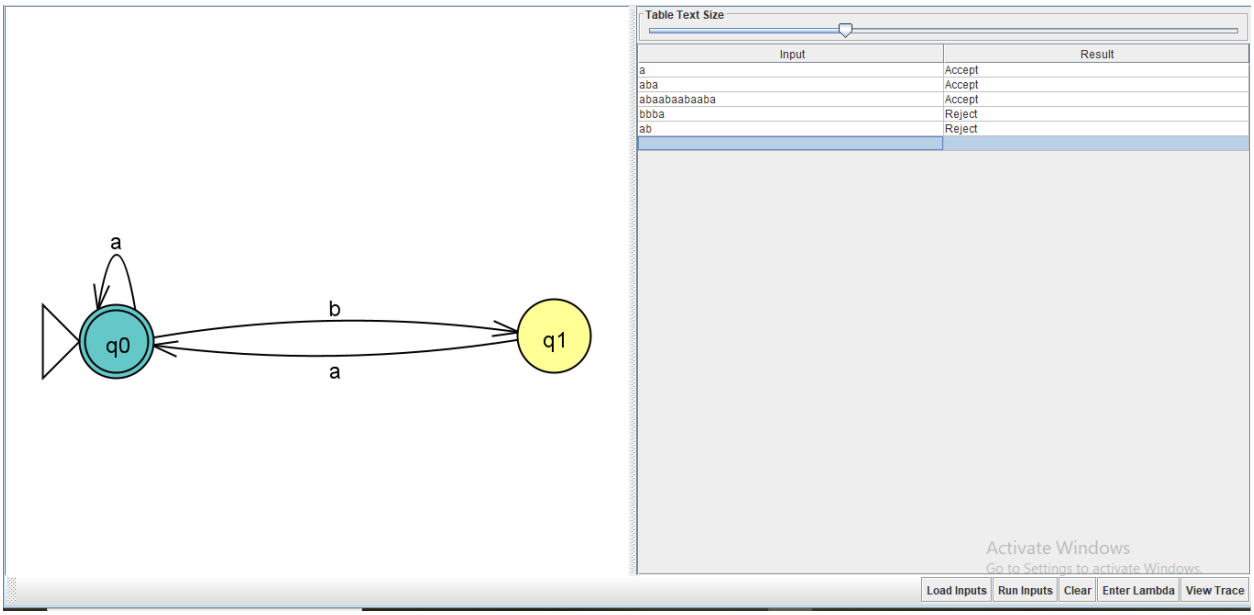


Accepted String : abba, bbbbbbbbbabbabbbbbbb

Rejected String : abb, bbabaab, aaaaaaaaaaaaaaaaaaaa

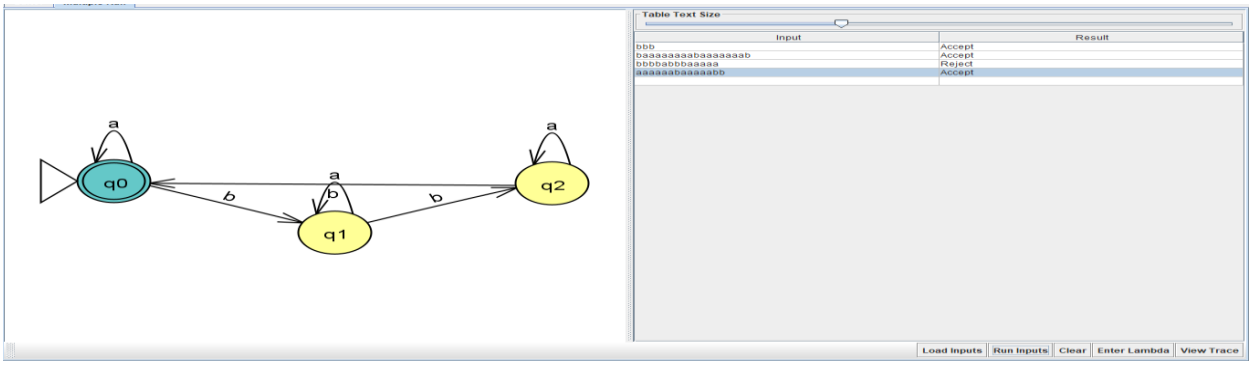
4) Palindromes:

R.E = ((a)*ba)*



5) Number of b's Divisible by 3:

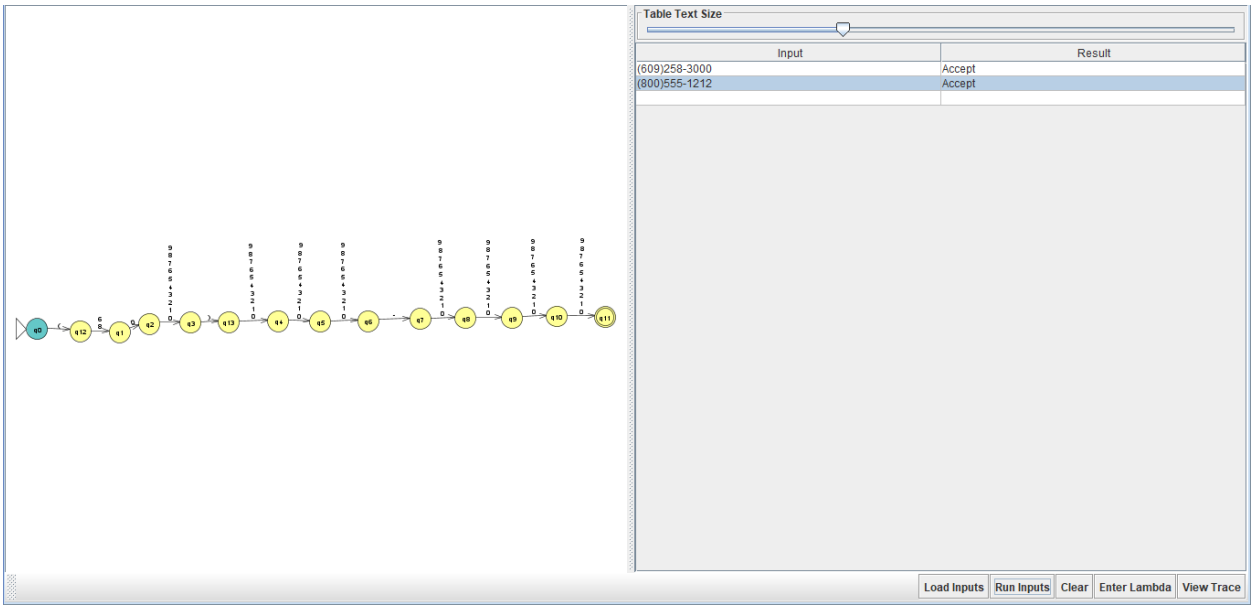
R.E= ((a)*b(a)*b(a)*b)*



	in the language	*not in the language*
palindromes	madamimadam amanaplanacanalpanama amoraroma	madamimbob madam, i'm adam not a palindrome
odd integers	3 101 583805233	2 100 2147483648
prime integers	3 101 583805233	0003 100 2147483648
integers z such that $x^2 + y^2 = z^2$ for some integers x, y	5 13 9833	2 16 9999
integers z such that $x^n + y^n = z^n$ for some integers x, y, n > 2	no integers	all integers
amino acid encodings	AAA ACA ATA AGA CAA CCA CTA CGA TCA TTA GAA GCA GTA GGA AAC ACC	CCC AAAAAAAAA ABCDE
U.S. telephone number	(609) 258-3000 (800) 555-1212	(99) 12-12-12 2147483648
English words	and middle computability	abc niether misunderestimate
legal English sentences	This is a sentence. I think I can.	xya a b.c.e?? Cogito ergo sum.
legal Java identifiers	a class $xyz3_XYZ	12 123a a((BC))*
legal Java programs	public class Hi { public static void main(String[] args) { } }	int main(void) { return 0; }

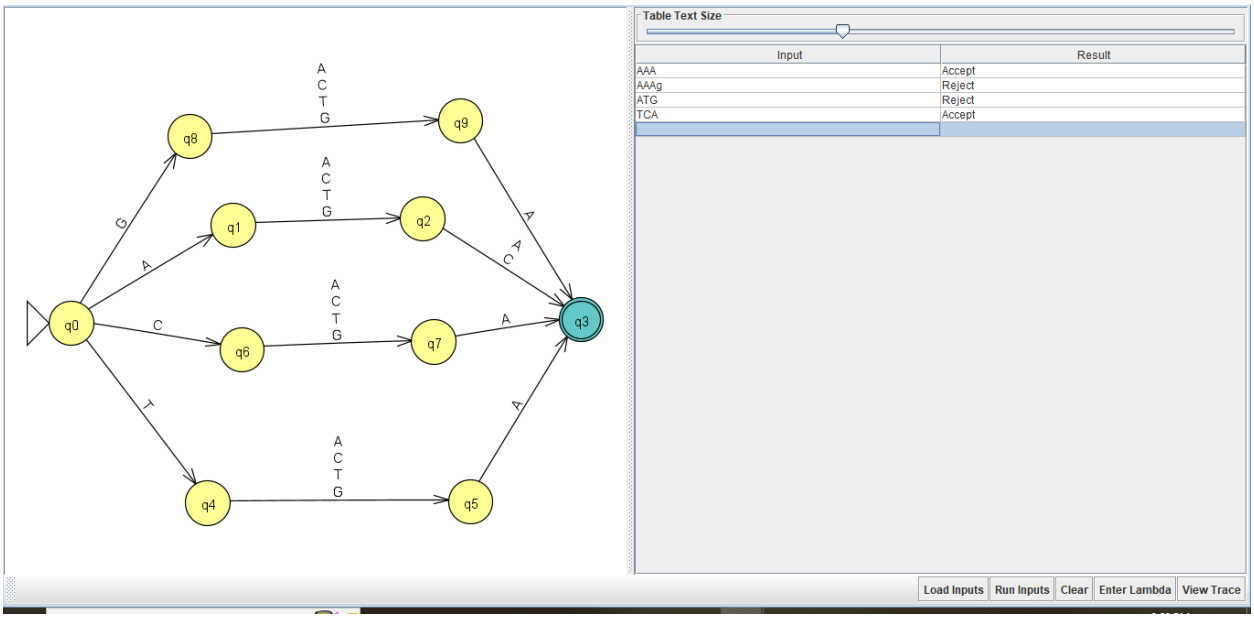
6) U.S Telephone Number :

R.E =
((8+6)0(0+1+2+3+4+5+6+7+8+9))(0+1+2+3+4+5+6+7+8+9)(0+1+2+3+4+5+6+7+8+9)(0+1+2+3+4+5+6+7+8+9)(0+1+2+3+4+5+6+7+8+9)(0+1+2+3+4+5+6+7+8+9)(0+1+2+3+4+5+6+7+8+9)(0+1+2+3+4+5+6+7+8+9)



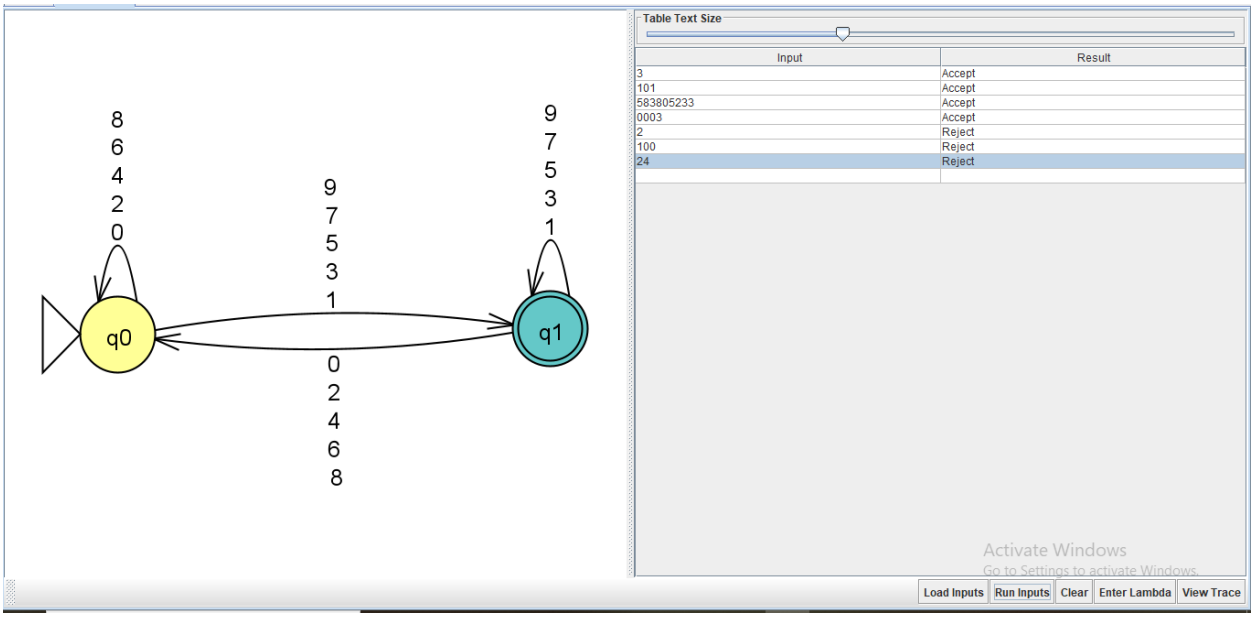
7) Amino Acid:

R.E : A(G+T+C+A)(C+A)+T(G+T+C+A)A+C(G+T+C+A)A+G(G+T+C+A)A



Old Integers:

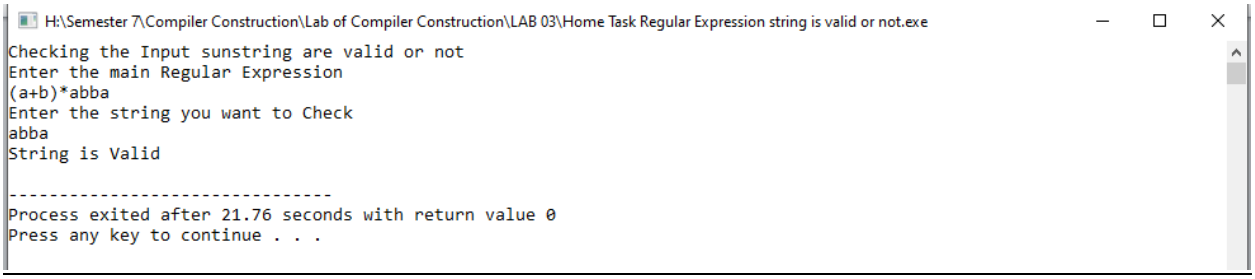
R.E: ((0+2+4+6+8)*(1+3+5+7+9)(1+3+5+7+9)*(0+2+4+6+8))*(0+2+4+6+8)*(1+3+5+7+9)(1+3+5+7+9)*

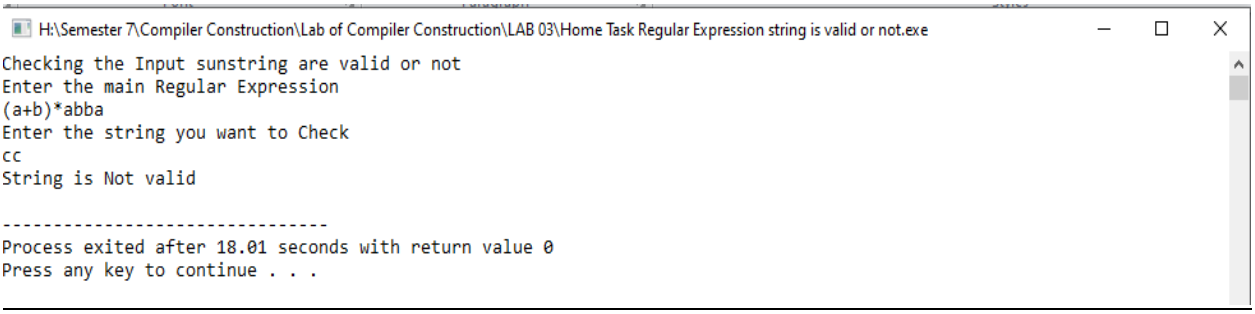


Code to verify the string is valid or not

1) Containing the Pattern abba:

R.E=(a+b)*abba





2) Second-last symbol is a :

R.E is = (a+b)*a(a+b)

```
H:\Semester 7\Compiler Construction\Lab of Compiler Construction\LAB 03\Home Task Regular Expression string is valid or not.exe
Checking the Input sunstring are valid or not
Enter the main Regular Expression
(a+b)*a(a+b)
Enter the string you want to Check
a
String is Valid

--------------------------------
Process exited after 14.65 seconds with return value 0
Press any key to continue . . .
```

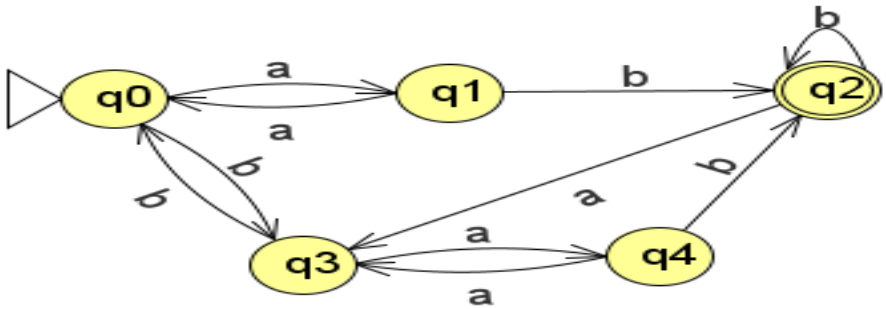
```
H:\Semester 7\Compiler Construction\Lab of Compiler Construction\LAB 03\Home Task Regular Expression string is valid or not.exe
Checking the Input sunstring are valid or not
Enter the main Regular Expression
(a+b)*a(a+b)
Enter the string you want to Check
ab
String is Not valid

--------------------------------
Process exited after 92.72 seconds with return value 0
Press any key to continue . . .
```

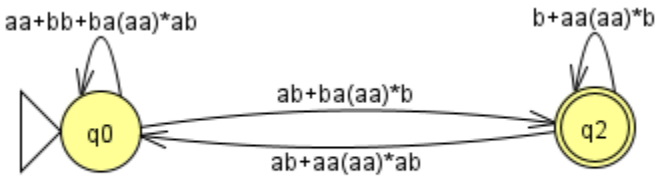
3) **Number of b's Divisible by 3:**

R.E= ((a)*b(a)*b(a)*b)*

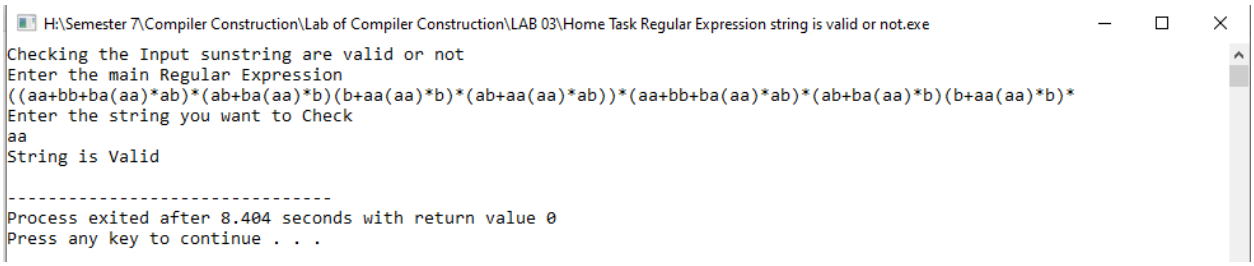
```
H:\Semester 7\Compiler Construction\Lab of Compiler Construction\LAB 03\Home Task Regular Expression string is valid or not.exe
Checking the Input sunstring are valid or not
Enter the main Regular Expression
((a)*b(a)*b(a)*b)*
Enter the string you want to Check
a*
String is Valid

--------------------------------
Process exited after 8.613 seconds with return value 0
Press any key to continue . . .
```

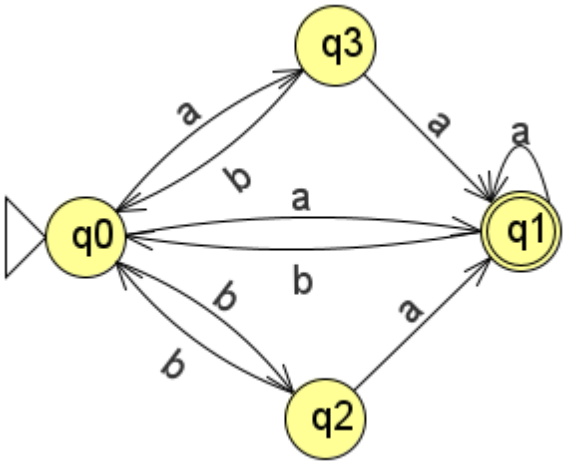
4. For the following FA's find the regular expression and use regex to code.

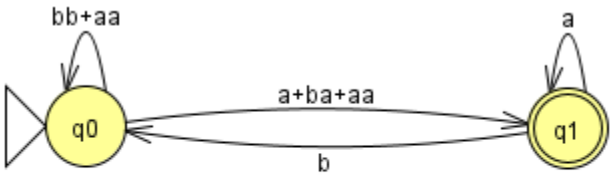




R.E=((aa+bb+ba(aa)*ab)*(ab+ba(aa)*b)(b+aa(aa)*b)*(ab+aa(aa)*ab))*(aa+bb+ba(aa)*ab)*(ab+ba(aa)*b)(b+aa(aa)*b)*







R.E=((bb+aa)*(a+ba+aa)a*b)*(bb+aa)*(a+ba+aa)a*

```
H:\Semester 7\Compiler Construction\Lab of Compiler Construction\LAB 03\Home Task Regular Expression string is valid or not.exe
Checking the Input sunstring are valid or not
Enter the main Regular Expression
((bb+aa)*(a+ba+aa)a*b)*(bb+aa)*(a+ba+aa)a*
Enter the string you want to Check
a+ba+aa
String is Not valid

--------------------------------
Process exited after 22.85 seconds with return value 0
Press any key to continue . . .
```

Lab Task :

Program to use regex_search() for finding the pattern in a given string with other parameters.

```
int main()
{
    string s="Some people, when confronted with a problem, think"
    "\"I know, I'll use regular expressions.\" "
    "Now they have two problems.";

    regex self_regex("REGULAR EXPRESSIONS", regex_constants::ECMAScript | regex_constants::icase);
    if (regex_search(s, self_regex)){
        cout<<"Text contains the phrase 'regular expressions'\n";
    }

    regex word_regex("(\\w+)");
    auto words_begin =
        sregex_iterator(s.begin(), s.end(), word_regex);
    auto words_end = sregex_iterator();


    cout <<"Found"
         << distance(words_begin, words_end)
         << "words\n";

    const int N = 6;
    cout << "Words longer than " << N << " characters:\n";
    for (sregex_iterator i = words_begin; i != words_end; ++i){
        smatch match = *i;
        string match_str = match.str();
        if (match_str.size() > N){
            cout << " " << match_str << '\n';
        }
    }

    regex long_word_regex("(\\w{7,})");
    string new_s = regex_replace(s, long_word_regex, "[$&]");
    cout << new_s << '\n';

    return 0;
}
```

```
H:\Semester 7\Compiler Construction\Lab of Compiler Construction\LAB 03\Task 1 find the regular expression in the given string.exe
Text contains the phrase 'regular expressions'
Found20words
Words longer than 6 characters:
 confronted
 problem
 regular
 expressions
 problems
Some people, when [confronted] with a [problem], think"I know, I'll use [regular] [expressions]." Now they have two [pro
blems].

--------------------------------
Process exited after 0.1739 seconds with return value 0
Press any key to continue . . .
```

i i. Write your observations after executing this piece of code.
ii ii. What is the purpose of using [$&], \\w{7, } and \\w+.

1) **Find total word in the sentences and find the length of each word and return those word whose length is greater than 6 character.**
2) **[$&] used for making starting and ending of the string.\\w{7,} marking those word which has length 7 and above of it. \\w+ for checking the word.**

Program to demonstrate the working of regex_match() for a given string.

```
#include <iostream>
#include <regex>

using namespace std;
int main()
{
    string a = "Pakistan is my Love,I Love Pakistan ";

    // Here b is an object of regex (regular expression)
    regex b("(Pakistan)(.*)"); // pakistan followed by any character

    // regex_match function matches string a against regex b
    if ( regex_match(a, b) )
        cout << "String 'a' matches regular expression 'b' \n";

    // regex_match function for matching a range in string
    // against regex b
    if ( regex_match(a.begin(), a.end(), b) )
        cout << "String 'a' matches with regular expression "
             "'b' in the range from 0 to string end\n";

    return 0;
}
```

```
H:\Semester 7\Compiler Construction\Lab of Compiler Construction\LAB 03\Task 3 matching match.exe
String 'a' matches regular expression 'b'
String 'a' matches with regular expression 'b' in the range from 0 to string end

--------------------------------
Process exited after 0.09287 seconds with return value 0
Press any key to continue . . .
```

After executing the above code, answer the followings:
i i. What is the purpose of using regex_match?

ii ii. What is the purpose of using a.begin() and a.end() in this code?

Answer:

1) **Regex_match purpose to check the existing word is present in the string or not**
2) **A.begin() check the first word and a.end() check the end word of the string.**

Program of using regex_replace()

```cpp
// C++ program to demonstrate working of regex_replace()
#include <iostream>
#include <string>
#include <regex>
#include <iterator>
using namespace std;

int main()
{
    string s = "I am looking for Pakistanfamily \n";

    // matches words beginning by "Geek"
    regex r("Pakistan[a-zA-z]+");

    // regex_replace() for replacing the match with 'geek'
    cout << std::regex_replace(s, r, "friend");

    string result;

    // regex_replace( ) for replacing the match with 'geek'
    regex_replace(back_inserter(result), s.begin(), s.end(),
                  r, "friend");

    cout << result;

    return 0;
}
```

```
H:\Semester 7\Compiler Construction\Lab of Compiler Construction\LAB 03\Task 4 regex_replace.exe
I am looking for friend
I am looking for friend

--------------------------------
Process exited after 0.114 seconds with return value 0
Press any key to continue . . .
```

Another C++ program to demonstrate the regex_replace() function.

```cpp
#include <iostream>
#include <string>
#include <regex>
#include <iterator>
using namespace std;

int main()
{
    string mystr = "This is software testing Help portal \n";

    cout<<"Input string: "<<mystr<<endl;

    regex regexp("p[a-zA-z]+");
    cout << "Replace the word 'portal' with word 'website': ";
    cout << regex_replace(mystr, regexp, "website");

    string result;

    cout << "Replace the word 'website' back to 'portal' : ";
    regex_replace(back_inserter(result), mystr.begin(), mystr.end(), regexp, "portal");

    cout << result;

    return 0;
}
```

```
H:\Semester 7\Compiler Construction\Lab of Compiler Construction\LAB 03\Task 6.exe
Input string: This is software testing Help portal

Replace the word 'portal' with word 'website': This is software testing Help website
Replace the word 'website' back to 'portal' : This is software testing Help portal

--------------------------------
Process exited after 0.1271 seconds with return value 0
Press any key to continue . . .
```

Program to demonstrate the validation of incoming data.

```cpp
#include <iostream>
#include <regex>
#include <string>
#include <iterator>

using namespace std;

int main(){
    string input;
    regex integer_expr("(\\+|-)?[[:digit:]]+");
    while(true){
        cout<<"Enter the input: ";
        cin>>input;
        if(!cin){
            break;
        }
        if(regex_match(input, integer_expr)){
            cout<<"Input is an integer"<<endl;
        }
        else{
            cout<<"Invalid input : Not an integer"<<endl;
        }
    }
    return 0;
}
```