

Madhav Institute of Technology & Science, Gwalior

(A Govt. Aided UGC Autonomous NAAC Accredited Institute Affiliated to RGPV Bhopal)

DEPARTMENT OF INFORMATION TECHNOLOGY



PRACTICAL FILE THEORY OF COMPUTATION

(240502)

SUBMITTED TO:

Prof. Abhilash Sonkar
(Assistant Professor)

SUBMITTED BY:

Janmejey Joshi
(0901AI201027)

Madhav Institute of Technology & Science, Gwalior

(A Govt. Aided UGC Autonomous NAAC Accredited Institute Affiliated to RGPV Bhopal)

Program : 1

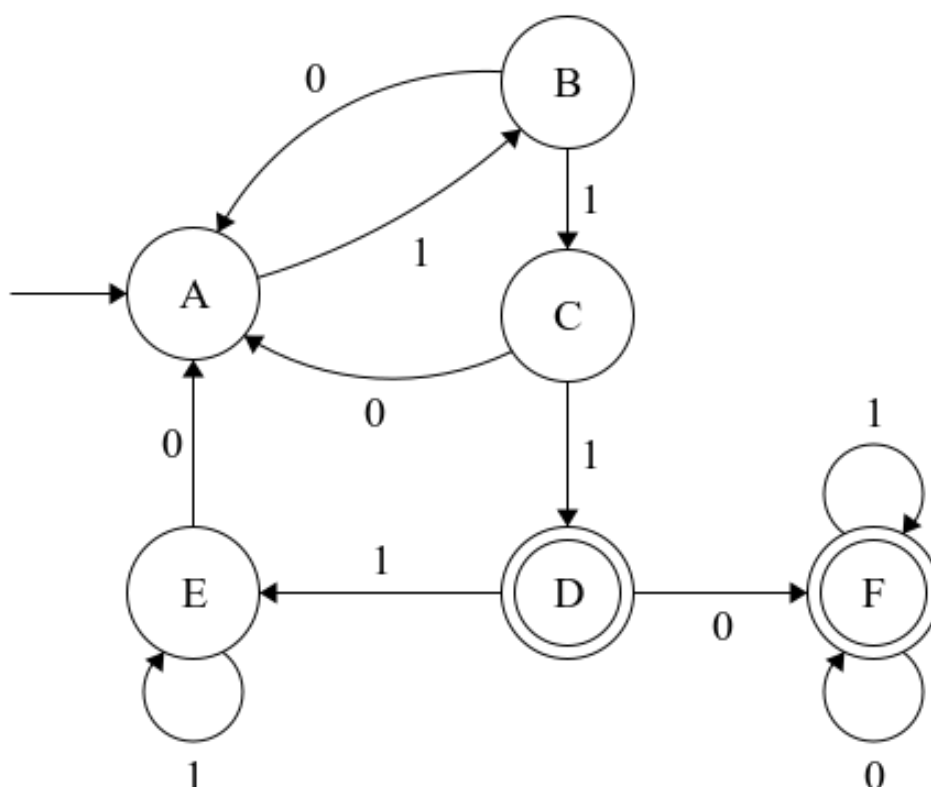
Aim :

Design a Program for creating machine that accepts three consecutive one.

Automata Table :

States	0	1
A	A	B
B	A	C
C	A	D
D	F	E
E	A	E
F	F	F

Automata Diagram :



Program :

```
#include <iostream>
#include <string>

bool has_3_ones(std::string input) {
    char state = 'a';

    for (auto &i : input) {
        switch (state) {

            case 'a':
                if (i == '1')
                    state = 'b';
                else if (i == '0')
                    state = 'a';
                break;

            case 'b':
                if (i == '1')
                    state = 'c';
                else if (i == '0')
                    state = 'a';
                break;

            case 'c':
                if (i == '1')
                    state = 'd';
                else if (i == '0')
                    state = 'a';
                break;

            case 'd':
                if (i == '1')
                    state = 'e';
                else if (i == '0')
                    state = 'f';
                break;

            case 'e':
                if (i == '1')
                    state = 'e';
                else if (i == '0')
                    state = 'a';
                break;

            case 'f':
                if (i == '1')
                    state = 'f';
                else if (i == '0')
                    state = 'f';
                break;
        }
    }
    return state == 'd' || state == 'f';
}

int main(int argc, char *argv[]) {
    std::string input;
    std::cout << "Enter String : ";
    std::cin >> input;
    std::cout << has_3_ones(input);

    return 0;
}
```

Madhav Institute of Technology & Science, Gwalior

(A Govt. Aided UGC Autonomous NAAC Accredited Institute Affiliated to RGPV Bhopal)

Program : 2

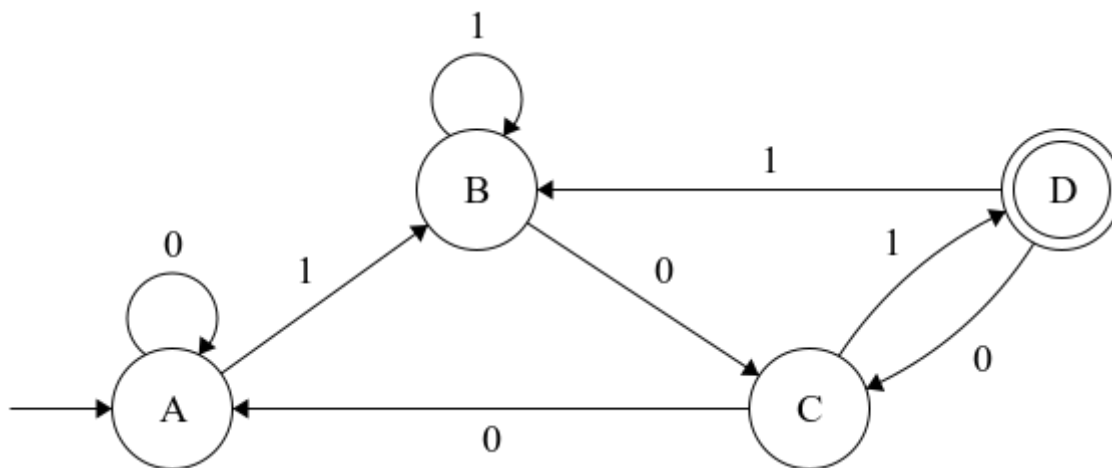
Aim :

Design a Program for creating machine that accepts the string ending with 101.

Automata Table :

States	0	1
A	A	B
B	C	B
C	A	D
D	C	B

Automata Diagram :



Program :

```
#include <iostream>
#include <string>

bool has_suffix(std::string input) {
    char state = 'a';

    for (auto &i : input) {
        switch (state) {

            case 'a':
                if (i == '1')
                    state = 'b';
                else if (i == '0')
                    state = 'a';
                break;

            case 'b':
                if (i == '1')
                    state = 'b';
                else if (i == '0')
                    state = 'c';
                break;

            case 'c':
                if (i == '1')
                    state = 'd';
                else if (i == '0')
                    state = 'a';
                break;

            case 'd':
                if (i == '1')
                    state = 'b';
                else if (i == '0')
                    state = 'c';
                break;
        }
    }

    return state == 'd';
}

int main(int argc, char *argv[]) {
    std::string input;

    std::cout << "Enter String : ";
    std::cin >> input;

    std::cout << has_suffix(input);

    return 0;
}
```

Madhav Institute of Technology & Science, Gwalior

(A Govt. Aided UGC Autonomous NAAC Accredited Institute Affiliated to RGPV Bhopal)

Program : 3

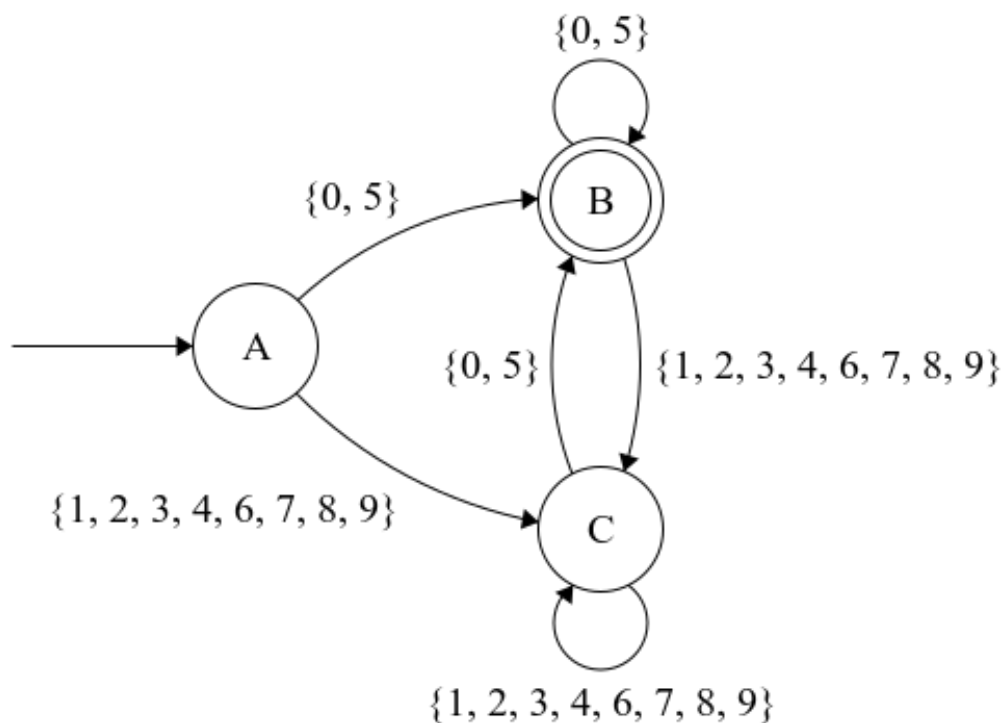
Aim :

Design a program for accepting decimal number divisible by 5.

Automata Table :

States	{0, 5}	{1, 2, 3, 4, 6, 7, 8, 9}
A	B	C
B	B	C
C	B	C

Automata Diagram :



Program :

```
#include <iostream>
#include <string>

bool is_divisible(std::string input) {
    char state = 'a';

    for (auto &i : input) {
        switch (state) {

            case 'a':
                if (i == '0' || i == '5')
                    state = 'b';
                else if (i == '1' || i == '2' || i == '3' || i == '4'
                        || i == '6' || i == '7' || i == '8' || i == '9')
                    state = 'c';
                break;

            case 'b':
                if (i == '0' || i == '5')
                    state = 'b';
                else if (i == '1' || i == '2' || i == '3' || i == '4'
                        || i == '6' || i == '7' || i == '8' || i == '9')
                    state = 'c';
                break;

            case 'c':
                if (i == '0' || i == '5')
                    state = 'b';
                else if (i == '1' || i == '2' || i == '3' || i == '4'
                        || i == '6' || i == '7' || i == '8' || i == '9')
                    state = 'c';
                break;

        }
    }

    return state == 'b';
}

int main(int argc, char *argv[]) {
    std::string input;

    std::cout << "Enter String : ";
    std::cin >> input;

    std::cout << is_divisible(input);

    return 0;
}
```

Madhav Institute of Technology & Science, Gwalior

(A Govt. Aided UGC Autonomous NAAC Accredited Institute Affiliated to RGPV Bhopal)

Program : 4

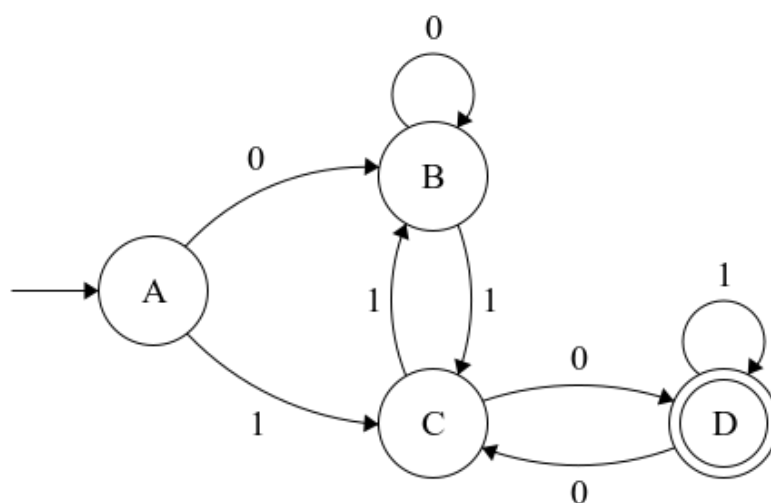
Aim :

Design a Program for creating machine, which accepts 2 Mod 3.

Automata Table :

States	0	1
A	B	C
B	B	C
C	D	B
D	C	D

Automata Diagram :



Program :

```
#include <iostream>
#include <string>

bool has_even_ones_zeros(std::string input) {
    char state = 'a';

    for (auto &i : input) {
        switch (state) {

            case 'a':
                if (i == '1')
                    state = 'c';
                else if (i == '0')
                    state = 'b';
                break;

            case 'b':
                if (i == '1')
                    state = 'c';
                else if (i == '0')
                    state = 'b';
                break;

            case 'c':
                if (i == '1')
                    state = 'b';
                else if (i == '0')
                    state = 'd';
                break;

            case 'd':
                if (i == '1')
                    state = 'd';
                else if (i == '0')
                    state = 'c';
                break;
        }
    }

    return state == 'd';
}

int main(int argc, char *argv[]) {
    std::string input;

    std::cout << "Enter String : ";
    std::cin >> input;

    std::cout << has_even_ones_zeros(input);

    return 0;
}
```

Madhav Institute of Technology & Science, Gwalior

(A Govt. Aided UGC Autonomous NAAC Accredited Institute Affiliated to RGPV Bhopal)

Program : 5

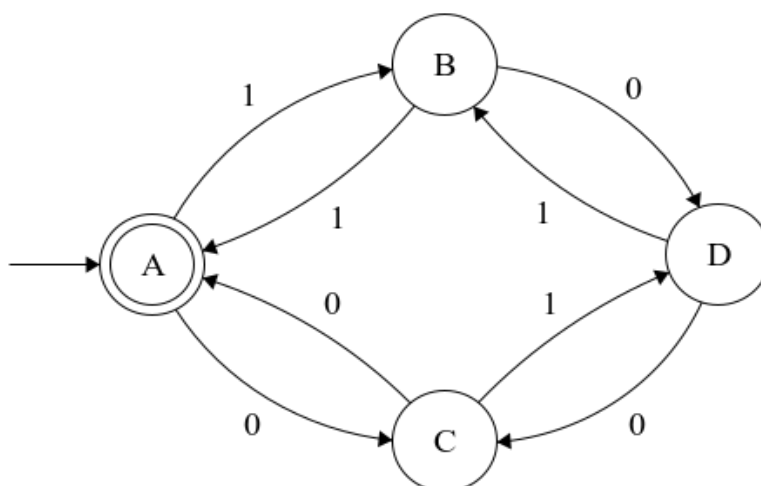
Aim :

Design a program for creating a machine, which accepts even of 1's and 0's.

Automata Table :

States	0	1
A	B	C
B	B	C
C	D	B
D	C	D

Automata Diagram :



Program :

```
#include <iostream>
#include <string>

bool has_even_ones_zeros(std::string input) {
    char state = 'a';

    for (auto &i : input) {
        switch (state) {

            case 'a':
                if (i == '1')
                    state = 'b';
                else if (i == '0')
                    state = 'c';
                break;

            case 'b':
                if (i == '1')
                    state = 'a';
                else if (i == '0')
                    state = 'd';
                break;

            case 'c':
                if (i == '1')
                    state = 'd';
                else if (i == '0')
                    state = 'a';
                break;

            case 'd':
                if (i == '1')
                    state = 'b';
                else if (i == '0')
                    state = 'c';
                break;
        }
    }

    return state == 'a';
}

int main(int argc, char *argv[]) {
    std::string input;

    std::cout << "Enter String : ";
    std::cin >> input;

    std::cout << has_even_ones_zeros(input);

    return 0;
}
```

Madhav Institute of Technology & Science, Gwalior

(A Govt. Aided UGC Autonomous NAAC Accredited Institute Affiliated to RGPV Bhopal)

Program : 6

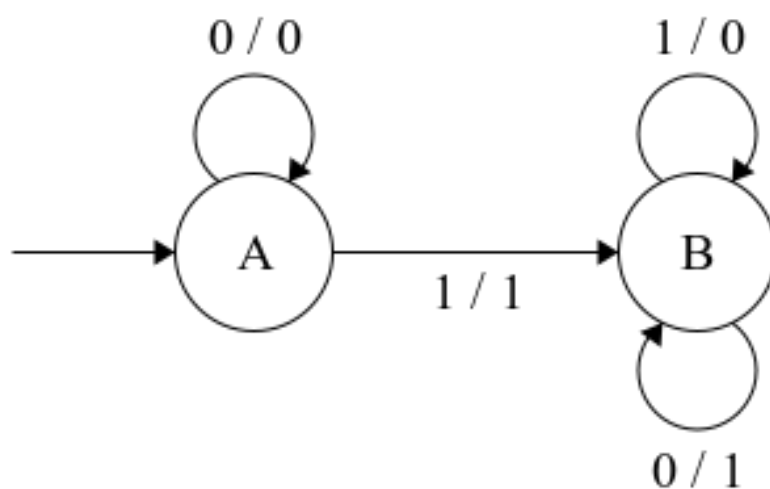
Aim :

Design a Program to find 2's complement of a given binary number.

Automata Table :

States	0	O/P	1	O/P
A	A	0	B	1
B	B	1	B	0

Automata Diagram :



Program :

```
#include <iostream>
#include <string>

std::string complement_2s(std::string input) {
    std::string output = "";
    char state = 'a';

    for (int i = input.size() - 1; i >= 0; i--) {
        switch (state) {

            case 'a':
                if (input[i] == '1') {
                    state = 'b';
                    output = "1" + output;
                } else if (input[i] == '0') {
                    state = 'a';
                    output = "0" + output;
                }
                break;

            case 'b':
                if (input[i] == '1') {
                    state = 'b';
                    output = "0" + output;
                } else if (input[i] == '0') {
                    state = 'b';
                    output = "1" + output;
                }
                break;
        }
    }

    return output;
}

int main(int argc, char *argv[]) {
    std::string input;

    std::cout << "Enter String : ";
    std::cin >> input;

    std::cout << complement_2s(input) << std::endl;

    return 0;
}
```

Madhav Institute of Technology & Science, Gwalior

(A Govt. Aided UGC Autonomous NAAC Accredited Institute Affiliated to RGPV Bhopal)

Program : 7

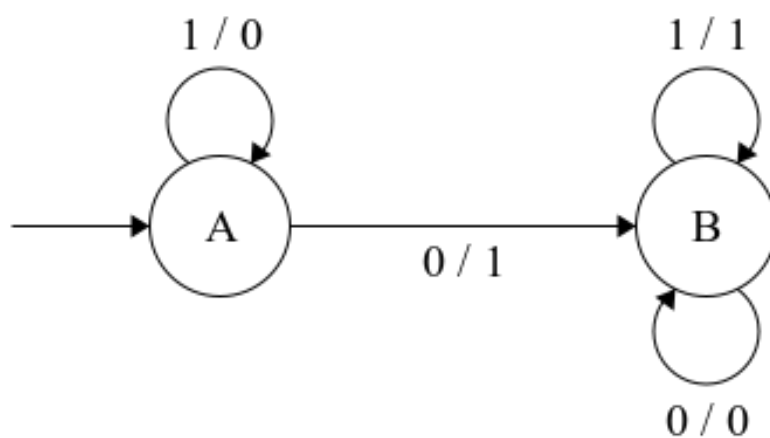
Aim :

Design a Program, which will increment the given binary number by 1.

Automata Table :

States	0	O/P	1	O/P
A	A	1	B	0
B	B	0	B	1

Automata Diagram :



Program :

```
#include <iostream>
#include <string>

std::string increment(std::string input) {
    std::string output = "";
    char state = 'a';

    for (int i = input.size() - 1; i >= 0; i--) {
        switch (state) {

            case 'a':
                if (input[i] == '1') {
                    state = 'a';
                    output = "0" + output;
                } else if (input[i] == '0') {
                    state = 'b';
                    output = "1" + output;
                }
                break;

            case 'b':
                if (input[i] == '1') {
                    state = 'b';
                    output = "1" + output;
                } else if (input[i] == '0') {
                    state = 'b';
                    output = "0" + output;
                }
                break;
        }
    }

    return output;
}

int main(int argc, char *argv[]) {
    std::string input;

    std::cout << "Enter String : ";
    std::cin >> input;

    std::cout << increment(input) << std::endl;

    return 0;
}
```

Madhav Institute of Technology & Science, Gwalior

(A Govt. Aided UGC Autonomous NAAC Accredited Institute Affiliated to RGPV Bhopal)

Program : 9

Aim :

Design a program to create PDA to accept $a^n b^n$ where $n > 0$.

Automata Table :

Input	<i>a</i>				<i>b</i>				ϵ			
Stack	a	b	\$	ϵ	a	b	\$	ϵ	a	b	\$	ϵ
q ₀												{(q ₁ , \$)}
q ₁	{(q ₁ , ϵ)}				{(q ₂ , ϵ)}							
q ₂					{(q ₂ , ϵ)}	{(q ₂ , b)}					{(q ₃ , ϵ)}	
q ₃												

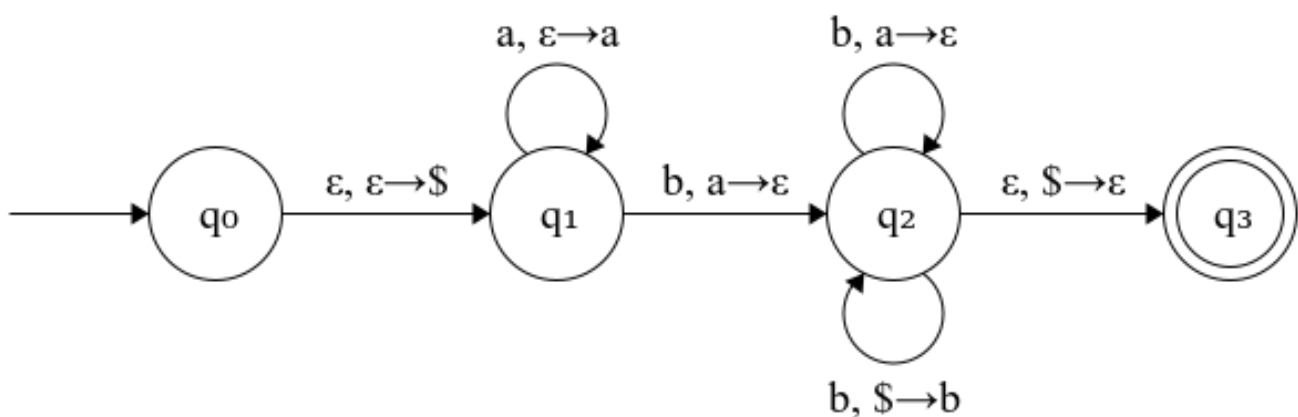


Empty Cells represent ϕ



\$ Symbolises to the Bottom of Stack

Automata Diagram :



Program :

```
#include <iostream>
#include <stack>
#include <string>

bool has_m_ab(std::string input) {
    std::stack<char> pda_stack;
    char state = '1';

    // State q_0
    state = '1';
    pda_stack.push('$');

    for (auto &i : input) {
        switch (state) {

            // State q_1
            case '1':
                if (i == 'a') {
                    state = '1';
                    pda_stack.push('a');
                } else if (i == 'b' && pda_stack.top() == 'a') {
                    state = '2';
                    pda_stack.pop();
                } else if (i == 'b' && pda_stack.top() == '$') {
                    state = '2';
                    pda_stack.push('b');
                }
                break;

            // State q_2
            case '2':
                if (i == 'b' && pda_stack.top() == 'a') {
                    state = '2';
                    pda_stack.pop();
                } else if (i == 'b' && pda_stack.top() == '$') {
                    state = '2';
                    pda_stack.push('b');
                }
                break;
        }
    }

    // State q_3
    if (pda_stack.top() == '$')
        return true;
    else
        return false;
}

int main(int argc, char *argv[]) {
    std::string input;

    std::cout << "Enter String : ";
    std::cin >> input;

    std::cout << has_m_ab(input);

    return 0;
}
```

Madhav Institute of Technology & Science, Gwalior

(A Govt. Aided UGC Autonomous NAAC Accredited Institute Affiliated to RGPV Bhopal)

Program : 10

Aim :

Design a Program to create PDA machine that accept the well-formed parenthesis.

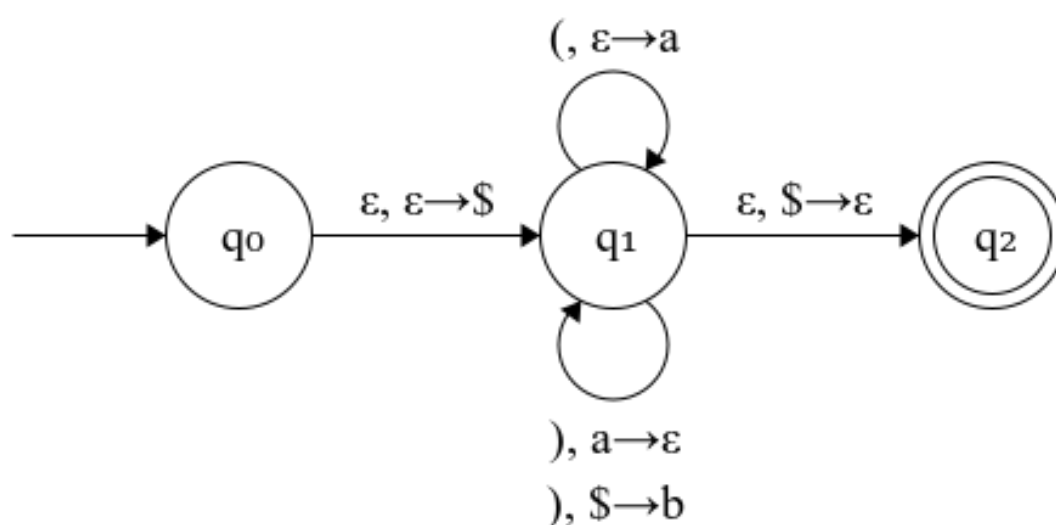
Automata Table :

Input	()				ϵ			
Stack	a	b	\$	ϵ	a	b	\$	ϵ	a	b	\$	ϵ
q ₀												{{q ₁ , \$}}
q ₁				{{q ₁ , a}}	{{q ₁ , ϵ }}		{{q ₁ , b}}				{{q ₂ , ϵ }}	
q ₂												

✚ Empty Cells represent ϕ

✚ \$ Symbolises to the Bottom of Stack

Automata Diagram :



Program :

```

#include <iostream>
#include <stack>
#include <string>

bool is_wellformed(std::string input) {
    std::stack<char> pda_stack;
    char state = '0';

    // State q_0
    state = '1';
    pda_stack.push('$');

    for (auto &i : input) {
        switch (state) {

            // State q_1
            case '1':
                if (i == '(') {
                    state = '1';
                    pda_stack.push('a');
                } else if (i == ')') && pda_stack.top() == 'a' {
                    state = '1';
                    pda_stack.pop();
                } else if (i == ')') && pda_stack.top() == '$' {
                    state = '1';
                    pda_stack.push('b');
                }
                break;
            }

        // State q_2
        if (pda_stack.top() == '$')
            return true;
        else
            return false;
    }

    int main(int argc, char *argv[]) {
        std::string input;

        std::cout << "Enter String : ";
        std::cin >> input;

        std::cout << is_wellformed(input);

        return 0;
    }

```

Madhav Institute of Technology & Science, Gwalior

(A Govt. Aided UGC Autonomous NAAC Accredited Institute Affiliated to RGPV Bhopal)

Program : 11

Aim :

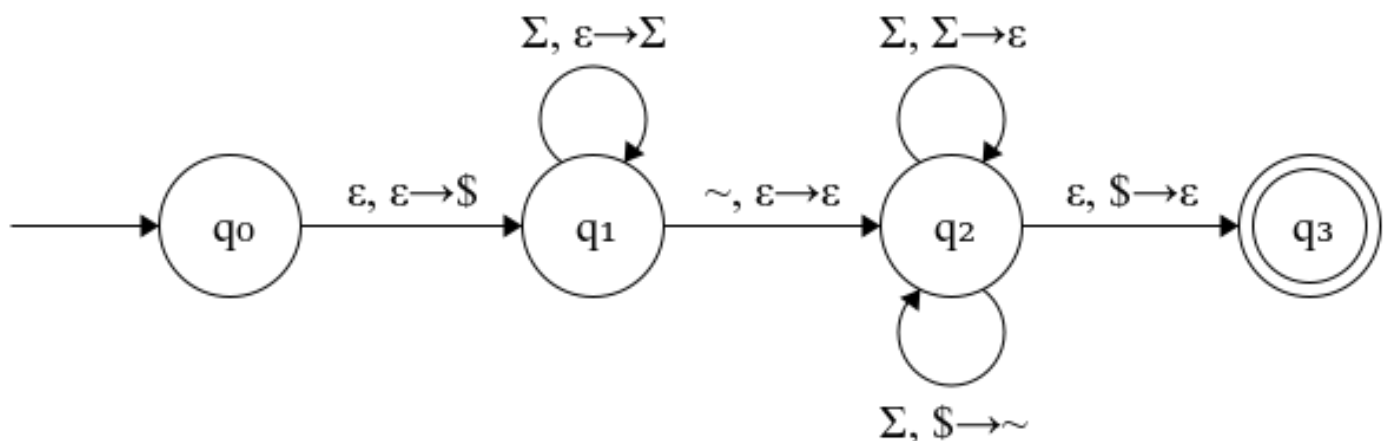
Design a program to create PDA to accept WCWR where w is any string, WR is reverse of that string, and C is a Special symbol.

Automata Table :

Input	Σ				\sim				ϵ			
Stack	Σ	\sim	$\$$	ϵ	Σ	\sim	$\$$	ϵ	Σ	\sim	$\$$	ϵ
q_0												$\{(q_1, \$)\}$
q_1				$\{(q_1, \Sigma)\}$				$\{(q_3, \epsilon)\}$				
q_2	$\{(q_2, \epsilon)\}$		$\{(q_2, \sim)\}$								$\{(q_3, \epsilon)\}$	
q_3												

- Empty Cells represent ϕ
- Special Character (C) is \sim
- Σ Refers to any Input Symbol except Special character
- $\$$ Symbolises to the Bottom of Stack

Automata Diagram :



Program :

```
#include <iostream>
#include <stack>
#include <string>

#define SPECIAL_SYMBOL '~'

bool is_wellformed(std::string input) {
    std::stack<char> pda_stack;
    char state = '1';

    // State q_1
    state = '2';
    pda_stack.push('$');

    for (auto &i : input) {
        switch (state) {

            // State q_2
            case '2':
                if (i == SPECIAL_SYMBOL) {
                    state = '3';
                } else {
                    state = '2';
                    pda_stack.push(i);
                }
                break;

            // State q_3
            case '3':
                if (pda_stack.top() == i) {
                    state = '3';
                    pda_stack.pop();
                } else if (pda_stack.top() == '$') {
                    state = '3';
                    pda_stack.push(SPECIAL_SYMBOL);
                }
                break;

        }
    }

    // State q_4
    if (pda_stack.top() == '$')
        return true;
    else
        return false;
}

int main(int argc, char *argv[]) {
    std::string input;

    std::cout << "Enter String : ";
    std::cin >> input;

    std::cout << is_wellformed(input);

    return 0;
}
```

Madhav Institute of Technology & Science, Gwalior

(A Govt. Aided UGC Autonomous NAAC Accredited Institute Affiliated to RGPV Bhopal)

Program : 12

Aim :

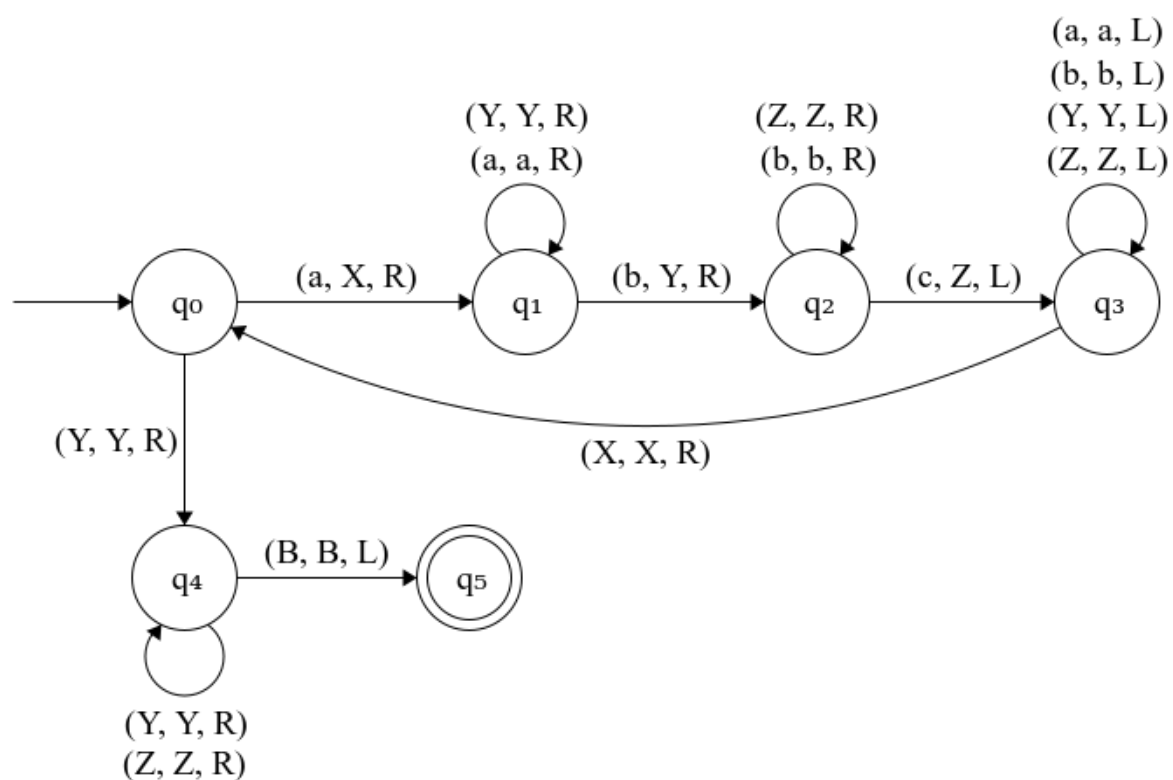
Design a Turing machine that accepts the following language $a^n b^n c^n$ where $n > 0$.

Automata Table :

	Input						
	a	X	b	Y	c	Z	B
States	q_0	(X, R)		(Y, R)			
	q_1	(a, R)	(Y, R)	(Y, R)			
	q_2		(b, R)		(Z, L)	(Z, R)	
	q_3	(a, L)	(X, R)	(b, L)	(Y, L)	(Z, L)	
	q_4			(Y, R)		(Z, R)	(B, L)
	q_5						

Empty Cells represent ϕ

Automata Diagram :



Program :

```
#include <iostream>
#include <stack>
#include <string>

#define SPECIAL_SYMBOL '~'

bool is_wellformed(std::string input) {
    std::stack<char> pda_stack;
    int flag = 0;

    for (auto &i : input) {
        switch (flag) {
            case 0:
                if (i == SPECIAL_SYMBOL) {
                    flag = 1;
                    break;
                }

                pda_stack.push(i);
                break;

            case 1:
                if (pda_stack.top() == i)
                    pda_stack.pop();
                break;
        }
    }

    return pda_stack.empty();
}

int main(int argc, char *argv[]) {
    std::string input;

    std::cout << "Enter String : ";
    std::cin >> input;

    std::cout << is_wellformed(input);

    return 0;
}
```