

OUTPUT:

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ detect_and_print_all_validthoken_inC.cpp -o detect_and_print_
all_validthoken_inC } ; if ($?) { .\detect_and_print_all_validthoken_inC }
Enter a C program statement: int main() { return 0; }
Token: int
Token: main
Token: return
Token: 0
PS D:\TOC> █
```

OUTPUT:

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ dfa_accepting_valid_c_identifiers.cpp -o dfa_accepting_valid_c_identifiers } ; if ($?) { .\dfa_accepting_valid_c_identifiers }
Enter an identifier: my_variable123
Valid C identifier
PS D:\TOC> █
```

OUTPUT:

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ dfa_ending_11.cpp -o dfa_ending_11 } ; if ($?) { .\dfa_ending_11 }
Enter any string of 0 and 1 :
1001111
Move(0,1)=1
Move(1,0)=0
Move(0,0)=0
Move(0,1)=1
Move(1,1)=2
Move(2,1)=2
Move(2,1)=2
STRING ACCEPTED
Enter any string of 0 and 1 :
1000
Move(2,1)=2
Move(2,0)=0
Move(0,0)=0
Move(0,0)=0
STRING NOT ACCEPTED
```

OUTPUT:

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ dfa_starting_aa.cpp -o dfa_starting_aa } ; if ($?) { .\dfa_starting_aa }
Enter any string of 'a' and 'b':
aabba
Move(0,a)=1
Move(1,a)=2
Move(2,b)=2
Move(2,b)=2
Move(2,a)=2
STRING ACCEPTED
Enter any string of 'a' and 'b':
ababab
Move(0,a)=1
Move(1,b)=3
Move(3,a)=3
Move(3,b)=3
Move(3,a)=3
Move(3,b)=3
STRING NOT ACCEPTED
```

OUTPUT:

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ dfa_with_aba_as_substring.cpp -o dfa_with_aba_as_substring }  
; if ($?) { .\dfa_with_aba_as_substring }
```

Enter any string of 'a' and 'b': abababa

String Accepted

Enter any string of 'a' and 'b': aaaaaa

String not Accepted

OUTPUT:

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ dfa_with_two_consecutive_1.cpp -o dfa_with_two_consecutive_1
} ; if ($?) { .\dfa_with_two_consecutive_1 }
Enter any string of 0 and 1 :
011001
Move(0,0)=0
Move(0,1)=1
Move(1,1)=2
Move(2,0)=2
Move(2,0)=2
Move(2,1)=2
STRING ACCEPTED
PS D:\TOC> 
```

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ dfa_with_two_consecutive_1.cpp -o dfa_with_two_consecutive_1
} ; if ($?) { .\dfa_with_two_consecutive_1 }
Enter any string of 0 and 1 :
010101001
Move(0,0)=0
Move(0,1)=1
Move(1,0)=0
Move(0,1)=1
Move(1,0)=0
Move(0,1)=1
Move(1,0)=0
Move(0,0)=0
Move(0,1)=1
STRING NOT ACCEPTED
PS D:\TOC> 
```

OUTPUT:

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ nfa_accepting_bab_as_substring.cpp -o nfa_accepting_bab_as_s
ubstring } ; if ($?) { .\nfa_accepting_bab_as_substring }
Enter any string:
aababaaa
STRING ACCEPTED
PS D:\TOC> 
```

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ nfa_accepting_bab_as_substring.cpp -o nfa_accepting_bab_as_s
ubstring } ; if ($?) { .\nfa_accepting_bab_as_substring }
Enter any string:
bbbbbbb
STRING NOT ACCEPTED
PS D:\TOC> 
```

OUTPUT:

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ nfa_ending_with_00.cpp -o nfa_ending_with_00 } ; if ($?) { .
\nfa_ending_with_00 }
Enter any string of 0s and 1s:
1100
STRING ACCEPTED
PS D:\TOC> 
```

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ nfa_ending_with_00.cpp -o nfa_ending_with_00 } ; if ($?) { .
\nfa_ending_with_00 }
Enter any string of 0s and 1s:
0011
STRING NOT ACCEPTED
PS D:\TOC> 
```


OUTPUT:

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ nfa_starting_with_11.cpp -o nfa_starting_with_11 } ; if ($?)  
{ .\nfa_starting_with_11 }
```

Enter any string of 0s and 1s:

1100

STRING ACCEPTED

PS D:\TOC> █

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ nfa_starting_with_11.cpp -o nfa_starting_with_11 } ; if ($?)  
{ .\nfa_starting_with_11 }
```

Enter any string of 0s and 1s:

0011

STRING NOT ACCEPTED

PS D:\TOC> █

OUTPUT:

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ PDA_accepting_the_language.cpp -o PDA_accepting_the_language } ; if ($?) { .\PDA_accepting_the_language }  
Enter a string: aaabbb  
String accepted.  
PS D:\TOC> █
```

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ PDA_accepting_the_language.cpp -o PDA_accepting_the_language } ; if ($?) { .\PDA_accepting_the_language }  
Enter a string: aabbb  
String not accepted.  
PS D:\TOC> █
```

OUTPUT:

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ regular_expression_containing_aba_as_substring.cpp -o regular_expression_containing_aba_as_substring } ; if ($?) { .\regular_expression_containing_aba_as_substring }  
Enter a string: aaabaaa  
String accepted  
PS D:\TOC> 
```

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ regular_expression_containing_aba_as_substring.cpp -o regular_expression_containing_aba_as_substring } ; if ($?) { .\regular_expression_containing_aba_as_substring }  
Enter a string: bbbbbb  
String not accepted  
PS D:\TOC> 
```

OUTPUT:

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ regexexpression_identifying_phoneno.cpp -o regexexpressi
on_identifying_phoneno } ; if ($?) { .\regexexpression_identifying_phoneno }
Enter a phone number: 9823666666
Valid phone number detected!
PS D:\TOC> █
```

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ regexexpression_identifying_phoneno.cpp -o regexexpressi
on_identifying_phoneno } ; if ($?) { .\regexexpression_identifying_phoneno }
Enter a phone number: 99334455
Not a valid phone number!
PS D:\TOC> █
```

```
PS D:\TOC> cd "d:\TOC\" ; if ($?) { g++ regexexpression_identifying_phoneno.cpp -o regexexpressi
on_identifying_phoneno } ; if ($?) { .\regexexpression_identifying_phoneno }
Enter a phone number: 9735362512
Valid phone number detected!
PS D:\TOC> █
```