In [22]:

```python
# sample example on Inner loop
a = 1
for r in range(1,6):
    for c in range(1,8):
        if a<=31:
            if c == 3:
                print("{}".format("||"),end=" ")
            elif r == 4:
                print("{}".format("[]"),end=" ")
            elif a%3==0:
                print("{}".format("##"),end=" ")
            elif a%5==0:
                print("{}".format("**"),end=" ")
            elif a%2==0:
                print("{}".format("&&"),end=" ")
            elif a>15:
                print("{}".format("()"),end=" ")
            elif a<10 and a>0:
                print("{}".format("$$"),end=" ")
            else:
                print("{}".format("@@"),end=" ")
        else:
            break
        a+=1
    print(end="\n")
```

```
$$ && || && ** ## $$
&& ## || @@ ## @@ &&
## && || ## () ** ##
[] [] || [] [] [] []
() ## ||
```

Functions:

```
    -> To perform a Specific task
    -> Reuseability of code
    -> Logic is to be implemented in function
    -> Errors are identified easily and can be rectified soon by a user
    -> Two Types of Functions they are
        - Pre Defined and
        - User Defined
    -> Pre Defined Functions:
        - Already Work has to be fixed to a particular Builtin functions
        - Example: print(),input(),max(),min(),range(),pow(),len(),str()
    -> User Defined Functions:
        - User has to assign a task for a function to execute
        - Example: sumadd(),sub(),largst(),etc.,
    -> Function Syntax:
        - Syntax:
                    def func_name(arguments):
                        //stmnts
                        return ?
    -> Four different Types in User Defined
        - With return_type and with arguments
        - With return_type and without arguments
        - Without return_type and with arguments
        - Without return_type and without arguments
```

In [ ]:

```
for i in range(1,6):
    print(i)
```