**FIFO / First Come First Serve**

**when to use 🡪**

**when u need things to happen in the same order those things are called, but your computer/printer etc is slow, so you give the functions in a queue and will be executed in that order, new function in end.**

**Process transactions or anything**

**JUST TO PRESERVE THE ORDER IN WHICH U SAID**

#include<iostream>

#include<queue>

using namespace std;

void printq(queue<int>& q) {

while (!q.empty()) {

cout << q.front() << " ";

q.pop();

}

}

int main() {

queue<int> myq;

myq.push(0);

myq.push(1);

myq.push(2);

myq.push(10);

myq.pop(); // FIRST IN FIRST OUT,,,ZERO WENT FIRST SO IT CAME OUT FIRST

cout << "size of the queue is: " << myq.size() << endl;

cout << "first element of the queue is: " << myq.front() << endl;

// remember, for stack, it was .top(), that gave the last element added as the first output

cout << "last element of the queue is: " << myq.back() << endl;

cout << "\nfull queue: " << endl;

printq(myq);

return 0;

}