```
/**
 * 情報Ⅰ　拡張ブロック
 */
enum ButtonNames {
    //% block="A"
    A,
    //% block="B"
    B,
    //% block="A+B"
    AB,
    //% block="A か B"
    AorB,
    //% block="A か A+B"
    AorAB,
    //% block="B か A+B"
    BorAB,
    //% block="A か B か A+B"
    AorBorAB,
    //% block="いずれか"
    ANY
}
//% weight=100 color=#0fbc11 icon="\uf0c3"
namespace Joho1ext {
    /**
        * @param b 入力要求ボタン eg："A"
        * 選択されたボタンが押されるまで待つブロック
        * A ボタンが押されたら戻り値：1
        * B ボタンが押されたら戻り値：2
        * AB ボタンが押されたら戻り値：3
    */
    //% block="%b=ButtonNames|のボタンが押されるまで待つ"
    export function 押されるまで待つ(b: ButtonNames): number {
        if (b == ButtonNames.A) {
            return A のボタンが押されるまで待つ()
        } else if (b == ButtonNames.B) {
            return B のボタンが押されるまで待つ()
        } else if (b == ButtonNames.AB) {
            return AB のボタンが押されるまで待つ()
        } else if (b == ButtonNames.AorB) {
            return A か B のボタンが押されるまで待つ()
        } else if (b == ButtonNames.AorAB) {
            return A か AB のボタンが押されるまで待つ()
        } else if (b == ButtonNames.BorAB) {
            return B か AB のボタンが押されるまで待つ()
        } else if (b == ButtonNames.AorBorAB || b == ButtonNames.ANY) {
            return A か B か AB のボタンが押されるまで待つ()
        } else {
            return -1
        }
    }
```

```
function A のボタンが押されるまで待つ(): number {
    while (!(input.buttonIsPressed(Button.A))) {
        basic.pause(10)
    }
    if (input.buttonIsPressed(Button.A)) {
        while(input.buttonIsPressed(Button.A)){

        }
        return 1
    } else {
        return -1
    }
}
function B のボタンが押されるまで待つ(): number {
    while (!(input.buttonIsPressed(Button.B))) {
        basic.pause(10)
    }
    if (input.buttonIsPressed(Button.B)) {
        while (input.buttonIsPressed(Button.B)) {
        }
        return 2
    } else {
        return -1
    }
}
function AB のボタンが押されるまで待つ(): number {
    while (!(input.buttonIsPressed(Button.AB))) {
        basic.pause(10)
    }
    if (input.buttonIsPressed(Button.AB)) {
        while (input.buttonIsPressed(Button.AB)) {
        }
        return 3
    } else {
        return -1
    }
}
```

```
    /**
         * A または A+B ボタンが押されるまで待つブロック
         * A ボタンが押されたら戻り値：1
         * AB ボタンが押されたら戻り値：3
    */
    //% block="A:1 か AB:3 のボタンが押されるまで待つ"
    function A か AB のボタンが押されるまで待つ(): number {
        while (!(input.buttonIsPressed(Button.A)) && !(input.buttonIsPressed(Button.B))
&& !(input.buttonIsPressed(Button.AB))) {
            basic.pause(10)
        }
        if (input.buttonIsPressed(Button.AB)) {
            while (input.buttonIsPressed(Button.AB)) {
            }
            return 3
        } else if (input.buttonIsPressed(Button.A)) {
            while (input.buttonIsPressed(Button.A)) {
            }
            return 1
        } else {
            return -1
        }
    }
    /**
         * B または AB ボタンが押されるまで待つブロック
         * B ボタンが押されたら戻り値：2
         * AB ボタンが押されたら戻り値：3
    */
    //% block="B:2 か AB:3 のボタンが押されるまで待つ"
    function B か AB のボタンが押されるまで待つ(): number {
        while (!(input.buttonIsPressed(Button.A)) && !(input.buttonIsPressed(Button.B))
&& !(input.buttonIsPressed(Button.AB))) {
            basic.pause(10)
        }
        if (input.buttonIsPressed(Button.AB)) {
            while (input.buttonIsPressed(Button.AB)) {
            }
            return 3
        } else if (input.buttonIsPressed(Button.B)) {
            while (input.buttonIsPressed(Button.B)) {
            }
            return 2
        } else {
            return -1
        }
    }
```

```
    /**
          * A または B または AB ボタンが押されるまで待つブロック
          * A ボタンが押されたら戻り値：1
          * B ボタンが押されたら戻り値：2
          * AB ボタンが押されたら戻り値：3
    */
    //% block="A:1 か B:2 か AB:3 のボタンが押されるまで待つ"
    function A か B か AB のボタンが押されるまで待つ(): number {
        while (!(input.buttonIsPressed(Button.A)) && !(input.buttonIsPressed(Button.B))
&& !(input.buttonIsPressed(Button.AB))) {
            basic.pause(10)
        }
        if (input.buttonIsPressed(Button.AB)) {
            while (input.buttonIsPressed(Button.AB)) {
            }
            return 3
        } else if (input.buttonIsPressed(Button.A)) {
            while (input.buttonIsPressed(Button.A)) {
            }
            return 1
        } else if (input.buttonIsPressed(Button.B)) {
            while (input.buttonIsPressed(Button.B)) {
            }
            return 2
        } else {
            return -1
        }
    }
    /**
        * A または B ボタンが押されるまで待つブロック
        * A ボタンが押されたら戻り値：1
        * B ボタンが押されたら戻り値：2
    */
    //% block="A:1 か B:2 のボタンが押されるまで待つ"
    function A か B のボタンが押されるまで待つ(): number {
        while (!(input.buttonIsPressed(Button.A)) && !(input.buttonIsPressed(Button.B))) {
            basic.pause(10)
        }
        if (input.buttonIsPressed(Button.A)) {
            while (input.buttonIsPressed(Button.A)) {
            }
            return 1
        } else if (input.buttonIsPressed(Button.B)) {
            while (input.buttonIsPressed(Button.B)) {
            }
            return 2
        } else {
            return -1
        }
    }
```

```
    /**
     * 入力を待つブロック
     * @param s 入力要求文字列 eg："abc"
     */
    //% block="〇〇と聞いて待つ %s"
    export function 〇〇と聞いて待つ(s: string): number {
        basic.showString(s)
        let 答え = 0
        basic.showNumber(答え)
        while (!(input.buttonIsPressed(Button.B))) {
            basic.pause(10)
            if (input.buttonIsPressed(Button.A)) {
                答え = (答え + 1) % 10
                basic.pause(200)
                basic.showNumber(答え)
            }
        }
        basic.clearScreen()
        return 答え
    }
}
```