

1 *if* нөхцөл шалгах үйлдэл

Программчлалд хувьсагчийн тодорхой утгаас хамааруулан өөр үйлдлүүд хийх асуудал гардаг. Үүнийг Си хэлэнд шийдэхдээ *if* түлхүүр үгийг хэрэглэдэг.

Дүрэм:

```
// 1. if-ээс өмнөх үйлдлүүд
if (нөхцөл) {
    // 2. үнэн үед хийгдэх үйлдлүүд
} else {
    // 3. худал үед хийгдэх үйлдлүүд
}
// 4. Ямар ч үед хийгдэх үйлдлүүд
```

Энд *<нөхцөл>* нь бүүлийн нөхцөл.

1. 0 бол худлаа байна.
2. 0-ээс ялгаатай бүх утгууд дээр үнэн байна.

Үнэн, худлын бүүлийн нөхцөлийг логик операторууд ашиглан бичнэ. Логик операторууд:

`==` Тэнцүү

`!=` Ялгаатай

`>` Эрс их

`<` Эрс бага

`<=` Их буюу тэнцүү

`>=` Бага буюу тэнцүү

Логик операторууд нь бүгд хоёр операндыг авч, логик үйлдлийг хийн утгыг буцаадаг. Хэрэв буцаасан утга 0 бол худал 1 бол үнэн байдаг. Жишээ нь:

```
#include <stdio.h>
int main() {
    int a, b;
    scanf("%d%d", &a, &b);
    printf("%d\n", a == b); // herev tentsuu бол 1 ugui
                           бол 0 iig hevlene
```

```

    if (a == b)          // a == b бол if (1)
                        // a != b бол if (0)
        printf("tentsuu");
    else
        printf("yalgaatai");
    return 0;
}

```

Си хэлэнд `{}` нь хэд хэдэн командуудыг багцлахад хэрэглэгддэг. Дээрх жишээнд үнэн үед ганц л үйлдэл хийж байгаа учраас `{}` хаалтыг хэрэглэхгүй байж болно.

Хэрэв хэд хэдэн нөхцөлийг зэрэг шалгах хэрэгтэй бол логик `and (&&)`, логик `or (||)` хэрэглэгдэнэ. Нөхцөлийг үгүйсгэх бол `!` операторыг хэрэглэнэ.

`<a> && ` хоёр нөхцөл хоёул үнэн бол нийтдээ үнэн байдаг. Хэрэв `<a>` нөхцөл худал бол `` нөхцөлийг ерөөсөө шалгахгүй.

`<a> || ` хоёр нөхцөлийн аль нэг нь л үнэн байхад үнэн байна. Хэрэв `<a>` нөхцөл үнэн бол `` нөхцөлийг шалгахгүй.

Жишээ нь:

```

    if (a <= b && b <= c)          // зов
        printf("a hamgiin бага\n");

    if (a < 0 || a > 0)             // зов
        printf("a ni 0-ees yalgaatai\n");

    if (a <= b <= c) // алдаа
        printf("%d\n", b);

    if (1) printf("unen\n");
    if (!0) printf("unen\n");

```

Мөн хэд хэдэн утгын аль нэгтэй нь тэнцэж байгаа эсэхийг шалгахыг хүсвэл шаталсан `if` бичиж болно. Шаталсан `if` бичвэл тухайн тохиолдолд ганц үнэн үйлдэл нь хийгдэх ба ямар ч үнэн тохиолдол байхгүй бол хамгийн доод талын `else` үйлдэл хийгдэнэ.

```

    if (a == 0)
        printf("teg\n");
    else if (a == 1)
        printf("neg\n");
    else if (a == 2)
        printf("2\n");
    else
        printf("2-oos ih\n");

```

Мэдээж `if` дотор ахиад `if` нөхцөл шалгах үйлдэл хийж болно.

Жишээ нь:

```

if (a > b) {
    if (a > c) {
        printf("a hamgiin ih");
    } else
        printf("c hamgiin ih");
}

```

2 Дасгалууд:

2.1 Ангид

1. n он өндөр жил эсэхийг шалга. Доорх хоёр тохиолдолд тухайн жил өндөр жил байна.
 - (a) 400-д хуваагддаг.
 - (b) 4-д хуваагддаг, 100-д хуваагддаггүй.

2. Алдааг ол.

```

if (a == b);
    printf("tentsuu\n");
else
    printf("yalgaatai\n");

```

3. Өгсөн таван бүхэл тооноос хамгийн их, хамгийн бага мөн дундах тоог нь ол.
4. (x_1, y_1) , (x_2, y_2) , (x_3, y_3) гэсэн 3 цэг өгөгдөхөд координатын эхээс хамгийн ойр, хол орших 2 цэгийг ол.

2.2 Гэрт

1. Өгөгдсөн тоо тэгш, эсвэл сондгой болохыг дэлгэцлэж тэгш байвал 10, 4-д эсвэл 2, 8-д зэрэг хуваагдах эсэхийг сондгой байвал 3, 7-д эсвэл 5, 9-д зэрэг хуваагдах эсэхийг шалга.
2. Он, сар, өдөр өгөгдөхөд ирээдүй, эсвэл өнгөрсөн эсэхийг шалга.
3. Өгсөн 6 хүртэлх оронтой бүхэл тоонд ямар цифр хамгийн олон орсныг ол.