

## 1 switch

Энэхүү командаар нөхцөлийг тоочиж шалгахад ашиглана. Ямар нэг илэрхийллийн утгаас хамаарч доорх бүлэг үйлдлүүдээс сонгож хийнэ.

Сонголтын үйлдэл нь доорх хэлбэртэйгээр бичигдэнэ.

```
switch(Илэрхийлэл эсвэл хувьсагч)
{
    case утга1:
        үйлдлүүд1;
        break;
    case утга2:
        үйлдлүүд2;
        break;
    case утга3:
        үйлдлүүд3;
        break;
    default:
        бусад_үйлдлүүд;
}
```

Дээрх *switch*-н хаалт дотор байгаа илэрхийллийн утга багц командын хаалтан доторх *case*-н ямар нэг утгатай тэнцүү байвал програмын удирдлага тухайн *case* доторх үйлдэлрүү шилжинэ. Тухайн *case* доторх бүлэг үйлдлийг хийж дуусах үед *switch* командыг төгсгөхийн тулд *break* командыг ашиглана. Хэрэв *break*-г бичээгүй тохиолдолд дараагийн *case* доторх үйлдлийг хийх бөгөөд *break* команд тулгарах эсвэл *switch*-ийн төгсгөл хүртэлх бүх үйлдлийг хийнэ.

*switch*-н хаалт дотор байгаа илэрхийллийн утгатай тэнцүү *case*-ийн утга олдохгүй бол *default*-ийн ард байгаа бүлэг үйлдлийг хийнэ. Тэнцүү утга олдоогүй, *default*-ийг тодорхойлоогүй үед ямар ч үйлдэл хийгдэхгүй.

Жишээ:

```
#include <stdio.h>
int main() {
    int n;
    scanf("%d", &n);
    switch(n) {
        case 1:
            printf("neg\n");
            break;
        case 2:
            printf("hojr\n");
            break;
        case 3:
            printf("gurav\n");
        default:
            printf("<0 or >3\n");
    }
    return 0;
}
```

Энд хэрэв  $n$ -ийн утга 1, 2, 3-ийн аль нь ч биш бол 0-ээс бага, 3-саа их гэсэн хариу гаргаж байна.

## 2 Давхар давталт

Давталт дотор ахин давталт хэрэглэх боломжтой. Тэгвэл хамгийн дотор талын давталт доторх үйлдлүүд гаднах давталтуудын тоогоор үржигдэн хийгдэнэ. Жишээ нь доорх давталт 1000 удаа "x" үсгийг хэвлэх юм.

```
int i, j, k;
for (i = 0; i < 10; i++) {
    for (j = 0; j < 10; j++) {
        for (k = 0; k < 10; k++) {
            printf("x");
        }
    }
}
```

Дээрх кодын гадна талын  $i$  давталтын доторх үйлдлүүд 10 удаа,  $j$  давталт  $i$  давталт дотор байгаа учраас 10-н удаа шинээр эхлэх тул  $j$  давталт доторх үйлдлүүд нийтдээ 100 удаа хийгдэнэ. Харин  $k$  давталт  $j$  давталт дотор байгаа тул 100 удаа шинээр эхлэх бөгөөд  $k$  давталтын доторх үйлдлүүд 1000 удаа хийгдэнэ гэсэн үг юм.

Одоо доорх тоонуудыг хэвлэх жишээг авч үзье.

Оролт:

2 3

Гаралт:

0 0  
0 1  
0 2  
1 0  
1 1  
1 2

Хэрэв  $n$ -д 2,  $m$ -д 3 гэсэн утга оноож өгсөн бол дараах код дээрх тоонуудыг хэвлэх юм.

```
int i, j, n, m;
scanf("%d%d", &n, &m);
for (i = 0; i < n; i++) {
    for (j = 0; j < m; j++) {
        printf("%d %d", i, j);
    }
}
```

## 3 Дасгалууд

### 3.1 Ангид

1. switch-г ашиглан сар оруулахад хэдэн хоногтойг нь хэвлэн гаргадаг програм зохио.

2. Дараах кодыг  $n$  нь тэгш үед "тэгш", сондгой үед "сондгой" гэж хэвлэх код болгон зас.

```
#include <stdio.h>
int main() {
    int n;
    scanf("%d", &n);
    switch(n % 2) {
        case 1:
            printf("songdgoi\n");
        case 0:
            printf("tegsh\n");
    }
    return 0;
}
```

3. 1-с 10 хүртэлх үржихийн хүрдийг хэвлэ.
4.  $n$  тоо өгөгдөхөд  $n$  мөр  $x$ -г дараах байдалтай хэвлэн харуул.

Оролт:

4

Гаралт:

x

xx

xxx

xxxx

5.  $n$  тоо өгөгдөхөд  $n$  х  $n$  хэмжээтэй доорх хөлгийг хэвлэ.

Оролт:

4

Гаралт:

хохо

охох

хохо

охох

6.  $n$  тоо өгөгдөхөд  $2 * n - 1$  мөр  $x$ -г доорх байдлаар хэвлэн харуул.

Оролт:

4

Гаралт:

x

xx

xxx

xxxx

xxx

xx

x

## Илүү дасгал

1.  $n$  тоо өгөгдөхөд тухайн тоо анхны тоо эсэхийг шалга. Анхны тоо гэдэг нь зөвхөн 1, эсвэл өөртөө хуваагдах тоог хэлнэ. Жишээ нь: 2, 5, 7, 11, 13 нь анхны тоонууд юм.
2.  $n$  тоо өгөгдөхөд  $x^2 - y^2$  нь  $n$ -д хуваагддаг байх бүх  $0 \leq x, y < n$  тоог хэвлэ.

## 3.2 Гэрт

1.  $n$  тоо өгөгдөхөд  $1 \dots n$  завсарт орших бүх анхны тоонуудыг ол.

Оролт:

10

Гаралт:

2 3 5 7

2.  $x^2 + y^2 = z^2$  ба  $1 \leq x \leq y \leq z \leq 100$  байх бүх  $x, y, z$  тоонуудыг хэвлэ.

Гаралт:

3 4 5

...

3.  $n$  тоо өгөгдөхөд  $n$  мөр  $x$ -г доорх байдлаар хэвлэн харуул.

Оролт:

4

Гаралт:

x

xxx

xxxxx

xxxxxxx