

# 1 Хэрэглэгчийн тодорхойлсон функцүүд

Тодорхой нэгэн даалгаврыг гүйцэтгэх нэг нэгж болж эмхтгэгдсэн програмын тушаалуудын дарааллыг **функц** гэнэ. Энэ нэг нэгж нь тухайн даалгавар гүйцэтгэгдэх болгонд програмд хэрэглэгдэнэ.

## 1.1 Функц зарлах

Ямар нэгэн функцийг хэрэглэхэд тухайн функц заавал өмнө нь зарлагдсан байх ёстой. Харин тухайн функцийн тодорхойлолтыг програмын хаана ч хийж болох юм.

Дүрэм:

буцаах\_төрөл нэр(төрөл параметр1, төрөл параметр2, ...);

Энд

**буцаах\_төрөл** нь тухайн функц хэрвээ ямар нэгэн утга *return* түлхүүр үгийг ашиглан буцааж байвал буцааж байгаа утгын төрөл юм. Хэрвээ ямар ч утга буцаадаггүй бол төрөл нь *void* байна. *int main()* функц бүхэл тоон утга буцаадаг учраас *буцаах\_төрөл* нь *int* байна.

**нэр** нь тухайн нэгжийг нэрлэх нэр бөгөөд функцийг дуудан програмд хэрэглэхэд хэрэглэгдэнэ.

**параметрууд** нь тухайн функцийг хэрэглэхийн тулд түүнд утга дамжуулах шаардлагатай бол энд тодорхойлно. Энд зарлагдсан утгууд зөвхөн тухайн функцийн хүрээнд л ашиглагдах ба функц дуусахад устана. Функцийг зарлахад параметруудийн зөвхөн төрлийг нь тавьж, нэрийг нь тавихгүй байж болдог. Харин функцээ тодорхойлохдоо заавал нэр төрлийг нь тавих шаардлагатай. Жишээ нь:

```
int max(int, int); // hoyr toonii ihig butsaah
```

## 1.2 Функц тодорхойлох

Функцийг өмнө нь зарласан бол түүнийг хаана ч тодорхойлж болно. Функцийг тодорхойлоход функцийн буцаах утга, нэр, параметруудийн тоо ширхэг, төрөл нь өмнө нь зарласантайгаа яг таарч байх ёстой. Харин энэ удаа параметруудийн нэрийг заавал тавьж тэдгээрийг зарлах ёстой. Параметруудийг зарласны дараа дугуй хаалтаа хаан функцийн гүйцэтгэх үйлдлүүдийг багц командын хаалтан дотор бичнэ. Хэрэв тодорхойлж буй функц утга буцаадаг функц бол заавал буцаах төрлийнхөө утгыг буцаах ёстой. Жишээ нь *int* төрлийн утга буцаадаг бол бүхэл тоо, *int \** төрлийн утга буцаах бол бүхэл тоон төрлийн хаяг буцаана гэсэн үг юм.

Дүрэм:

```
буцаах_төрөл нэр(төрөл параметр1, төрөл параметр2, ...) {  
    // функцийн гүйцэтгэх  
    // үйлдлүүд  
}
```

Функц дотор зарлагдсан бүхий л хувьсагчид тухайн функц хэрэглэгдэх үед үүсэж, дуусах үед бүгд устдаг. Тиймээс функцийн дотоод хувьсагчид тухайн функцийн хүрээндээ л хэрэглэгдэнэ. Функц аль болохоор функцийн гадна хувьсагчуудаас хамааралгүй байх ёстой. Энэ нь хамааралтай утгуудаа параметрээрээ авч ашиглавал, функц ямар нэгэн гадаад нөхцөлөөс хамааралгүй болно гэсэн үг юм.

### 1.3 Функцийг хэрэглэх

Функцийг хэрэглэхдээ тухайн функцийг нэрээр нь дуудан хэрэглэнэ. Хэрэв тухайн функц параметртэй функц бол параметруудад нь тохирсон утгыг дарааллаар нь бичиж өгнө.

Хэрэв дуудаж буй функц утга буцаадаг функц бол буцаасан утгыг утга оноох оператораар хувьсагчид хадгална.

Дүрэм:

хувьсагч = функцийн\_нэр(утга1, утга2, ...);

Энд

**хувьсагч** тухайн функцийн буцаасан утгыг хадгалах хувьсагч.

**функцийн\_нэр** дуудаж байгаа функцийн нэр.

**утга1, утга2, ...** тухайн функцийн параметруудад дамжуулж байгаа утгууд. Утгуудынх нь тоо, төрөл нь дуудаж байгаа функцийнхээ параметруудийн тоо, төрөлтэй нь тэнцэж байх ёстой.

Жишээ 1: Хоёр тооны ихийг олдог функц.

```
#include <stdio.h>
int max(int, int);
int main() {
    int x, y;
    scanf("%d%d", &x, &y);
    printf("%d\n", max(x, y)); // funktsiin damjuulsan utgiig
    // shuud hevlene
    return 0;
}

// doorh a, b ni main() доторh a, b-ees yalgaatai
int max(int a, int b) {
    if (a > b)
        return a;

    return b;
}
```

Жишээ 2: Функцрүү дамжуулсан тоо анхны тоо мөн бол нэгийг, үгүй бол тэгийг буцаадаг функц.

```
#include <stdio.h>
int prime(int);
```

```

int main() {
    int k;
    scanf("%d", &k);
    // k-iin utgiig prime-ruu damjuulj, herev
    // 1-iig butsaaval k ni anhnii too
    // 0-iig butsaaval k ni anhnii too bish
    if (prime(k) == 1)
        printf("%d ahnii too\n", k);
    else
        printf("%d ahnii too bish\n", k);

    return 0;
}

int prime(int n) {
    int i;
    for (i = 2; i < n; i++) {
        if (n % i == 0) // huvaagdval shuud 0-iig butsaa
            return 0; // tsaash ni urgeljluulj shalgah hereg bhgui
    }

    // end hurvel n ni anhnii too mon
    // bish baisan bol deerh uyd ali hediin tegiig butsaasan bna.
    return 1;
}

```

## 1.4 Дасгалууд

### 1.4.1 Ангид

1. Тэгш бол 1, сондгой бол 0-г буцаах доорх функцийг тодорхойл.

```
int is_even(int, int);
```

Дээрх бичсэн функцээ ашиглан  $n$  тоог гараас авч,  $n$  хүртэлх бүх сондгой тоонуудыг хэвлэн харуул.

2. Дараах функц нь хүснэгтэд  $n$  ширхэг тоог гараас авч хадгалах функц бол үүнтэй адилаар хүснэгтийн элементүүдийг хэвлэх *print* функцийг бич.

```

void read(int a[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        scanf("%d", &a[i]);
}
void print(int a[], int n)
{
}

```

Дээрх хоёр функцийг ашиглан  $A, B$  хоёр хүснэгтэд тус бүр 5, 3 ширхэг утга гараас авч хэвлэн харуул.

3. Хүснэгтэд байгаа элементүүдийн тоо өгөгдөхөд хамгийн хүснэгтийн элементүүдээс хамгийн ихийг нь олох функцийг бич.

```
int max(int a[], int n);
```

Өмнөх дасгалаа ашиглан  $A, B$  хүснэгтийн хамгийн их элементүүдийг харьцуулан аль хүснэгт хамгийн том тоог агуулж байгааг ол.

4. 5 тооны багыг буцаадаг функцийг тодорхойл. Ингэхдээ хоёр тооны багыг буцаах функцийг ашиглана уу.

```
int min5(int, int, int, int, int);
```

5. Доорх  $a^n$ -ийг олдог функцийн алдаануудыг зас.

```
#include <stdio.h>
int t = 1;
int main() {
    int n, a;
    // a, b-ийн utgiig hereglegchees unshina
    scanf("%d%d", &a, &n);
    // a-ийн n zergiig bodoh
    printf("%d\n", power(a, n));
    // dahin 2 buhel too unshina
    scanf("%d%d", &a, &n);
    // ahiad tednii zergiig olno
    printf("%d\n", power(a, n));
    return 0;
}

void power(int a, int b); {
    int i;
    for (i = 1; i <= b; i++)
        t *= a;

    return t;
}
```

6. Параметрээр дамжуулсан  $a, b$ -ийн хооронд орших анхны тоонуудыг олж, глобалаар зарлагдсан  $A$  хүснэгтэд хадгалж, хэдэн анхны тоо олсныг буцаадаг дараах функцийг тодорхойл. Олсон анхны тоонуудыг  $main()$  функц дотор хэвлэнэ. Уг дасгалын хийхдээ дээр тодорхойлсон  $prime$  функцийг ашиглана.

```
int A[100];
int prime_range(int a, int b);
```

## 1.4.2 Гэрт

- Өгөгдсөн бүхэл тоог палиндром тоо мөн байвал 1 үгүй бол 0-ийг буцаах функц тодорхойл.
- $N$  тооны цифруудийн нийлбэрийг буцаадаг доорх хэрэглэгчийн функцийг ашиглан  $1...N$  завсарт орших цифруудийнхээ нийлбэрт хуваагддаг тоонуудыг ол.

```
int digit_sum(int n);
```

*digit\_sum(1234)* гэвэл уг функц 10-г буцаахыг анхаарна уу.

3. Хэрэглэгчээс хоёр хүснэгтийг уншин, тэдгээрийг *join()* функцрүү дамжуулан эхний хүснэгт дээр хоёр дахь хүснэгтийг залгах доорх функцийг хэрэгжүүл. *join()* функц нь нийлүүлсэн хүснэгтийн хэмжээг буцаана.

```
void read(int A[], int n); // husnegtiin utgiig hereglegchees unshih funkts
void print(int A[], int n); // husnegtiin utgiig hevleh funkts
int join(int A[], int B[], int n, int m) {
    //....
}
int main()
{
    int a[100], b[100], n, m;
    scanf("%d", &n);
    read(a, n);
    print(a, n);
    scanf("%d", &m);
    read(b, m);
    print(b, m);
    int k = join(a, b, n, m);
    print(a, k);
    return 0;
}
```