

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 өмнө нь заавал 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 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 ihiiг 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 a, b;  
    scanf("%d%d", &a, &b);  
    printf("%d\n", max(a, b)); // funktsiin damjuulsan  
                                utgiig  
}
```

```

        // shuud hevlene
        return 0;
    }

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

        return b;
    }

```

Жишээ 2: Функцируу 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-ийн utgiig prime-руу дамжуулж, herev
    // 1-ийг butsaaval k ni anhnii too
    // 0-ийг 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-ийг
            butsaa
            return 0; // tsaash ni urgeljluulj
            shalgah hereg bhgui
    }

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

```

1.4 Дасгалууд

1.4.1 Ангид

1. Хоёр тооны ХИЕХ буцаах доорх функцийг хэрэгжүүл.

```
|| int gcd(int, int);
```

2. 5 тооны багыг буцаадаг функцийг тодорхойл.

```
|| int min(int, int, int, int, int);
```

3. 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;
}
```

4. 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.

```
|| int prime_range(int a, int b);
```

1.4.2 Гэрт

1. Өгөгдсөн бүхэл тоог палиндром тоо мөн байвал 1 үгүй бол 0-ийг буцаах функц тодорхойл.

2. N тооны цифрүүдийн нийлбэрийг буцаадаг доорх хэрэглэгчийн функцийг ашиглан $1 \dots N$ завсарт орших цифрүүдийнхээ нийлбэрт хуваагддаг тоонуудыг ол.

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

3. Дамжуулсан хүснэгтийг эрэмбэлэх, хэрэглэгчээс утга авах, хэвлэх доорх функцүүдийг гүйцээ.

```
void read(int A[], int n);  
void print(int A[], int n);  
void sort(int A[], int n) {  
    //....  
}  
int main()  
{  
    int too[100], n;  
    scanf("%d", &n);  
    read(too, n);  
    sort(too, n);  
    print(too, n);  
    return 0;  
}
```