

# Гарчиг

<b>1</b>	<b>Функцтэй холбоотой нэмэлт боломжууд</b>	<b>3</b>
1.1	Онол ба зорилго . . . . .	3
1.2	Параметрын тоо нь хувьсах функц . . . . .	3
1.2.1	Параметрын тоо нь хувьсах функц зарлах . . . . .	3
1.2.2	Функц тодорхойлох . . . . .	4
1.2.3	Функцийг хэрэглэх . . . . .	5
1.3	Командын мөрөөс аргумент дамжуулах . . . . .	5
1.4	Функцийн заагч . . . . .	5
1.4.1	Функцын заагчийг зарлах . . . . .	6
1.4.2	Функцийн заагчид утга олгох . . . . .	6
1.5	Дасгалууд . . . . .	7
1.5.1	Ангид . . . . .	7
1.5.2	Гэрт . . . . .	8



# Бүлэг 1

## Функцтэй холбоотой нэмэлт боломжууд

### 1.1 Онол ба зорилго

Уг ажлаар бид СИ хэлэнд байдаг функцтэй ажиллах 2 боломжтой танилцах болно. Үүнд:

- Параметрийн тоо нь хувьсах функц
- Функцийн заагч

Хэрэглэгчийн функцийг тогтсон тооны аргументтайгаар тодорхойлохыг бид өмнө нь үзсэн. Мөн бид хувьсан аргументтэй функцийг тодорхойлон ашиглаж болно. Ингэхийн тулд програмын эхэнд `<stdarg.h>` толгой файлыг хавсаргасан байх шаардлагатай. Энэ файл нь параметрийн тоо нь тогтмол биш функцийг боловсруулах дараах төрөл болон функцүүдийг агуулдаг.

### 1.2 Параметрын тоо нь хувьсах функц

#### 1.2.1 Параметрын тоо нь хувьсах функц зарлах

Параметрийн тоо нь хувьсах хэрэглэгчийн функцийг зарлахдаа дараах дүрмээр зарлана. Дүрэм:

`буцаах_төрөл нэр(хуурмаг параметрууд, ...);`

Энд

**буцаах** 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.

**хуурмаг параметрууд** хуурмаг параметруудийн жагсаалтанд эхлээд төрөл ба нэр нь тодорхой байгаа параметруудийг жагсаагаад дараа нь хувьсах тооны параметруудтэй гэдгийг заасан цуваа гурван цэгийг (...) бичнэ. Параметрийн тоо нь хувьсах хэрэглэгчийн функц нь ядаж нэг тодорхой параметртэй байх бөгөөд тодорхой параметруудийн эхнийхээр ихэвчлэн функцэд дамжих параметрийн тоог заадаг.

## 1.2.2 Функц тодорхойлох

Жишээл болгон аргументийн тоо нь хувьсах дундаж олох функцийн кодыг дор харууллаа. Үүнд:

[va\_list arg] нь va\_list нь төрөл зарлах түлхүүр үг, arg нь параметруудийг боловсруулахад ашиглах хувьсагч.

[va\_start(arg, count)] нь arg нь параметруудийг боловсруулахад ашиглах arg хувьсагчид утгуудыг онооно.

[va\_arg(args, double)] double нь төрлийн нэр ба энэ функц хэрэглэгчийн НБ функцийн хандалтанд байгаа жинхэнэ параметруудийн жагсаалтаас ээлжит параметрийг double гэсэн төрөлтөйгөөр arg гэсэн нэрээр уншина.

[va\_end(args)] args хувьсагчийг чөлөөлнө.

**Параметрын тоо нь хувьсах функц:**

```
1  #include <stdio.h>
2  #include <stdarg.h>
3
4  // Function to calculate the average of variable number of arguments
5  double average(int count, ...) {
6      va_list args;           // Variable argument list
7      double sum = 0.0;
8
9      va_start(args, count);  // Initialize variable argument list
10
11     // Summing up all the arguments
12     for (int i = 0; i < count; i++) {
13         sum += va_arg(args, double);
14     }
15
16     va_end(args);           // Clean up the variable argument list
17
18     // Calculating the average
19     return sum / count;
20 }
21
22 int main() {
23     // Calculate average of 3 numbers
24     double avg1 = average(3, 2.5, 3.5, 4.5);
25     printf("Average of 2.5, 3.5, and 4.5 is: %.2f\n", avg1);
26
27     // Calculate average of 5 numbers
28     double avg2 = average(5, 1.0, 2.0, 3.0, 4.0, 5.0);
29     printf("Average of 1.0, 2.0, 3.0, 4.0, and 5.0 is: %.2f\n", avg2);
30
31     return 0;
32 }
33
34 }
```

### 1.2.3 Функцийг хэрэглэх

Функцийг хэрэглэхдээ эхний параметрт араас нь хэдэн параметр өгөхийг заадаг. Түүний араас параметруудийг өгч дуудаж ажиллуулна.

Дүрэм:

```
average(3, 2.5, 3.5, 4.5);
```

## 1.3 Командын мөрөөс аргумент дамжуулах

Си хэл нь үндсэн функцэд командын мөрөөс C программыг ажиллуулах үед аргументуудыг дамжуулах боломжийг олгодог. Эдгээр аргументуудыг програмыг удирдан чиглүүлэхэд ашиглаж болно. Си хэл дээрх командын мөрийн аргументуудыг зарлахдаа `main()`-д хоёр аргументийг тодорхойлж өгнө.

```
int main(int argc, char *argv[]) { /* ... */ }
int main(int argc, char **argv) { /* ... */ }
```

Энд

**argc** нь (ARGument Count) буюу программын нэрийг оруулаад хэрэглэгчийн дамжуулсан командын мөрийн аргументуудын тоог хадгалдаг бүхэл тоон хувьсагч юм. Хэрэв бид програм руу нэг утгыг дамжуулах юм бол argc-ийн утга 2 (нэг нь аргумент, нэг нь програмын нэр) болно. argc-ийн утга нь сөрөг биш байх ёстой.

**argv** нь (ARGument Vector) буюу бүх аргументуудыг жагсаасан тэмдэгт заагчийн массив юм.

Командын мөрийн аргументыг авч хэвлэн харуулах програм:

```
1 // command line arguments
2 #include <stdio.h>
3 int main(int argc, char *argv[]) {
4
5     // argc ni heden argument uudugdsniig zaana
6     // argv      argumentuudiin muriin husnegt
7     printf("Number of arguments: %d\n", argc);
8     //
9     for (int i = 0; i < argc; i++) {
10        printf("Argument %d: %s\n", i, argv[i]);
11    }
12    return 0;
13 }
```

## 1.4 Функцийн заагч

Өмнөх хичээлүүдээр бүхэл тоо, бодит тооноос эхлэн бүтцийн элементийн заагчуудын талаар үзэж танилцсан. Эдгээрээс гадна Си хэлэнд хэрэглэгчийн функцийн төрлийн буюу функцийн заагчийг хэрэглэж болно. Функцийн заагч нь параметрын тоо ба функцийн төрөл нь ижил байх хэд хэдэн хэрэглэгчийн функцийг нэгэн заагч хэрэглэн ажиллуулах боломжийг олгодог. Энэ нь компайл хийгдэх үед бус ажиллах үед сонгогдон ажиллаж болохоороо онцлогтой.

### 1.4.1 Функцийн заагчийг зарлах

Функц заах заагчыг зарлахдаа дараах дүрмээр зарлана. **Дүрэм:**

буцаах-Төрөл (\*функцийн-заагчийн-нэр) (параметрууд);

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.

**\*функцийн-заагчийн-нэр** нь нь функцийн заагчид олгож буй нэр бөгөөд энэ нэрийг хаалтан дотор бичнэ.

**параметрууд** нь дээрх заагчтай холбогдох хэрэглэгчийн функцийн хийсвэр параметрууд юм. Эдгээр параметрууд нь хэрэглэгчийн функцүүдийнхээ параметруудтэй тоо ба төрөл нь таарч байх ёстой.

### 1.4.2 Функцийн заагчид утга олгох

Функцийг ашиглахдаа заагчид хэрэглэгчийн функцүүдийнхээ нэгийн нэрийг дараах байдлаар зааж өгнө.

функцийн-заагч = хэрэглэгчийн-функцийн-нэр;

Ингэснээр уг заагчаар дамжуулан хэрэглэгчийн функцэд хандах боломтой болно. Жишээг дор хавсаргав.

**Функц заагч ашиглан 4 үйлдлийн функцийг сонгох програм:**

```
1 #include <stdio.h>
2 float add(float x,float y) {return (x+y);}
3 float sub(float x,float y) {return (x-y);}
4 float divi(float x,float y) {return (x/y);}
5 float mul(float x,float y) {return (x*y);}
6
7 int main(){
8     int a,b,c;
9     float (*pf)(float m, float n);
10    printf("a=");
11    scanf("%d",&a);
12    printf("b=");
13    scanf("%d",&b);
14    printf("uilde1:");
15    scanf("%d",&c);
16    switch(c)
17    {
18        case 1: pf=add; break;
19        case 2: pf=sub; break;
20        case 3: pf=divi; break;
21        case 4: pf=mul; break;
22    }
```

```
23     printf("result: %.1f",pf(a,b));
24     return 0;
25
26 }
```

## 1.5 Дасгалууд

### 1.5.1 Ангид

1. Хувьсах тооны бүхэл тоон параметер өгөгдөхөд түүнээс хамгийн их элементийг олох функц бич.

```
1 int max_integers(int count, ...)
```

2. Доорх кодыг нэмж өөрчилнэ. Ингэхдээ хүснэгтээр өгөгдсөн бүхэл тоон элементүүдийг функцийн заагчаар дамжуулагдсан функцээс хамааран өсөх болон буурахаар эрэмбэлэх боломжтой болгон гүйцээж хэрэгжүүлнэ үү.

```
1 #include <stdio.h>
2 // usuhuur erembeleh haritsuulah function
3 int compare_asc( /*implement*/ ) {
4     /*implement*/
5 }
6
7 // buurahaar erembeleh haritsuulah function
8 int compare_desc( /*implement*/ ) {
9     /*implement*/
10 }
11
12 //haritsuulah function ni zaagchaar ugugdsun erembeleh function
13 void sort_array(int *arr, int size, int (*compare)(const int *, const int
14     *)) {
15     /* heregjuuleh*/
16
17 }
18
19 //elementuudiig hevleh
20 void print_array(int *arr, int size) {
21     for (int i = 0; i < size; i++) {
22         printf("%d ", arr[i]);
23     }
24     printf("\n");
25 }
26
27 int main() {
28     int arr[] = {5, 2, 7, 1, 9, 3};
29     int size = sizeof(arr) / sizeof(arr[0]);
30
31     printf("Original array: ");
32     print_array(arr, size);
```

```
33
34     // usuhuur erembelj hevlej haruulah
35     sort_array(arr, size, compare_asc);
36     printf("Array sorted in ascending order: ");
37     print_array(arr, size);
38
39     //buurahaar erembelj hevlej haruulah
40     sort_array(arr, size, compare_desc);
41     printf("Array sorted in descending order: ");
42     print_array(arr, size);
43
44     return 0;
45 }
```

### 1.5.2 Гэрт

1. Хувьсах *count* ширхэг тэмдэгт мөрөн параметер өгөгдөхөд тэдгээрийг залган нэг урт тэмдэгт мөр үүсгэн буцаадаг функц бичнэ.

```
1 char *concatenate_strings(int count, ...)
```

2. Ангид өгөгдсөн хоёрдугаар дасгалын эрэмбэлэх функцийг int, char, double төрлийн өгөгдлийг эрэмбэлэх боломжтой болгон авч буй өгөгдлийн төрлүүдийг void болгон өөрчлөнө. Тухайн өгөгдлийн төрлийн харьцуулах функцийг мөн хэрэгжүүлнэ.