

# С++ Хэлний үндсэн ойлголтуудыг ашиглан бодлогуудыг бодох Лабораторийн Ажил (Лаборатори №1)

Н. Мөнхжин

ХШУИС, Компьютерийн ухааны оюутан, nyamkamunhjin@gmail.com

## 1. ОРШИЛ/УДИРТГАЛ

С++ хэлний `#define` макрофункц, дотоод функц, оролт гаралтын урсгалууд `cin`, `cout` үйлдлүүдийг ашиглан 1 хэмжээст хүснэгтийн бодлогуудыг бодох даалгавруудыг хийж гүйцэтгэнэ.

## 2. ЗОРИЛГО

Гараас өгөгдсөн хүснэгтэн утга, дурын бүхэл тоог өгөгдсөн шаардлагын дагуу функцээр дамжуулан үйлдлүүдийг хийж үр дүн гаргаж авах. Үүний тулд дараах зорилтуудыг тавьж ажилласан:

1. `Cin`, `cout` оролт гаралтын урсгалыг ашиглан гараас утга авах, хэвлэх.
2. `#define` зарлагч ашиглан `macro-function` зарлах.
3. Энгийн функц болон дотоод функцийг зарлах, тодорхойлох.

## 3. ОНОЛЫН СУДАЛГАА

### 3.1 Функц

Функц нь багц үйлдлүүдийг цогц хэлбэр юм. Програмыг бичихэд функцийг ашиглах нь олон давтан бичих ижил багц үйлдлүүдыг багасгаж өгдөг.

Функц нь ямар нэгэн төрлийн утгыг авч эсвэл авахгүй байж болох ба түүнээс хамааран багц үйлдлүүдыг гүйцэтгэж утга буцаах эсвэл буцаахгүй байж болдог.

Функц тодорхойлох:

1. Функцыг тодорхойлоход заавал эхлээд утга буцаах төрлийг тодорхойлно.
2. Функц-д нэр өгөх.
3. Параметр утга авах, авахгүй.
4. Функцийг багц үйлдлийг тодорхойлох.
5. Үр дүн буцаах, буцаахгүй.

Жишээ нь: `void countToTen() {`

```
    for(int i=1; i <= 10; i++)
```

```

        cout << I << endl;

    }

```

### 3.2 #define, Macro-function

#define команд нь тогтмол хувьсагчийг програмын pre-processor-д зарлаж өгдөг. Macro-function нь compile хийхээс өмнө уншсан байдаг. Давуу тал нь заагчийг өөрчлөхгүйгээр утгуудыг шууд орлуулан хурдан ажилладаг.[2]

Жишээ нь: #define MAX\_USERS 100;

```
#define max(a, b) a > b ? a : b;
```

### 3.3 Дотоод функц

C++ хэлний inline функц бол программын хурдыг нэмэгдүүлэх зориулалтаар бүтээгдсэн боломж юм. Дотоод мөр функц ба ерийн C функц 2-ийн ялгаа нь дараагийн үйлдэл хийх хаягийг хадгалах, сру-д регистр хадгалах гэх мэт олон үйлдлийг хийхгүйгээр ажиллах тул хурдан ажиллана.[1]

Жишээ: inline int findMax(int arr[], int size) {

```

    ...

    ...

    return max;

}

```

### 3.4 Хаяг

Хаяг нь хувьсагчийн санах ой дээр байрлах хаягыг зааж өгдөг ба ингэснээр хувьсагчийн утгыг хувьсагч ашиглахгүйгээр өөрчилж болдог. Хаягыг з арлахдаа ямар төрлийн хувьсагчийн хаягийг авахаас хамааран төрлөөр нь зарлаж өгдөг.[4]

Жишээ: 1. int a = 5;

```
Int *p = &a;
```

P бол a хувьсагчийн хаягийг хадгална.

## 4. ХЭРЭГЖҮҮЛЭЛТ

### 1. a, b, c гурван тооны хамгийн ихийг олох define макро функц бич.

Жиших нөхцөл ашиглан их, бага утгуудыг тус тус олно.

```
#define max(A, B, C) (((A) > (B) ? (A) : (B)) > (C) ? ((A) > (B) ? (A) : (B)) : (C));
```

## 2. Өгөгдсөн N ширхэг тооны хамгийн их/хамгийн багийг олох код бич.

Хүснэгтээс давталтаар их, багыг олох нөхцөл ашиглан олно.

## 3. Өгөгдсөн хүснэгтийн тэгш элементүүдийн тоог олох дотоод функц бич.

Тоог 2-т хуваахад үлдэгдэл 0-той тэнцэж байвал тэгш тоо мөн.

Хүснэгтийн элементүүдийг давталтаар үлдэгдэл 0 той тэнцэж байгааг тоолно.

## 4. Өгөгдсөн хүснэгтийн сондгой элементүүдийн арифметик дунджийг олох дотоод функц бич.

Тоог 2-т хуваахад үлдэгдэл 1-тэй тэнцэж байвал сондгой тоо мөн.

Хүснэгтийн элементүүдийг давталтаар үлдэгдэл 1-тэй тэнцэж байгаа тооний нийлбэрүүдийг олно

## 5. Өгөгдсөн натурал тооны оронгийн тоо болон цифрүүдийг олох дотоод функц бич.

Даалгаврын 5-р бодлогын хүрээнд функц олон утга буцаах шаардлага гарсан тул хаягыг ашиглан шийдэв. Харин тоог цифрээр задлахын тулд % 10-т хувааж өгч олно. Дараа нь 10 хувааснаар сүүлчийн цифр тооноос хасагдана.

## 5. ДҮГНЭЛТ

Дээрх даалгавруудын хүрээнд cin, cout урсгалаар гараас утга авч хэвлэж, #define макро, inline функц ашиглан энгийн функцүүдийг бүтцийн хувьд илүү цэгцтэй, хугацааны хувьд хэмнэлттэй болгож сурав.

## 6. АШИГЛАСАН МАТЕРИАЛ

1. Объект хандлагат технологийн C++ програмчлал, Ж.Пүрэв, 2008, Улаанбаатар. Хуудас 72
2. <https://gcc.gnu.org/onlinedocs/cpp/Macro-Arguments.html>
3. [https://www.techonthenet.com/c\\_language/constants/create\\_define.php](https://www.techonthenet.com/c_language/constants/create_define.php)
4. <https://fresh2refresh.com/c-programming/c-pointer/>

## 7. ХАВСРАЛТ

```
/*  
1. a, b, c гурван тооны хамгийн ихийг олох define макро функц бич.  
*/  
#include <stdio.h>  
#include <iostream>  
using namespace std;
```

```
#define max(A, B, C) (((A) > (B) ? (A) : (B)) > (C) ? ((A) > (B) ? (A) : (B)) : (C));
```

```
int main() {  
    int a, b, c;  
    cout << "input: ";  
    cin >> a >> b >> c;  
    cout << max(a, b, c);  
}
```

```
/*  
2. Өгөгдсөн N ширхэг тооны хамгийн их/хамгийн багийг олох код бич.  
*/
```

```
#include <stdio.h>  
#include <iostream>  
using namespace std;
```

```
int max(int *arr, int n) {  
    int max = arr[0];  
    for(int i = 0; i < n; i++)  
        if(arr[i] > max)  
            max = arr[i];  
  
    return max;  
}
```

```
int min(int *arr, int n) {  
    int min = arr[0];  
    for(int i = 0; i < n; i++)  
        if(arr[i] < min)  
            min = arr[i];  
  
    return min;  
}
```

```
int main() {  
    int arr[5] = {1, 2, -5, 3, 21};  
  
    cout << "max: " << max(arr, 5) << endl;  
    cout << "min: " << min(arr, 5) << endl;  
}
```

```
/*  
3. Өгөгдсөн хүснэгтийн тэгш элементүүдийн тоог олох дотоод функц  
бич.  
*/
```

```
#include <stdio.h>  
#include <iostream>  
using namespace std;  
  
inline int evenCount(int *arr, int n) {  
    int counter = 0;  
    for(int i = 0; i < n; i++)  
        if(arr[i] % 2 == 0)  
            counter++;  
  
    return counter;  
}  
  
int main() {  
    int arr[6] = {0, 1, 2, -5, 6, 21};  
  
    cout << "Even number: " << evenCount(arr, 6) << endl;  
}
```

```
/*  
4. Өгөгдсөн хүснэгтийн сондгой элементүүдийн арифметик дунжийг  
олох дотоод функц бич.  
*/
```

```
#include <stdio.h>  
#include <iostream>  
using namespace std;  
  
inline float oddAvg(int *arr, int n) {  
    int total = 0;  
    int counter = 0;  
    for(int i = 0; i < n; i++)  
        if(arr[i] % 2 == 1 || arr[i] % 2 == -1) {  
            total += arr[i];  
            counter++;  
        }  
    return counter > 0 ? (float)total / counter : 0;  
}
```

```

        counter++;
    }

    return (float) total/counter;
}

int main() {
    int arr[6] = {0, 1, 2, -5, 6, 22};

    cout << "Avg of odds: " << oddAvg(arr, 6) << endl;
}

```

```

/*
5. Өгөгдсөн натурал тооны оронгийн тоо болон цифрүүдийг олох
дотоод функц бич.
*/

```

```

#include <stdio.h>
#include <iostream>
using namespace std;

inline int* countDigit(int num) {
    int numCpy = num;
    int counter = 0;
    int *arr = new int[100];
    // count
    while(numCpy != 0) {
        arr[counter] = numCpy % 10;
        //      cout << arr[counter] << endl;
        numCpy /= 10;
        counter++;
    }

    int *saveArr = new int[counter + 1];
    saveArr[0] = counter + 1;

    for(int i = 1; i < counter + 1; i++)
        saveArr[i] = arr[counter - i];

    delete arr;
    return saveArr;
}

```

```
int main() {  
    int num = 123456;  
    int *arr = countDigit(num);  
    for(int i = 1; i < arr[0]; i++)  
        cout << arr[i] << " ";  
}
```