

“Програмчлалын хэл Си” хичээлээ сэргээн C++ хэлийг ашиглаж сурах

(Лаборатори №1)

Э.Түвшинтөгс | 20B1NUM0448

МУИС - ХШУИС, Компьютерийн Ухаан, 3-р түвшин, tuvshintugs@mail.com

1. ОРШИЛ

Өмнө нь сурсан C хэлний хичээлээ сэргээж, C++ хэл дээрх өөрчлөгдсөн шинэ объектууд болон шинээр ашиглаж сурсан толгой файлуудаа хэрэгжүүлэн лабораторийн дасгалуудыг бодно.

2. ТАЙЛАНГИЙН ЗОРИЛГО

C++ хэл дээрх суурь шаардлагатай функцууд болон оролт гаралтын объектуудыг болон бодлогод ашиглаж буй namespace, istream зэрэг толгой файлуудыг ашиглалтыг харуулах. Бодлогын нөхцөлд шаардагдаж буй нөхцөл, хэрэглэгчийн функц, түүний шаардлагын дагуу кодыг тайлагнах.

3. ОНОЛЫН СУДАЛГАА

3.1 Input, output

C болон C++ хэл дээр буй эхний ялгаа нь оролт, гаралтын объектууд.

cin – console input гэсэн үгний товчлол бол cout – console output гэсэн үгний товчлол болдог. Эдгээр нь тухайн програм, кодлуу кэйбордоор бичиж утга оруулах, дэлгэцэнд тухайн код, програмыг харуулах үүргүүдтэй.

cin нь “istream” классын объект бол cout нь “ostream” классын объект. Энэ 2 нийлээд istream толгой файлд тодорхойлогддог. Input нь >> ашигладаг бол, Output нь << ашиглана.

3.2 Функц

Функц гэдэг нь ямар нэгэн үйлдэлд зориулагдсан, тухайн үйлдэлд ямар нэгэн үүрэг гүйцэтгэдэг хэсэг код юм. Функци бичснээр тухайн хэсэг кодоо ахин дахин бичих шаардлагагүй болох бөгөөд, хүссэн үедээ дуудаж ашиглах боломжтой. Хэрвээ бичиж буй функцияа зарлахдаа void гэж зарлавал утга буцаахгүй. Харин утга буцаах шаардлагатай гэж үзвэл тухайн буцаах утгын төрлийг бичиж өгнө. Жишээ нь int, double гэх мэт.

3.3 Bubble Sort

Хүснэгтийн элементүүдийг эхнээс нь хоёр талд нь байгаа элементүүдтэй жишиж, дараагийн элементээсээ их байна уу эсвэл бага байна уу гэх мэт алхамуудаар явсаар хүснэгтийн хэмжээ хэд байна, тэр хүртэл явсаар зөв дараалалд нь оруулахыг хэлнэ.

Хамгийн амархан тайлбарлах арга гэвэл $a[j] > a[j+1]$ гэсэн нөхцөл хангагдвал тухайн элементүүдийн байрыг солино гэсэн үг. Хэрэв нөхцөл хангагдахгүй бол хэвээр үлдээж, хамгийн их элемент хүснэгтийн хамгийн сүүлийн байранд үлдэнэ гэсэн үг. Хэрэв хамгийн багыг олох бол эсрэгээрээ.

4. ХЭРЭГЖҮҮЛЭЛТ

I) a, b, c гурван тооны хамгийн ихийг олж дэлгэцэнд хэвэл.

Гурван тоогоо хадгалах хувьсагч болон хамгийн ихийг нь олсон утгаа хадгалах хувьсагчуудыг зарлаж өгсөн. Түүний дараагаар 3 тоогоо жиших үйлдлийг хийсэн. Хамгийн эхлээд a тоог бусад 2 тоотой нь жишсэн. Хэрэв a тоо b, c тоонуудаасаа их үед a тоог хамгийн их тоо гэж үзэн тах-ийн утгад хадгалан улмаар хамгийн их мөн гэж үзэн a тоог /хамгийн их/ дэлгэцэд харуулна. Хэрэв дээрх 2 нөхцөлийн аль нэг нь л биелээгүй тохиолдолд мөн адил b тоон дээр жиших үйлдэл ашиглана. Гэхдээ энэ удаад a тоотой жиших шаардлага байхгүй. Учир нь b тоо нь a тооноос бага гэдэг нь баттай болсон учир зөвхөн c тоотой л жишнэ. Биелж байвал тах-ийн утгад b-г өгнө. Мөн л нөхцөл биелээгүй тохиолдолд үлдсэн ганц боломж буюу c-г шууд хамгийн их гэж үзэн тах-ийн утгад оноож өгнө. Эцэст нь тах-ийн утгад ямар тоо оноогдсон, тэр тоог дэлгэцэд хамгийн их гэж харуулна.

```
cin >> a >> b >> c;
// Хамгийн ихийг нь олох 3 тоогоо гараас авна.

if(a > b && a > c){
    // if нөхцөл болон логик үйлдэл && нарыг ашиглан a тоо нь b болон c тоонуудаас их эсэхийг шалгана.
    max = a;
    // хэрэв a тоо нь b, c тоонуудаас их байвал тах хувьсагчид хадгалж өгнө.
} else if (b > c){
    // Харин дээрх нөхцөл хангагдаагүй үед else if-ийг ашиглан b тоо c-ээс их байгаа эсэхийг шалгана.
    max = b;
    // b тоо нь c-ээс их гэсэн тохиолдолд b-ийн утгыг тах хувьсагчдаа оноож өгч хадгална.
} else{
    max = c;
    // Дээрх бүх нөхцөл хангагдаагүй бол c нь хамгийн их тоо болох учир тах гэсэн хувьсагчид c хадгалж өгнө.
}

cout << "Hamgiin ih: " << max;
// Олсон хамгийн их утгаа харуулж байна.
```

Жишээ:

Оролт:

a = 11, b = 29, c = 2002

Хамгийн их: 2002

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\tuvsh\Desktop\oop\Lab 01> ./1
11 29 2002
Hamgiin ih: 2002
PS C:\Users\tuvsh\Desktop\oop\Lab 01> 
```

II) Тойргийн радиусыг гараас авч талбайг бодож хэвэл.

Тойргийн радиусыг гараас авсан. Мөн пи тооны утгыг зарлаж өгсөн. Мөн талбай нь бодит тоо байх магадлалтай учир талбайг double-аар зарласан.

Гараас радиусыг авч, тойргийн талбайг олох томъёогоор талбайг бодож олсон.

```
int main(){
    int r;
    // Гараас авах радиусыг хадгалах хувьсагчийг зарлаж байна.
    int pi = 3.1415;
    // π тоогийн утгыг зарлаж өгч байна.
    double s;
    // Олсон талбайг хадгалах хувьсагч зарлаж байна.

    cin >> r;
    // Радиусыг гараас авна.

    s = pi * (r * r);
    // Гурвалжингийн талбайг олох  $A = \pi r^2$  томъёогоор талбайг бодож олно.

    cout << "Talbai: " << s;
    // Олсон талбайгаа дэлгэцэнд харуулна.

    return 0;
}
```

Жишээ:

Радиус : 9

Талбай: 254.469

```
PS C:\Users\tuvsh\Desktop\oop\Lab 01> ./2
9
Talbai: 254.469
PS C:\Users\tuvsh\Desktop\oop\Lab 01> █
```

III) Тэгш өнцөгтийн талуудыг өгөхөд талбайг буцаадаг хэрэглэгчийн функц бич.

Гараас авсан тэгш өнцөгтийн урт болон өргөнийг тэгш өнцөгтийн талбайг олох RectangleArea функц ашиглан, функцдээ дуудаж тухайн тэгш өнцөгтийн талбайг олох функцийг бичсэн. Мөн талбайг олж, буцаах функц бич гэсэн учир талбайг олох функцээ зарлахдаа int ашиглан дотор нь шууд талбайг олж буцаах томъёоны орлуулга хийж буцаасан.

```

int RectangleArea(int a, int b){
    return (a * b);
    // Өгөгдсөн урт, өргөнийг үржүүлэн тэгш өнцөгтийн талбайг буцааж байна.
}

int main(){
    int a, b;
    // Гараас авч буй тэгш өнцөгтийн талуудыг хадгалах хувьсагчийг зарлана.

    cin >> a >> b;
    // Тэгш өнцөгтийн талуудыг гараас авч байна.

    /* Талуудыг ӨГӨХ гэхэд нь шууд хувьсагчдад утга онооё гэж бодсон ч гараар авахаар шийдэв.*/

    cout << "Talbai: " << RectangleArea(a, b);
    // Дээр бичсэн талбайг буцаадаг хэрэглэгчийн функцаа ашиглан талбайг олон утгыг хэвлэж байна.

    return 0;
}

```

Жишээ:

Урт: 15

Өргөн: 10

Талбай: 150

```

PS C:\Users\tuvsh\Desktop\oop\Lab 01> ./3
15
10
Talbai: 150
PS C:\Users\tuvsh\Desktop\oop\Lab 01> 

```

IV) Өгөгдсөн N хүртэлх тооны нийлбэр, арифметик дундажыг ол.

Гараас хэд хүртэлх тоог авахыг оруулсан. Дараагаар нь нийлбэр болон дунджийг нь олохдоо 1-ээс N хүртэл нэг нэгээр нэмэгдэж давтах үйлдлийг гүйцэтгэсэн. Түүнийгээ нийлбэрийг хадгалах sum гэсэн хувьсагчид 1-ээр нэмэгдэж N хүрэх хүртэл хадгалснаар N хүртэлх тооны нийлбэрийг олсон. Харин дунджийг олохдоо олсон нийлбэрээ нийт тоондоо буюу N-дээ хувааж өгөн дунджийг олсон.

```

int main() {
    int n, sum = 0;
    // Гараас авах n, n тоо хүртэлх тоонуудын нийлбэрийг хадгалах хувьсагч зарлана.

    cin >> n;
    // n тоог гараас авна.

    for (int i = 1; i <= n; i++) {
        sum = i + sum;
    }
    // 1-ээс эхлэн n тоо хүртэлх тоонуудын нийлбэрийг олно.

    cout << "Niilber: " << sum << endl;
    // Тоонуудын нийлбэрийг хэвлэнэ.

    cout << "Dundaj: " << (double)sum / n;
    // Тоонуудын нийлбэрийг олно. Олохдоо n хүртэлх тоонуудын нийлбэрийг n тоондоо хуваана.

    return 0;
}

```

N тоо: 20

N хүртэлх тоонуудын нийлбэр: 210

Дундаж: 10.5

```

PS C:\Users\tuvsh\Desktop\oop\Lab 01> ./4
20
Niilber: 210
Dundaj: 10.5
PS C:\Users\tuvsh\Desktop\oop\Lab 01> 

```

V) Гараас өгөгдсөн N ширхэг тооны нийлбэр, үржвэрийг ол.

Эхлээд хэдэн тоо авах, мөн оруулах тоонууд болон нийлбэр, үржвэрийг хадгалах хувьсагчдыг зарлаж өгсөн.

Дараа нь N ширхэг тоогоо авч, гараас тоо авах бүртээ тухайн тоонуудыг нэмэх болон авсан тоонуудыг үржүүлэх үйлдлүүдийг N удаа давтан нийлбэр болон үржвэрүүдийг олсон.

```

int main(){
    int n, x, sum = 0, multiply = 1;

    cin >> n;
    // Гараас нийт хэдэн тоо авахыг оруулж өгнө.

    for(int i = 0; i < n; i++){
        cin >> x;
        // Гараас тоонуудыг авна.
        sum = x + sum;
        // Гараас авсан тоонуудыг өмнөх тоонуудтайгаа нэмж байна.
        multiply = x * multiply;
        // Гараас авсан тоонуудаа үржиж байна.
    }
}

```

N тоо:

x: 12, 3, 4, 5, 6, 7

Нийлбэр: 37

Үржвэр: 30240

```

6
12
3
4
5
6
7
Nilber: 37
Urjver: 30240
PS C:\Users\tuvsh\Desktop\oop\Lab 01> 

```

VI) Өгөгдсөн N ширхэг тооны хамгийн их/хамгийн багийг ол.

Эхлээд хэдэн тоо авахаа оруулаад тухайн оруулж буй тоонуудын хамгийн эхний тоог хамгийн их, бага утгыг хадгалж буй хувьсагчууддаа оноож өгсөн. Харин үүний дараагаар нөхцөлт давталтаар хүснэгтийн хоёр дахь элементээс эхлэн N хүртэлх тоонуудыг нэг бүрчлэн оруулж, оруулах бүртээ min, max-ийн утгатай жишиж байсан. Ингэснээр эцэст нь хамгийн их утгад хамгийн их тоо хадгалагдаж, хамгийн бага утгад ч мөн хамгийн бага тоо хадгалагдана.

```

int main() {
    int n, x, min, max;
    // Хэдэн тоо авах, оруулах тоонууд, хамгийн бага, их тоонуудыг хадгалах хувьсагч зарлаж өгнө.

    cin >> n;
    // Хэдэн тоо оруулахаа гараас оруулж өгнө.

    cin >> x;
    // N ширхэг тоог оруулна.
    min = max = x;
    // Хамгийн их болон бага тоог хадгалах хувьсагчид хамгийн эхний тоогоо хадгалж өгнө.

    for (int i = 1; i < n; i++) {
        cin >> x;

        if (x > max)
            max = x;
        // Давталтад харгалзаж буй тоо нь max-аас их байвал тухайн тоог их гэж үзээд max-ийн утгад өгнө.
        if (x < min)
            min = x;
        // Давталтад харгалзаж буй тоо нь min-ээс их байвал тухайн тоог их гэж үзээд min-ийн утгад өгнө.
    }
}

```

N тоо: 4

x: 12, 8, 9, 1

Хамгийн их: 12

Хамгийн бага: 1

```

PS C:\Users\tuvsh\Desktop\oop\Lab 01>

./6

4
12 8 9 1
Hamgiin ih: 12
Hamgiin бага: 1
PS C:\Users\tuvsh\Desktop\oop\Lab 01> 

```

VII) Өгөгдсөн хүснэгтийн тэгш элементүүдийн тоог ол.

Хүснэгтийг өгөгдсөн гэсэн учир ямар ч хамаагүй нэг хүснэгтийг үүсгэсэн.

Дараа нь тухайн хүснэгтийн хэмжээ хэд байгаа хүртэл тухайн хүснэгтэд буй тоонуудыг нэг нэгээр нь 2-т үлдэгдэлтэй хуваахад 0 гарч байгаа эсэхийг нь шалган 0 гарч байвал тэгш гэж үзээд, тэгш тоонуудыг тоолох хувьсагчаа 1-ээр нэмэгдүүлсэн.

```

int a[] = {2, 0, 0, 2, 1, 1, 2, 9, 11, 29};
// Өгөгдсөн хүснэгт

int n = sizeof(a) / sizeof(a[0]);
// Өгөгдсөн хүснэгтэд хэдэн элемент байгааг олно.

int even = 0;
// Тэгш элементүүдийн тоог хадгалах хувьсагч зарлана.

for(int i = 0; i < n; i++){
    // Хүснэгтийн тоо бүрээр давтана.
    if (a[i] % 2 == 0){
        // Тухайн тоо тэгш эсэхийг шалгана. Үлдэгдэлтэй 2-т хуваахад хариу нь 0 гарч байвал тэгш тоо гэж үзнэ.
        even++;
        // Тэгш тоо олдсон учир тэгш тооны хувьсагчаа нэгээр нэмэгдүүлнэ.
    }
}

```

Хүснэгт: 2, 0, 0, 2, 1, 1, 2, 9, 11, 29

Тэгш элементүүдийн тоо: 5

VIII) Өгөгдсөн хүснэгтийн арифметик дунджийг олдог хэрэглэгчийн функц бич.

Өгөгдсөн хүснэгтийн дунджийг олохын тулд нийлбэр болон хэдэн тоо байгааг эсэх нь хэрэгтэй тул нийлбэр хадгалах болон хэмжээг хадгалах хувьсагчууд зарлан нийлбэрүүдийг олсон. Дараа нь нийлбэрээ, хүснэгтийн хэмжээндээ хувааж дундаж утгыг олсон.

```

float average(int a[], int size) {
    float sum = 0;
    // Хүснэгтэд байгаа тоонуудын нийлбэрийг хадгалах хувьсагч
    for (int i = 0; i < size; i++) {
        sum += a[i];
    }
    // Хүснэгтэд байгаа тоонуудын нийлбэрийг олох давталт

    return sum / size;
    // Нийлбэрийг нь нийт хэдэн тоо байгаад хуваана.
    // Ингэснээр дундаж утга буцаагдах болно.
}

int main() {
    int a[] = {2, 0, 0, 2, 1, 1, 2, 9, 11, 29};
    // Өгөгдсөн хүснэгт
    int size = sizeof(a) / sizeof(a[0]);
    // Өгөгдсөн хүснэгтэд хэдэн элемент байгааг олно.

    cout << "Ugugdsun husnegtiin dundaj: " << average(a, size);
    // Хэрэглэгчийн функцаа ашиглан дундаж утгыг хэвлэнэ.
    return 0;
}

```

Хүснэгт: 2, 0, 0, 2, 1, 1, 2, 9, 11, 29

Дундаж: 5.7


```
PS C:\Users\tuvsh\Desktop\oop\Lab 01> ./8
Ugugdsun husnegtiin dundaj: 5.7
PS C:\Users\tuvsh\Desktop\oop\Lab 01> █
```

IX) Хүснэгтийг өсөх/буурахаар эрэмбэл.

Bubble Sort-ийн аргыг ашиглаж өсөх, буурахаар эрэмбэлсэн.

Bubble Sort-ийн арга нь тухайн хүснэгтэд буй тоонуудын болон элементүүдийг буруу дараалалд байвал нөхцлийн дагуу зөв дараалалд ортол ажилладаг эрэмбэлэх арга.

Доорх тохиолдолд эхлээд хамгийн их, бага тоог олж хамгийн сүүлийн байрлалд аваачиж тавьсан. Улмаар тэрний өмнөх хамгийн бага, их тоонуудыг олох гэх мэт дарааллар явсаар бүх өгөгдсөн тоонуудыг зөв дараалалд оруулсан.

```
for (int i = 0; i < n - 1; i++) {
    // n - 1 хүртэл удаа давтаж тухайн хүснэгтийн хамгийн их тоог хамгийн сүүлчийн байрлалд аваачна.
    for (int j = 0; j < n - i - 1; j++) {
        // Давхар давталт нь n - i - 1 удаа ажиллана. Учир нь хамгийн их тоо нь гаднах давталтад хамгийн сүүлийн байрлалд аваачна.
        if (a[j] > a[j + 1]) {
            // a[j] -д байгаа утга a[j + 1] -ээс их байвал a[j], a[j + 1] 2-ийн байрыг сольж дарааллыг өөрчилнө.
            int temp = a[j];
            a[j] = a[j + 1];
            a[j + 1] = temp;
        }
    }
}

cout << "Усuh: " << endl;
for (int i = 0; i < n; i++) {
    cout << a[i] << " ";
}

cout << endl;

for (int i = 0; i < n - 1; i++) {
    // n - 1 хүртэл удаа давтаж тухайн хүснэгтийн хамгийн бага тоог хамгийн сүүлчийн байрлалд аваачна.
    for (int j = 0; j < n - i - 1; j++) {
        // Давхар давталт нь n - i - 1 удаа ажиллана. Учир нь хамгийн бага тоо нь гаднах давталтад хамгийн сүүлийн байрлалд аваачна.
        if (a[j] < a[j + 1]) {
            // a[j] -д байгаа утга a[j + 1] -ээс бага байвал a[j], a[j + 1] 2-ийн байрыг сольж дарааллыг өөрчилнө.
            int temp = a[j];
            a[j] = a[j + 1];
            a[j + 1] = temp;
        }
    }
}

cout << "Buurah: " << endl;
for (int i = 0; i < n; i++) {
    cout << a[i] << " ";
}

cout << endl;
```

Хүснэгт:

0, 2, 2, 8, 15, 2, 3, 4, 29, 11

Үр дүн:

```
PS C:\Users\tuvsh\Desktop\oop\Lab 01> ./9
Usuh:
0 2 2 2 3 4 8 11 15 29
Buurah:
29 15 11 8 4 3 2 2 2 0
PS C:\Users\tuvsh\Desktop\oop\Lab 01> █
```

X) Өгөгдсөн тоо анхны тоо эсэхийг шалга.

Анхны тоо гэж өөртөө болон 1-т хуваагддаг тоог хэлдэг. 0 болон 1 нь өөрсдөө анхны тоо биш учраас өгөгдсөн тоо нь 0, 1 байх юм бол шууд анхны тоо биш гэж үзнэ. Дараагаар нь давталтыг 2-оос эхлүүлнэ. Учир нь 2 бол хамгийн бага анхны тоо. Мөн өгөгдсөн тоо хүртэл давтаж тухайн өгөгдсөн тоо нь давталтын тоологчийг үлдэгдэлтэй хуваагаад 0 гарч байвал анхны тоо гэж үзнэ. Учир нь энэ нь өөр тоонд хуваагдаж байгаа гэсэн үг болох учир тухайн тоо анхны тоо биш гэж үзээд биш буюу 0-ийг буцаана. Харин эсрэг тохиолдол буюу зөвхөн өөртөө болон 1-т хуваагдаж байвал анхны тоо мөн гэж үзэн 1-ийг буцаана.

```
int isPrime(int n){
    // Анхны тоо мөн эсэхийг шалгах функцийг зарлана.
    if(n <= 1){
        return 0;
    }
    // Хэрвээ өгөгдсөн тоо 1 болон түүнээс бага байвал 0-ийг буцаана.
    // Учир нь 1 болон 0 нь анхны тоо биш.

    for(int i = 2; i <= n/2; i++){
        // Өгөгдсөн тоог хуваадаг тоонуудыг шалгана.
        if(n % i == 0){
            return 0;
            // Хэрвээ өгөгдсөн тоо ямар нэгэн тоонд үлдэгдэлгүй хуваагдаж байвал анхны тоо биш.
            // Тиймээс анхны тоо биш гэж үзэн 0-ийг буцаана.
        }
    }

    return 1;
    // Харин эсрэгээрээ буюу ямар нэгэн тоонд хуваагдахгүй байвал 1-ийг буцаана.
    // Энэ нь тус тоо анхны тоо мөн гэсэн үг.
}
```

Өгсөн анхны тоо:

123, Анхны тоо биш

13, Анхны тоо мөн

```
PS C:\Users\tuvsh\Desktop\oop\Lab 01> ./10
123
123 Anhnii too bish
PS C:\Users\tuvsh\Desktop\oop\Lab 01> ./10
13
13 Anhnii too
PS C:\Users\tuvsh\Desktop\oop\Lab 01> █
```

5. ДҮГНЭЛТ

C++ хэлийг ашиглан өмнө нь үзсэн C хэлээ сэргээж оролт, гаралтын үйлдлүүд болох функц гэх мэт C++ хэл дээрх өөрчлөлтүүдтэй дасан ажиллаж, илүү хурдацтайгаар, хялбар хийх боломжтой боллоо. Хэрэглэгчийн функц ашиглаж сурснаар кодыг олон дахин бичихгүйгээр нэг бичсэн кодоо хялбар хэрэглэхээр болов.

6. АШИГЛАСАН МАТЕРИАЛ

<https://www.geeksforgeeks.org/bubble-sort/>

https://www.w3schools.com/c/c_functions.php

<https://www.tutorialology.com/cplusplus/data-types/>

7. ХАВСРАЛТ

Github Link: <https://github.com/Tuvshintugsgccc/ICSI201/tree/main/Lab%2001>

```
1.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  // Лаб 1)
5  // 1. a, b, c гурван тооны хамгийн ихийг олж дэлгэцэнд хэвэл.
6
7  int main() {
8      int a, b, c, max;
9      // Гараас авах 3 тооны хувьсагч болон хамгийн
10     // их тоог нь хадгалах хувьсагчийг зарлаж өгч байна.
11
12     cin >> a >> b >> c;
13     // Хамгийн ихийг нь олох 3 тоогоо гараас авна.
14
15     if(a > b && a > c){
16         // if нөхцөл болон логик үйлдэл && нарыг ашиглан a тоо нь b болон c тоонуудаас их эсэхийг шалгана.
17
18         max = a;
19         // хэрэв a тоо нь b, c тоонуудаас их байвал max хувьсагчид хадгалж өгнө.
20     } else if (b > c){
21         // Харин дээрх нөхцөл хангагдаагүй үед else if-ийг ашиглан b тоо c-ээс их байгаа эсэхийг шалгана.
22         max = b;
23         // b тоо нь c-ээс их гэсэн тохиолдолд b-ийн утгыг max хувьсагчдаа оноож өгч хадгална.
24     } else{
25         max = c;
26         // Дээрх бүх нөхцөл хангагдаагүй бол c нь хамгийн их тоо болох учир max гэсэн хувьсагчид c хадгалж өгнө.
27     }
28
29     cout << "Hamgiin ih: " << max;
30     // Олсон хамгийн их утгаа харуулж байна.
31
32     return 0;
33 }
```

```

2.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  // Лаб 1)
5  // 2. Тойргийн радиусыг гараас авч талбайг бодож хэвэл.
6
7  int main(){
8
9      int r;
10     // Гараас авах радиусыг хадгалах хувьсагчийг зарлаж байна.
11     int pi = 3.1415;
12     //  $\pi$  тоогийн утгыг зарлаж өгч байна.
13     double s;
14     // Олсон талбайг хадгалах хувьсагч зарлаж байна.
15
16     cin >> r;
17     // Радиусыг гараас авна.
18
19     s = pi * (r * r);
20     // Гурвалжингийн талбайг олох  $A = \pi r^2$  томъёогоор талбайг бодож олно.
21
22     cout << "Talbai: " << s;
23     // Олсон талбайгаа дэлгэцэнд харуулна.
24
25     return 0;
26 }

```

```

3.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  // Лаб 1)
5  // 3. Тэгш өнцөгтийн талуудыг өгөхөд талбайг буцаадаг хэрэглэгчийн функц бич.
6
7  int RectangleArea(int a, int b){
8      return (a * b);
9      // Өгөгдсөн урт, өргөнийг үржүүлэн тэгш өнцөгтийн талбайг буцааж байна.
10 }
11
12 int main(){
13     int a, b;
14     // Гараас авч буй тэгш өнцөгтийн талуудыг хадгалах хувьсагчийг зарлана.
15
16     cin >> a >> b;
17     // Тэгш өнцөгтийн талуудыг гараас авч байна.
18
19     /* Талуудыг ӨГӨХ гэхэд нь шууд хувьсагчдад утга онооё гэж бодсон ч гараар авахаар шийдэв.*/
20
21     cout << "Talbai: " << RectangleArea(a, b);
22     // Дээр бичсэн талбайг буцаадаг хэрэглэгчийн функцаа ашиглан талбайг олон утгыг хэвлэж байна.
23
24     return 0;
25 }
26

```

```

4.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n, sum = 0;
6      // Гараас авах n, n тоо хүртэлх тоонуудын нийлбэрийг хадгалах хувьсагч зарлана.
7
8      cin >> n;
9      // n тоог гараас авна.
10
11     for (int i = 1; i <= n; i++) {
12         sum = i + sum;
13     }
14     // 1-ээс эхлэн n тоо хүртэлх тоонуудын нийлбэрийг олно.
15
16     cout << "Niilber: " << sum << endl;
17     // Тоонуудын нийлбэрийг хэвлэнэ.
18
19     cout << "Dundaj: " << (double)sum / n;
20     // Тоонуудын нийлбэрийг олно. Олохдоо n хүртэлх тоонуудын нийлбэрийг n тоондоо хуваана.
21
22     return 0;
23 }

```

```

5.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  // Лаб 1)
5  // 5. Гараас өгөгдсөн N ширхэг тооны нийлбэр, үржвэрийг ол.
6
7  int main(){
8      int n, x, sum = 0, multiply = 1;
9
10     cin >> n;
11     // Гараас нийт хэдэн тоо авахыг оруулж өгнө.
12
13     for(int i = 0; i < n; i++){
14         cin >> x;
15         // Гараас тоонуудыг авна.
16         sum = x + sum;
17         // Гараас авсан тоонуудыг өмнөх тоонуудтайгаа нэмж байна.
18         multiply = x * multiply;
19         // Гараас авсан тоонуудаа үржиж байна.
20     }
21
22
23     cout << "Niilber: " << sum << endl;
24     // Олсон нийлбэрийг хэвлэнэ.
25     cout << "Urjver: " << multiply;
26     // Олсон үржвэрийг хэвлэнэ.
27
28     return 0;
29 }

```

```

6.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  // Лаб 1)
5  // 6. Өгөгдсөн N ширхэг тооны хамгийн их/хамгийн багийг ол.
6
7  #include <iostream>
8  using namespace std;
9
10 int main() {
11     int n, x, min, max;
12     // Хэдэн тоо авах, оруулах тоонууд, хамгийн бага, их тоонуудыг хадгалах хувьсагч зарлаж өгнө.
13
14     cin >> n;
15     // Хэдэн тоо оруулахаа гараас оруулж өгнө.
16
17     cin >> x;
18     // N ширхэг тоог оруулна.
19     min = max = x;
20     // Хамгийн их болон бага тоог хадгалах хувьсагчид хамгийн эхний тоогоо хадгалж өгнө.
21
22     for (int i = 1; i < n; i++) {
23         cin >> x;
24
25         if (x > max)
26             max = x;
27         // Давталтад харгалзаж буй тоо нь max-аас их байвал тухайн тоог их гэж үзээд max-ийн утгад өгнө.
28         if (x < min)
29             min = x;
30         // Давталтад харгалзаж буй тоо нь min-ээс их байвал тухайн тоог их гэж үзээд min-ийн утгад өгнө.
31     }
32
33
34     cout << "Hamgiin ih: " << max << endl;
35     cout << "Hamgiin бага: " << min << endl;
36     // Хамгийн их болон бага утгуудыг хэвлэнэ.
37
38     return 0;
39 }

```

```

7.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  // Лаб 1)
5  // 7. Өгөгдсөн хүснэгтийн тэгш элементүүдийн тоог ол.
6
7  int main(){
8
9      int a[] = {2, 0, 0, 2, 1, 1, 2, 9, 11, 29};
10     // Өгөгдсөн хүснэгт
11
12     int n = sizeof(a) / sizeof(a[0]);
13     // Өгөгдсөн хүснэгтэд хэдэн элемент байгааг олно.
14
15     int even = 0;
16     // Тэгш элементүүдийн тоог хадгалах хувьсагч зарлана.
17
18     for(int i = 0; i < n; i++){
19         // Хүснэгтийн тоо бүрээр давтана.
20         if (a[i] % 2 == 0){
21             // Тухайн тоо тэгш эсэхийг шалгана. Үлдэгдэлтэй 2-т хуваахад хариу нь 0 гарч байвал тэгш тоо гэж үзнэ.
22             even++;
23             // Тэгш тоо олдсон учир тэгш тооны хувьсагчаа нэгээр нэмэгдүүлнэ.
24         }
25     }
26
27
28     cout << "Husnegted bui tegsh elementuudiin too: " << even;
29     // Хүснэгтэд буй тэгш элементүүдийг тоог хэвлэнэ.
30
31     return 0;
32
33 }

```

```

8.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  // Лаб 1)
5  // 8. Өгөгдсөн хүснэгтийн арифметик дунжийг олдог хэрэглэгчийн функц бич.
6
7  float average(int a[], int size) {
8      float sum = 0;
9      // Хүснэгтэд байгаа тоонуудын нийлбэрийг хадгалах хувьсагч
10     for (int i = 0; i < size; i++) {
11         sum += a[i];
12     }
13     // Хүснэгтэд байгаа тоонуудын нийлбэрийг олох давталт
14
15     return sum / size;
16     // Нийлбэрийг нь нийт хэдэн тоо байгаад хуваана.
17     // Ингэснээр дундаж утга буцаагдах болно.
18 }
19
20 int main() {
21     int a[] = {2, 0, 0, 2, 1, 1, 2, 9, 11, 29};
22     // Өгөгдсөн хүснэгт
23     int size = sizeof(a) / sizeof(a[0]);
24     // Өгөгдсөн хүснэгтэд хэдэн элемент байгааг олно.
25
26     cout << "Ugugdsun husnegtiin dundaj: " << average(a, size);
27     // Хэрэглэгчийн функцаа ашиглан дундаж утгыг хэвлэнэ.
28     return 0;
29 }

```

```

9.cpp > main()
1  #include <iostream>
2
3  using namespace std;
4
5  // Лаб 1)
6  // 9. Хүснэгтийг өсөх/буурахаар эрэмбэл.
7
8  int main() {
9      int a[10] = {0, 2, 2, 8, 15, 2, 3, 4, 29, 11};
10     int n = 10;
11
12     for (int i = 0; i < n - 1; i++) {
13         // n - 1 хүртэл удаа давтах тухайн хүснэгтийн хамгийн их тоог хамгийн сүүлчийн байрлалд аваачна.
14         for (int j = 0; j < n - i - 1; j++) {
15             // Давхар давталт нь n - i - 1 удаа ажиллана. Учир нь хамгийн их тоо нь гаднах давталтад хамгийн сүүлчийн байрлалд оччихсон байгаа учир дахин тэр үйлдлийг хийх шаардлагагүй.
16             if (a[j] > a[j + 1]) {
17                 // a[j] -д байгаа утга a[j + 1] -ээс их байвал a[j], a[j + 1] 2-ийн байрыг сольж дарааллыг өөрчилнө.
18                 int temp = a[j];
19                 a[j] = a[j + 1];
20                 a[j + 1] = temp;
21             }
22         }
23     }
24
25     cout << "Усuh: " << endl;
26     for (int i = 0; i < n; i++) {
27         cout << a[i] << " ";
28     }
29
30     cout << endl;
31
32
33     for (int i = 0; i < n - 1; i++) {
34         // n - 1 хүртэл удаа давтах тухайн хүснэгтийн хамгийн бага тоог хамгийн сүүлчийн байрлалд аваачна.
35         for (int j = 0; j < n - i - 1; j++) {
36             // Давхар давталт нь n - i - 1 удаа ажиллана. Учир нь хамгийн бага тоо нь гаднах давталтад хамгийн сүүлчийн байрлалд оччихсон байгаа учир дахин тэр үйлдлийг хийх шаардлагагүй.
37             if (a[j] < a[j + 1]) {
38                 // a[j] -д байгаа утга a[j + 1] -ээс бага байвал a[j], a[j + 1] 2-ийн байрыг сольж дарааллыг өөрчилнө.
39                 int temp = a[j];
40                 a[j] = a[j + 1];
41                 a[j + 1] = temp;
42             }
43         }
44     }
45
46     cout << "Buurah: " << endl;
47     for (int i = 0; i < n; i++) {
48         cout << a[i] << " ";
49     }
50
51     return 0;

```



```

10.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  // Лаб 1)
5  // 10. Өгөгдсөн тоо анхны тоо эсэхийг шалга.
6
7  int isPrime(int n){
8      // Анхны тоо мөн эсэхийг шалгах функцийг зарлана.
9      if(n <= 1){
10         return 0;
11     }
12     // Хэрвээ өгөгдсөн тоо 1 болон түүнээс бага байвал 0-ийг буцаана.
13     // Учир нь 1 болон 0 нь анхны тоо биш.
14
15     for(int i = 2; i <= n/2; i++){
16         // Өгөгдсөн тоог хуваадаг тоонуудыг шалгана.
17         if(n % i == 0){
18             return 0;
19             // Хэрвээ өгөгдсөн тоо ямар нэгэн тоонд үлдэгдэлгүй хуваагдаж байвал анхны тоо биш.
20             // Тиймээс анхны тоо биш гэж үзэн 0-ийг буцаана.
21         }
22     }
23
24     return 1;
25     // Харин эсрэгээрээ буюу ямар нэгэн тоонд хуваагдахгүй байвал 1-ийг буцаана.
26     // Энэ нь тус тоо анхны тоо мөн гэсэн үг.
27 }
28
29 int main(){
30     int num = 11;
31     // Өгөгдсөн тоо гэсэн учир өөрөө дур мэдэн нэг тоог оноож өгөв.
32
33     int result = isPrime(num);
34
35     // Дээрх функцийн үр дүнг result хүснэгтэд хадгална.
36
37     if(result == 1){
38         //
39         cout << num << " Anhnii too" << endl;
40     }
41     else{
42         cout << num << " Anhnii too bish";
43     }
44     // Үр дүнгүүдээ хэвлэнэ.
45 }

```