

С++ хэл дээр бодлого бодож, cin болон cout объектыг хэрэглэж сурах (Лаборатори №1)

Э.Багабанди

ХШУИС, 3 – р түвшиний оюутан, dorjoo@num.edu.mn

1. ОРШИЛ/УДИРТГАЛ

С++ хэлийг ашиглан бодлого бодож, тухайн хэлтэй танилцан cin болон cout объектыг хэрэглээг ойлгох.

2. ЗОРИЛГО

Энэхүү лабораторийн ажлын хүрээнд дараах 5н бодлогыг бодож тайлбарлана. Үүнд:

1. Өгөгдсөн 3н тооны ихийг олж буцаах хэрэглэгчийн функц хэрэгжүүлэх.
2. Нэг for давталт ашиглан өгөгдсөн N ширхэг тооны хамгийн их/хамгийн багийг олох.
3. Өгөгдсөн хүснэгтийн тэгш элементүүдийн тоог олох хэрэглэгчийн функцийг хэрэгжүүлэх.
4. Өгөгдсөн хүснэгтийн сондгой элементүүдийн нийлбэр болон тоог олж, арифметик дунжийг олох хэрэглэгчийн функцийг хэрэгжүүлэх.
5. Өгөгдсөн натурал тооны оронгийн тоо болон цифрүүдийг олж, олсон цифрүүдийг дамжуулсан хүснэгтэд хадгалах хэрэглэгчийн функцийг хэрэгжүүлэх.

3. ОНОЛЫН СУДАЛГАА

3.1 cout болон cin объект /iostream толгой файлд байдаг.

- cout объект/insert буюу оруулгын оператор
 - Стандарт гаралтын урсгал/өгөгдлийн урсгалтай хамааралтай хийсвэр ойлголт/ - тай хамаатай объект.
 - Стандарт гаралтын урсгал бол стандарт гаралтын төхөөрөмж болох дэлгэж рүү гаргах мэдээллийн цуваа юм.
 - Си хэлний бит шилжүүлэх '<<' операторын үүргийг дахин тодорхойлсон бөгөөд эл оператор түүний баруун талд бичиж өгсөн мэдээллийг гаралтын урсгалд нэмж оруулах үүрэгтэй.
- cin объект/extraction буюу ялгаж авах оператор
 - '>>' нь дараалал\гарын буфер\ - т байгаа мэдээллээс унших мэдээллээ ялган авч тодорхой хувиргалт хийгээд хувьсагч руу хийх

зориулалтын оператор.

- CIN – объект нь стандарт оролтын төхөөрөмж, өөрөөр хэлбэл гарыг хаяглана.

4. ХЭРЭГЖҮҮЛЭЛТ

■ Бодлого №3

- Тухайн бодлогын хүрээнд countEven() гэх хэрэглэгчийн функц тодорхойлсон бөгөөд тухайн функц нь int төрлийн хүснэгтэн/int a[]/ болон int төрлийн/int n/ хувьсагчин хоёр параметр авдаг. Мөн int төрлийн утга буцаана. Хүснэгтэн параметрийн тэгш элементүүдийн тоог тоолохын тулд cnt гэх дотоод хувьсагч зарласан бөгөөд тухайн хувьсагч нь тоолуур хувьсагч тул int төрөлтэй, анхны утга нь 0 байх юм. Үүний дараа хүснэгтийн 0 – ээс (n - 1) хүртэлх индесктэй буюу бүх элементээр гүйх 'int i' гэх удирдлагын хувсагчтай for давталт үүсгэж, давталтын бие дотор a[i] элементийг 2 – т үлдэлгүй хуваагдаж байгаа эсэхийг шалгах бөгөөд тухайн элемент нь 2 – т үлдэлгүй хуваагдах үед cnt хувьсагчийн утгыг нэгээр нэмэгдүүлнэ. Харин тухайн элемент 2 – т үлдэгдэлтэй хуваагдах үед ямар нэг үйлдэл хийх шаардлагагүй. Мөн давталтын бие дэх бүх үйлдэл хийгдэж дууссаны дараа удирдлагын хувьсагч/i/ - ийн утга нэгээр нэмэгдэнэ. Эцэст нь cnt хувьсагчийн утгыг буцааснаар уг бодлогын зорилго нь биелэгдэх юм.

- **Туршилт:** int countEven(int a[], int n);// a[] = {34, 234, 35, 89, 4}, n = 5 байх үед

1. int cnt = 0; // анхны утга нь 0 байх int төрлийн cnt хувьсагчийг зарлаж байна.
2. for(int i = 0; i < n; i++){// i < 5 байна; i = 0; i = 1; i = 2; i = 3; i = 4; i = 5 болж давталтаас гарна.
3. if(a[i]%2 == 0) cnt++;// p[0]%2 = 34%2 = 0 тул cnt=1 болно; p[1]%2 = 234%2 = 0 тул cnt=2 болно; p[2]%2 = 35%2 = 1!=0 тул ямар ч үйлдэл хийхгүй; 89%2 = 1!=0 тул ямар ч үйлдэл хийхгүй; 4%2 = 0 тул cnt=3 болно;
4. }
5. return cnt;// cnt = 3 утгыг буцаана.

■ Бодлого №4

- Тухайн бодлогын хүрээнд avgOdd() гэх хэрэглэгчийн функц

тодорхойлсон бөгөөд тухайн функц нь `int` төрлийн хүснэгтэн/`int a[]`/ болон `int` төрлийн/`int n`/ хувьсагчин хоёр параметр авдаг. Гэхдээ `float` буюу бутархай төрлийн утга буцаана. Учир нь дундаж утга бутархай байх шаардлагатай. Хүснэгтэн параметрийн сондгой элементүүдийн тоог тоолохын тулд `cnt` гэх дотоод хувьсагч зарласан бөгөөд тухайн хувьсагч нь тоолуур хувьсагч тул `int` төрөлтэй, анхны утга нь 0 байх юм. Мөн сондгой элементүүдийг нийлбэрийг хадгалах `float` төрлийн анхны утга нь 0 байх `sum` хувьсагч зарласан ба тухайн хувьсагчийг бутархай төрлөөр зарласнаар `sum/cnt = float/int = float` утгатай болох юм. Мөн хүснэгтийн сондгой хувьсагчдын нийлбэрийг `sum` – д хадгалж авахын тулд анхны утгыг 0 гэж зарласан болно. Үүний дараа хүснэгтийн 0 – ээс (`n - 1`) хүртэлх индесктэй буюу бүх элементээр гүйх '`int i`' гэх удирдлагын хувсагчтай `for` давталт үүсгэж, давталтын бие дотор `a[i]` элементийг 2 – т үлдэлгүй хуваагдаж байгаа эсэхийг шалгах бөгөөд тухайн элемент нь 2 – т үлдэлтэй хуваагдах үед `sum` хувьсагчид `a[i]` нийлбэрийг хадгалж, `cnt` хувьсагчийн утгыг нэгээр нэмэгдүүлнэ. Мөн давталтын бие дэх бүх үйлдэл хийгдэж дууссаны дараа удирдлагын хувьсагч/`i`/ - ийн утга нэгээр нэмэгдэнэ. Харин тухайн элемент 2 – т үлдэгдэлтэй хуваагдах үед ямар нэг үйлдэл хийх шаардлагагүй. Эцэст нь `cnt` хувьсагчийн утгыг буцааснаар уг бодлогын зорилго нь биелэгдэх юм.

- **Туршилт:** `float avgOdd(int a[], int n)`// `a[] = {34, 234, 35, 89, 4}`, `n = 5` байх үед

1. `int cnt = 0;` // `int` төрлийн гарааны утга нь 0 байх `cnt` хувьсагч зарлав. Тоолуур хувьсагч
2. `float sum = 0;` // `float` төрлийн гарааны утга нь 0 байх `sum` хувьсагч зарлав. Нийлбэр хадгалах хувьсагч
3. `for(int i = 0; i < n; i++){` // `i < 5` байна; `i = 0`; `i = 1`; `i = 2`; `i = 3`; `i = 4`; `i = 5` болж давталтаас гарна.
4. `if(a[i]%2 == 1){` // `a[0]%2 = 34%2=0==1` худал тул ямарч үйлдэл хийхгүй; `234%2=0==1` худал тул ямарч үйлдэл хийхгүй; `a[2]%2 = 35%2=1==1` үнэн тул `if` дотор орно; `a[3]%2 = 89%2=1==1` үнэн тул `if` дотор орно; `a[4]%2 = 4%2=0==1` худал тул ямарч үйлдэл хийхгүй;
5. `sum += a[i];` // `sum = 0 + a[2] = 35` болно; `sum = 35 + a[3] = 35 + 89 = 124` болно;
6. `cnt++;` // `cnt = cnt + 1 = 0 + 1 = 1` болно; `cnt = cnt + 1 = 1`

+ 1 = 2 болно;

7. }

8. }

9. return sum/cnt; // sum = 124.0 болон cnt = 2 гэдгээс sum/cnt = 124.0/2 = 62.0 утгыг буцаана. Тухайн буцаж буй утга бутархай учир sum/cnt = float/int=float байдаг.

■ Бодлого №5

- Уг бодлогын хүрээнд otoo() гэх хэрэглэгчийн функц тодорхойлсон бөгөөд тухайн функц нь int төрлийн хүснэгтэн/int ci[]/ болон int төрлийн хувьсагчин/int a/ хоёр параметр авдаг. Мөн int төрлийн утга буцаадаг. Хүснэгт 'ci[]' нь main() функц дотор зарлагдах бөгөөд otoo() функцид параметрээр ирэхдээ хоосон байх юм. Харин int төрлийн a параметр нь гараас өгөгдсөн тоо бөгөөд уг функц нь тухайн тооны оронгийн тоог буцаах ба дамжиж ирсэн хүснэгтэн параметрт түүний цифрүүдийг хадгалана. Тухайн тооны оронгийн тоог хадгалах, анхны утга нь 0 байх от хувьсагч зарласан. Үүний дараа a параметрийн утга нь 0 – ээс их (a>0) гэсэн нөхцөлтэй while давталт үүсгэсэн бөгөөд давталтын бие дотор a тооны сүүлийн цифрийг тасалж авсан буюу 10 – т үлдэгдэлтэй хуваахад гарсан утгыг(a%10) хадгалахын тулд c гэх int хувьсагч зарласан. Мөн тухайн утгыг ci хүснэгтийн от дугаар элемент болгон хадгалж авсан бөгөөд ингэснээр ci хүснэгт a тооны бүх цифрүүдийг хадгалж авах юм. Давталтын биеийн төгсгөлд a тоог 10 – т бүхлээр хувааж, өөрт нь тухайн утгыг хадгалсан(a=a/10). Энэ үйлдлийн үр дүнд a тооны цифрүүд араасаа нэг нэгээр хасагдах бөгөөд мөн оронгийн тоо нь нэг нэгээр буурах юм. Иймд уг бодлогын зорилго биелэгдэж, otoo() функцээс оронгийн тоо буцаж ирнэ. Харин дамжуулж өгсөн хүснэгтэд цифрүүд хадгалагдсан байх юм.

- **Туршилт:** int otoo(int a, int ci[])// a = 3456 байх үед

1. int ot = 0; // int төрлийн гарааны утга нь 0 байх cnt хувьсагч зарлав. Тоолуур хувьсагч
2. while(a > 0){// 3456 > 0 үнэн тул давталт дотор орно; 345 > 0 үнэн тул давталт дотор орно; 34 > 0 үнэн тул давталт дотор орно; 3 > 0 үнэн тул давталт дотор орно; 0 > 0 худал тул давталтаас гарна;
3. int c = a%10;// c = a%10 = 3456%10 = 6 болно; c = a%10 =

5 болно; $c = a \% 10 = 4$ болно; $c = a \% 10 = 3$ болно;

4. $ci[ot++] = c;$ // $c[0] = 6$ болно; $c[1] = 5$ болно; $c[2] = 4$ болно; $c[3] = 3$ болно;
5. $a /= 10;$ // $a = a/10 = 3456/10 = 345$ болно; $a = a/10 = 345/10 = 34$ болно; $a = a/10 = 34/10 = 3$ болно; $a = a/10 = 3/10 = 0$ болно;
6. }
7. $return ot;$ // $ot = ot++ = 4$ буюу а тооны оронгийн тоог буцаана. Харин $ci[]$ хийсвэр параметрт дамжин ирсэн хүснэгтэд цифрүүд хадлагдсан байна.

5. ДҮГНЭЛТ

Энэхүү лабораторын ажлын бодлогууд дээр хүснэгт, заагч, хэрэглэгчийн функц гэх ойлголтуудыг ашиглан шийдвэрлэсэн бөгөөд 4, 5 - аар бодлогод хэрэглэгчийн функцийн параметрт хүснэгтэн хувьсагчийн утгыг оноож өгөх дөө зөвхөн нэрийг дамжуулсан. Учир нь хүснэгтийг функц руу ингэж дамжуулах үед тухайн хүснэгтийн бүх утга уг функцэд очдоггүй бөгөөд хамгийн эхний элементийн хаяг нь очдог. Өөрөөр хэлбэл заагч ашиглан хүснэгтийг функцүүдэд дамжуулсан. Харин функц тухайн хүснэгтийн хамгийн эхний элементийн хаягийг ашиглаж, бусад элементүүдтэй харьцах боломжтой болдог учир нь хүснэгтийн элементүүд санах ойд дарааллан байрладаг. Иймд тухайн хэрэглэгчийн функцүүдэд дамжуулсан хүснэгтэн параметрүүдийн элементүүдэдтэй чөлөөтэй харилцах юм.

6. АШИГЛАСАН МАТЕРИАЛ

1. Объект хандлагат технологийн C++ програмчлал, Ж.Пүрэв, 2008, Улаанбаатар.

7. ХАВСРАЛТ

Кодыг энд оруулна.

Бодлого №1:

1. `/*1. a, b, c гурван тооны хамгийн ихийг олох хэрэглэгчийн функц бич.*`
2. `#include<iostream>`
3. `using namespace std;`
4. `int max3num(int, int, int);`
5. `main(){`
6. `int a, b, c;`
7. `cout << "Enter 3 number: ";`

```

8.    cin >> a >> b >> c;
9.    cout << "Max number in given numbers: " << max3num(a, b, c);
10.   return 0;
11. }
12.
13. int max3num(int a, int b, int c){
14.     int mx;
15.     if(a<b) mx = b;
16.     else mx = a;
17.     if(mx < c) mx = c;
18.     return mx;
19. }

```

Бодлого №2:

```

1.  /*2. Өгөгдсөн N ширхэг тооны хамгийн их/хамгийн багийг олох код бич.*/
2.  #include<iostream>
3.  using namespace std;
4.  main(){
5.      int n;
6.      cout << "Enter a number: ";
7.      cin >> n;
8.      int m, mx, mn;
9.      cout << "Enter " << n << " numbers: ";
10.     cin >> m;
11.     mx = m;
12.     mn = m;
13.     for(int i = 1; i < n; i++){
14.         cin >> m;
15.         if(mn > m) mn = m;

```

```

16.     if(mx < m) mx = m;
17. }
18.     cout << "Max number in given numbers: " << mx << endl;
19.     cout << "Min number in given numbers: " << mn << endl;
20.     return 0;
21. }

```

Бодлого №3:

```

1.  /*3. Өгөгдсөн хүснэгтийн тэгш элементүүдийн тоог олох хэрэглэгчийн функц бич.*/
2.  #include<iostream>
3.  using namespace std;
4.  int countEven(int a[], int n);
5.  main(){
6.      int n;
7.      cout << "Enter s size in Array: ";
8.      cin >> n;
9.      int a[n];
10.     cout << "Enter elements in Array: " << endl;
11.     for(int i = 0; i < n; i++){
12.         cin >> a[i];
13.     }
14.     cout << "Even numbers count: " << countEven(a, n) << endl;
15.     return 0;
16. }
17.
18. int countEven(int a[], int n){
19.     int cnt = 0;
20.     for(int i = 0; i < n; i++){
21.         if(a[i]%2 == 0) cnt++;

```

```
22.    }  
23.    return cnt;  
24. }
```

Бодлого №4:

```
1. /*4. Өгөгдсөн хүснэгтийн сондгой элементүүдийн арифметик дунжийг олох  
хэрэглэгчийн функц бич.*/  
2. #include<iostream>  
3. using namespace std;  
4. float avgOdd(int a[], int n);  
5. main(){  
6.     int n;  
7.     cout << "Enter s size in Array: ";  
8.     cin >> n;  
9.     int a[n];  
10.    cout << "Enter elements in Array: " << endl;  
11.    for(int i = 0; i < n; i++){  
12.        cin >> a[i];  
13.    }  
14.    cout << "Odd numbers avg: " << avgOdd(a, n) << endl;  
15.    return 0;  
16. }  
17.  
18. float avgOdd(int a[], int n){  
19.     int cnt = 0;  
20.     float sum = 0;  
21.     for(int i = 0; i < n; i++){  
22.         if(a[i]%2 == 1){  
23.             sum += a[i];  
24.             cnt++;
```



```

25.     }
26. }
27. return sum/cnt;
28. }

```

Бодлого №5:

```

1. /* 5. Өгөгдсөн натурал тооны оронгийн тоо болон цифрүүдийг олох хэрэглэгчийн
   функц бич.*/
2. #include<iostream>
3. using namespace std;
4. int otoo(int a, int ci[]);
5. main(){
6.     int n;
7.     cout << "Enter a number: ";
8.     cin >> n;
9.     int ciqr[100];
10.    int c = otoo(n, ciqr);
11.    cout << "Orongiin too: " << c << endl;
12.    cout << "Ciquud: " << endl;
13.    for(int i = 0; i < c; i++)
14.        cout << ciqr[i] << " ";
15.    return 0;
16. }
17.
18. int otoo(int a, int ci[]){
19.     int ot = 0;
20.     while(a > 0){
21.         int c = a%10;
22.         ci[ot++] = c;
23.         a /= 10;

```

```
24.  }  
25.  return ot;  
26. }
```