

Лабораторын ажил

(Лаборатори No2)

Ц. Намсрайжамц

МУИС, ХШУИС, МКУТ, Программ хангамжийн хөтөлбөрийн 3-р түвшиний оюутан

[18B1NUM1671@stud.num.edu.mn](mailto:18B1NUM1671@stud.num.edu.mn)

**1. Санах ойн хаяг гэж юу вэ? Хаяган хувьсагч гэж юу вэ? Хаяган хувьсагчийн хэмжээ хэдэн байт байдаг вэ?**

- а) Санах ойн хаяг гэдэг нь
- б) Хувьсагчийн хаягийг хадгалах хувьсагчийг хаяган хувьсагч гэнэ.
- с) Хаягийг нь зааж буй хувьсагч ямар гэдгээс үл хамааран хаяган хувьсагчийн хэмжээ 8 байт байна.(хэрэв 32 bit computer бол 4 байт байна.)

2.

```
char *p1;  
  
int *p2;  
  
double *p3;  
  
cout<<sizeof(p1)<<sizeof(p2)<<sizeof(p3);
```

Энэ код нь char, int , double гэсэн 3 төрлийн хаяган хувьсагчийн санах ойд эзлэх зайн хэмжээг харуулж байна.

3.

```
int a=125; // int төрөлтэй а хувьсагчид 125 гэдэг утга оноож байна  
  
int *p = &a; // p заагч хувьсагчид а хувьсагчийн хаягийг оноож байна  
  
cout<<p<<"\n"; // p хувьсагчийн зааж буй хаягийн утгыг хэвлэж байна  
  
cout<<*p<<"\n"; // p хувьсагчийн зааж буй хаягийг хэвлэж байна  
  
p++; // p хаяган хувьсагчийн утгыг 1 ээр нэмэж байна  
  
cout<<p<<"\n"; // p хувьсагчийн одоогийн зааж буй хаягийг хэвлэж байна  
  
cout<<*p<<"\n"; // p хувьсагчийн одоогийн зааж буй хаяг дээрх утгыг хэвлэж байна.
```

```
0x7ffee611b94c  
125  
0x7ffee611b950  
-435046040
```

4.

```
int numbers[5]; // numbers хүснэгтийг зарлаж байна

int * p; // хаяган хувьсагч зарлаж байна

p = numbers; *p = 10; // p хувьсагчаар дамжуулан numbers ийн эхний утгын 10 болгож байна

p++; *p = 20; // p хувьсагчийг дараагийн хягийг заалган тэр хаягт 20 гэдэг утга оноож байна.

p = &numbers[2]; *p = 30; // p д numbers ийн 2 дахь index ийн хаягийг дамжуулан тэр хаягт 30 утга оноож байна

p = numbers + 3; *p = 40; // p ийн зааж 4 дэх хаягийг заан 40 утга авч байна

p = numbers; *(p+4) = 50; // p ийн зааж буй хаягийг 4 ээр ихэсгэн тэр хаягт 50 утга олгож байна

for (int n=0; n<5; n++)

    cout << numbers[n] << ", "; // хэвлэх
```

5. Хаяган хувьсагч ашиглан (функцин параметер нь хаяган хувьсагч байна) хоёр хувьсагчийн утгыг солих хэрэглэгчийн функц бич.

```
lab2_5.cpp ×
Users > macbook > Downloads > lab2_5.cpp > main()
1  #include <iostream>
2  using namespace std;
3  void changeValue( int *p1, int *p2){
4      int x , y;
5      y = *p2;
6      *p2 = *p1;
7      *p1 = y;
8  }
9  int main()
10 {
11     int a = 5, b = 3;
12     int *ptr1 = &a;
13     int *ptr2 = &b;
14     cout<<a<<" , " << b<< "\n";
15     changeValue(ptr1, ptr2);
16     cout<<a<<" , " << b<< "\n";
17 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Macbooks-MacBook-Pro:Downloads macbook$ g++ lab2_5.cpp
Macbooks-MacBook-Pro:Downloads macbook$ ./a.out
5 , 3
3 , 5
Macbooks-MacBook-Pro:Downloads macbook$
```

## 6. Заалтан хувьсагч гэж юу вэ? Ямар давуу талтай вэ?

Заалтан хувьсагч нь тодорхой нэг нэртэй хувьсагчид дахин өөр нэр өгөх (нэг санах ой хоёр өөр нэртэй байх). Заалтан хувьсагчийг тодорхой нэр бүхий хувьсагчтай холбож үүсгэх ба түүгээр дамжуулж утга өөрчлөх боломжтой.

Функцийн параметрээр хэрэглэхэд тохиромжтой. Ингэснээр санах ой хэмнэнэ.

7. Заалтан параметр (функцин параметер нь заалт байна) ашиглан хоёр хувьсагчийн байрыг солих хэрэглэгчийн функц бич.

```
lab2_5.cpp lab2_7.cpp X
Users > macbook > Downloads > lab2_7.cpp > changeValue
1  #include <iostream>
2  using namespace std;
3  void changeValue( int &, int &);
4  int main()
5  {
6      int a = 431, b = 123;
7      cout<<a<<" , " << b<< "\n";
8      changeValue(a, b);
9      cout<<a<<" , " << b<< "\n";
10 }
11 void changeValue( int &a_ref , int &b_ref){
12     int k;
13     k = a_ref;
14     a_ref = b_ref;
15     b_ref = k;
16 }
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Macbooks-MacBook-Pro:Downloads macbook$ ./a.out
431 ,123
123 ,431
Macbooks-MacBook-Pro:Downloads macbook$
```

**8. Дотоод мөр функц гэж юу вэ? Яагаад үүнийг хэрэглэх шаардлагатай байдаг вэ?**

Дотоод функц ашигласнаар түүний командуудыг функц хэрэглэсэн командын мөр бүрд хувилж үүсгэнэ. Хэрэглэгчийн функцийг дотоод мөр функц болгоход inline үгийг ашиглана. Энэ нь эмх цэгцтэй код болоход болон функцийг дахин дуудаж ажилуулахад чухал ач холбогдолтой.

Ач холбогдол нь програмын хэмжээ ихсэх ч гэсэн хурд нэмэгдэнэ.

**9. Ойн цоорхой гэж юу вэ? Ямар тохиолдолд үүсдэг вэ? Жишээгээр тайлбарла.**

Malloc ашиглан үүсгэсэн санах ойн хаяган хувьсагчийг орхисон, хандах боломжгүй ой үүсэхийг ойн цоорхой гэнэ. Malloc оор үүсгэсэн санах ойг чөлөөлж заавал delete болон, free гээр чөлөөлөх нь зүйтэй.