

CS200 – Програмчлалын үндэс

Лекц 08

Хэвшүүлсэн оролт/гаралт

Профессор А.Эрдэнэбаатар

Лекцийн агуулга

- Хэвшүүлсэн оролт/гаралт

Урсгал

- Урсгал
 - Мөр мөрөөр зохион байгуулагдсан тэмдэгтүүдийн цуваа
 - Мөр бүр 0 буюу түүнээс их тэмдэгтийг агуулах ба шинэ мөрийн тэмдэгтээр төгсөнө
 - ANSI C хэл 254 хүртэл тэмдэгттэй мөрийг дэмждэг
 - Бүх оролт, гаралтыг гүйцэтгэнэ
 - Урсгал голдуу тогтсон чиглэлтэй байдаг
 - Стандарт оролт – гар
 - Стандарт гаралт – дэлгэц
 - Стандарт алдаа – дэлгэц

printf ашигласан хэвшүүлсэн гаралт

- **printf**

- Гаралтыг нарийн хэвшүүлэх
 - Хувиргалтын тодорхойлолт: флаг, талбарын өргөн, нарийвчлал г.м.
- Нарийвчлах, баганаар тэгшитгэх, баруун/зүүн жигдрүүлэлт, тэмдэгт оруулах, илтгэгч болон 16 –тын формат, тогтмол өргөн ба нарийвчлалыг хийж болно

- Хэвшүүлэлт

- **printf (хэвшүүлэлт_удирдах_мөр, бусад_аргумент) ;**
- Хэвшүүлэлтийг удирдах мөр: гаралтын хэвийг заана
- Бусад аргумент: хэвшүүлэлтийг удирдах мөрөнд орсон хувиргалтын тодорхойлолт бүрт харгалзана
 - Тодорхойлолт бүр % тэмдгээр эхэлж хувиргалт тодорхойлогчоор төгсөнө

Бүхлийг хэвлэх

- Бүхэл
 - Бүхэл тоо(аравтын цэггүй): 25, 0, -9
 - Эерэг, сөрөг, эсвэл 0(тэг)
 - Өгөгдмөлөөр зөвхөн хасах тэмдэг л хэвлэгдэнэ (үүнийг яаж өөрчилж болохыг удахгүй үзнэ)

Бүхэл хувиргалт тодорхойлогч

Хувиргалт тодорхойлогч	Тайлбар
d	Тэмдэгтэй 10 –тын бүхэл байдлаар гаргана
i	Тэмдэгтэй 10 –тын бүхэл байдлаар гаргана(I болон d тодорхойлогч нь scanf –д ялгаатай байдгийг санаарай)
o	Тэмдэггүй 8 –тын бүхэл байдлаар гаргана
u	Тэмдэггүй 10 –тын бүхэл байдлаар гаргана
x эсхүл X	Тэмдэггүй 16 –тын бүхэл байдлаар гаргана. Тодорхойлогч том үсэгтэй бол 16 –тын үсгэн цифрүүд томоор(0-9, A-F), жижиг бол жижгээр(0-9, a-f) гарна
h эсхүл l	Бүхэл хувиргалтын тодорхойлогчийн өмнө бичигдэнэ. Ингэснээрээ short эсхүл long бүхэл болохыг заана. Эдгээрийг уртыг тодорхойлогч гэж нарийвчлан нэрлэж болно

Бүхэл хувиргалт тодорхойлогч

```
1. /* Ex_74 Using the integer conversion specifiers */
2. #include <stdio.h>
3.
4. int main( void )
5. {
6.     printf( "%d\n", 455 );
7.     printf( "%i\n", 455 ); /* i нь short тоог заана */
8.     printf( "%d\n", +455 ); /* l нь long тоог заана */
9.     printf( "%d\n", -455 );
10.    printf( "%hd\n", 32000 ); /* o нь 8-тын бүхлийг заана */
11.    printf( "%ld\n", 2000000000L ); /* L suffix makes literal a long */
12.    printf( "%o\n", 455 );
13.    printf( "%u\n", 455 ); /* u нь unsigned бүхлийг заана */
```

d, i тэмдэгтэй бүхлийг заана

h нь **short** тоог заана

l нь **long** тоог заана

o нь 8-тын бүхлийг заана

u нь **unsigned** бүхлийг заана

Бүхэл хувиргалт тодорхойлогч

```
14. printf( "%u\n", -455 );
15. printf( "%x\n", 455 );
16. printf( "%X\n", 455 );
17.
18. return 0; /* indicates successful termination */
19.
20. } /* end main */
455
455
455
-455
32000
2000000000
707
455
```

x, X 16 -тын бүхлийг заана

4294966841
1c7
1c7

Санамж: -455 гэдэг утгыг %u (14-р мөр) уншихдаа тэмдэггүй (unsigned) 4294966841 гэсэн тоо болгож хувиргана. Өөрөөр хэлвэл сөрөг тоо хэвлэхэд хувиргалтын тодорхойлогч unsined утга болгон алдаа гаргадаг

Хөвөгч-таслалтай тоог хэвлэх

- Хөвөгч таслалтай тоо
 - Аравтын цэгтэй байна (33.5)
 - Илтгэгчтэй тэмдэглэгээ (шинжлэх ухааны тэмдэглэгээний компьютерийн хувилбар)
 - 150.3 бол 1.503×10^2 – шинжлэх ухаанаар
 - 150.3 бол $1.503E+02$ – илтгэгчтэйгээр (E - экспонент)
 - e эсхүл E –г ашиглана
 - **f** – хөвөгч таслалтай тоог аравтын таслал/цэгээс өмнө ядаж нэг оронтой хэвлэх
 - **g** эсвэл **G** – **f** болон **e** –д сүүл 0 –гүй хэвлэх (1.2300 нь 1.23 болно)
 - Илтгэгчтэй хэлбэрийг экспонент нь -4 –өөс багагүй, 6 –аас ихгүй тохиолдолд ашиглах нь тохиромжтой (өгөгдмөлөөр 6 цифр орно гэсэн үг)

Хөвөгч-таслалтай хувиргалт тодорхойлогч

Хувиргалт тодорхойлогч	Тайлбар
e эсхүл E	Хөвөгч-таслалтай утгыг илтгэгчтэй тэмдэглэгээгээр гаргана
f	Хөвөгч-таслалтай утгыг тогтмол-таслалтай тэмдэглэгээгээр гаргана
g эсхүл G	Хөвөгч-таслалтай утгыг түүний утгаас хамаарч хөвөгч-таслалтай (f), эсхүл илтгэгчтэй (e , E) хэлбэрийн аль нэгээр гаргана
L	Хөвөгч-таслалтай хувиргалт тодорхойлогчийн өмнө орсноор гарах утга long double гэдгийг заана

Хөвөгч-таслалтай хувиргалт тодорхойлогч

```
1. /* Ex_75 Printing floating-point numbers with
2.    floating-point conversion specifiers */
3.
4. #include <stdio.h>
5.
6. int main( void )
7. {
8.     printf( "%e\n", 1234567.89 );
9.     printf( "%e\n", +1234567.89 );
10.    printf( "%e\n", -1234567.89 );
11.    printf( "%E\n", 1234567.89 );
12.    printf( "%f\n", 1234567.89 );
13.    printf( "%g\n", 1234567.89 );
14.    printf( "%G\n", 1234567.89 );
15.
16.    return 0; /* indicates successful termination */
17.
18.} /* end main */
1.234568e+006
1.234568e+006
-1.234568e+006
1.234568E+006
1.234567 0.000000
```

e, E нь илтгэгчтэй тэмдэглэгээг заана

f нь тогтмол-таслалтай тэмдэглэгээг заана

g, G нь тооны хэмжээнээс хамаарч e болон f тэмдэглэгээуна аль нэг нь гэдгийг заана

Мөр ба тэмдэгтийг хэвлэх

- **c**
 - **char** төрлийн аргументийг хэвлэнэ
 - Мөрийн эхний тэмдэгтийг хэвлэхэд хэрэглэж болохгүй
- **s**
 - аргумент нь **char** төрлийн заагч байна
 - **NULL** (`'\0'`) тэмдэгт тааралдтал тэмдэгтийг хэвлэнэ
 - **char** төрлийн аргументийг хэвлэж чадахгүй
- Тогтоох зүйлс
 - Тэмдэгт тогтмолыг дан хашилтанд бичнэ (`'z'`)
 - Мөрийг давхар хашилтанд хийнэ (`"z"` – энэ нь үнэндээ `'z'` , `'\0'` гэсэн хоёр тэмдэгт юм)

Мөр ба тэмдэгтийг хэвлэх

```
1. /* Ex_76 Printing strings and characters */
2. #include <stdio.h>
3.
4. int main( void )
5. {
6.     char character = 'A'; /* initialize char */
7.     char string[] = "This is a string"; /* initialize char array */
8.     const char *stringPtr = "This is also a string";
9.
10.    printf( "%c\n", character );
11.    printf( "%s\n", "This is a string" );
12.    printf( "%s\n", string );
13.    printf( "%s\n", stringPtr );
14.
15.    return 0; /* indicates successful termination */
16.
17.} /* end main */
```

с нь тэмдэгт хэвлэхийг заана

s нь мөр хэвлэхийг заана

A

This is a string

This is a string

This is also a string

Бусад хувиргалт тодорхойлогч

- **p**

- Заагчийн утга буюу хаягийг системийн шийдэлд захирагдаж гаргана (ихэнх систем 16 –тын тэмдэглэгээг ашигладаг)

- **n**

- Тухайн **printf** –ээр хэдэн тэмдэгт гарсныг хадгалдаг
- Бүхэл төрлийн заагчийг аргумент болгож авдаг
- **%n** тодорхойлогчоор юу ч хэвлэгдэхгүй
- **printf** бүр утга буцаадаг (гарсан тэмдэгтийн тоо; сөрөг бол алдаа)

- **%**

- Процентийн тэмдгийг хэвлэнэ - **%%**

Бусад хувиргалт тодорхойлогч

```
1. /* Ex_77 Using the p, n, and % conversion specifiers */
2. #include <stdio.h>
3.
4. int main( void )
5. {
6.     int *ptr;      /* define pointer to int */
7.     int x = 12345; /* initialize int x */
8.     int y;         /* define int y */
9.
10.    ptr = &x;       /* assign address of x to ptr */
11.    printf( "The value of ptr is %p\n", ptr );
12.    printf( "The address of x is %p\n\n", &x );
13.
14.    printf( "Total characters printed on this line:%n", " " );
15.    printf( " %d\n\n", y );
16.
17.    y = printf( "This line has 28 characters\n" );
18.    printf( "%d characters were printed\n\n", y );
19.
20.    printf( "Printing a %% in a format control string\n" );
```

p нь санах ойн хаягийг
хэвлэхийг заана

n нь мөрөнд хэдэн тэмдэгт
хэвлэснийг хадгална

p нь процентийн тэмдгийг хэвлэнэ

Бусад хувиргалт тодорхойлогч

```
21.  
22.     return 0; /* indicates successful termination */  
23.  
24.} /* end main */
```

The value of ptr is 0012FF78

The address of x is 0012FF78

Total characters printed on this line: 38

This line has 28 characters

28 characters were printed

Printing a % in a format control string

Талбарын өргөн болон нарийвлал зааж хэвлэх

- Тайлбарын өргөн
 - Өгөгдлийг хэвлэх талбарын хэмжээ
 - Хэрэв өгөгдлөөсөө их бол өгөгдмөлөөр **баруун жигдрүүлэлт** хийнэ
 - Хэрэв бага бол өгөгөдөл багтахаар болгож ихэсгэнэ
 - Хасах тэмдэг нэг тэмдэгтийн байр эзлэнэ
 - Өргөнийг заасан бүхэл % болон хувиргалт тодорхойлогчийн дунд бичигдэнэ
 - %4d – дөрвөн тэмдэгтийн өргөнтэй талбар

Талбарын өргөн болон нарийвлал зааж хэвлэх

```
1. /* Ex_78 Printing integers right-justified */
```

```
2. #include <stdio.h>
```

```
3.
```

```
4. int main( void )
```

```
5. {
```

```
6.     printf( "%4d\n", 1 );
```

```
7.     printf( "%4d\n", 12 );
```

```
8.     printf( "%4d\n", 123 );
```

```
9.     printf( "%4d\n", 1234 );
```

```
10.    printf( "%4d\n\n", 12345 ); /* data too large */
```

```
11.
```

```
12.    printf( "%4d\n", -1 );
```

```
13.    printf( "%4d\n", -12 );
```

Талбарын өргөн 4. Иймд 4 тэмдэгтийн
зайнд тоог хэвлэх гэж оролдоно

Хэвлэхэд хасах тэмдгийг анхаарна

Талбарын өргөн болон нарийвлал зааж хэвлэх...

```
1. printf( "%4d\n", -123 );  
2.     printf( "%4d\n", -1234 ); /* data too large */  
3.     printf( "%4d\n", -12345 ); /* data too large */  
4.  
5.     return 0; /* indicates successful termination */  
6.  
7. } /* end main */
```

```
1  
12  
123  
1234  
12345  
  
-1  
-12  
-123  
-1234  
-12345
```

Талбарын өргөн болон нарийвлал зааж хэвлэх

- Нарийвчлал
 - Утга нь өгөгдлийн төрлөөс хамаарна
 - Бүхэл тоо
 - Хэвлэх минимум цифрийн тоо. Хэрвээ өгөгдөл нь бага бол өмнө нь 0 –д бичигдэнэ
 - Хөвөгч таслалтай тоо
 - Аравтын цэгийн ард орох цифрийн тоо (e, f). g –ийн хувьд зайлшгүй байх максимум цифрийн тоо
 - Анхны тооны бутархайн ороноос бага бол утгыг тоймлоно
 - Мөр
 - Мөрөөс гаргах максимум тэмдэгтийн тоо
 - Загвар
 - Цэгийн ард нарийвлалын оронгийн тоог бичнэ % . 3f

Талбарын өргөн болон нарийвлал зааж хэвлэх

- Талбарын өргөн ба нарийвчлал
 - Аль алиныг нь зааж болно: *%өргөн . нарийвчлал*
 - %5.3f
 - Талбарын өргөн сөрөг утгатай – зүүн жигдрүүлэлт
 - Талбарын өргөн эерэг утгатай– баруун жигдрүүлэлт
 - Нарийвчлал бол эерэг байна
 - Талбарын өргөн, нарийвчлалд бүхэл илэрхийллийг ашиглаж болно
 - Талбарын өргөн, нарийвлалын байранд од (*) –г бичиж болно. Тэгвэл аргументийн жагсаалтанд харгалзах бүхэл аргумент байна
 - Жишээ: `printf("%*.*f", 7, 2, 98.736) ;`

Талбарын өргөн болон нарийвлал зааж хэвлэх

```
1. /* Ex_79 Using precision while printing integers,  
2.    floating-point numbers, and strings */  
3. #include <stdio.h>  
4.  
5. int main( void )  
6. {  
7.     int i = 873;           /* initialize int i */  
8.     double f = 123.94536;  /* initialize double f */  
9.     char s[] = "Happy Birthday"; /* initialize  
10.  
11.     printf( "Using precision for integers\n" );  
12.     printf( "\t%.4d\n\t%.9d\n\n", i, i );  
13.  
14.     printf( "Using precision for floating-point  
15.     printf( "Using precision for floating-point numbers\n", f, f, s );
```

Бүхэл тооны хувьд нарийвчлал нь минимум хэвлэх тэмдэгтийн тоог заадаг

f болон e –н хувьд нарийвлал нь аравтын цэгийн ард байх цифрийг заадаг

g –н хувьд нарийвчлал нь максимум шаардлагатай цифрийг заадаг

Талбарын өргөн болон нарийвлал зааж хэвлэх

```
16.  
17. printf( "Using precision for strings\n" );  
18. printf( "\t%.11s\n", s );  
19.  
20. return 0; /* indicates successful termination */  
21.  
22. } /* end main */
```

Мөрийн хувьд нарийвчлал нь максимум хэвлэх тэмдэгтийн тоог заадаг

Using precision for integers

000000873

Using precision for floating-point numbers

123.945

1.239e+002

124

Using precision for string

Happy Birth

printf –н хэвшүүлэлт удирдах мөрөнд флаг ашиглах

- Флаг
 - Хэвшүүлэлтийн нэмэлт боломж
 - Процентийн тэмдгийн яг ард бичнэ
 - Хэдэн флагийг хослуулж болно

Флаг	Тайлбар
- (хасах тэмдэг)	Талбартаа зүүн жигдрүүлэлт хийх
+ (нэмэх тэмдэг)	Эерэг, сөрөг тооны өмнө нэмэх, хасах тэмдэг тавих
зай	+ флаггүй эерэг тооны өмнө зай авах
#	0 угтварыг 8 –тын тооны өмнө, 0x эсхүл 0X угтварыг 16 –тын тооны өмнө гаргана e, E, f, g, G хувиргалттай бутархай хэсэггүй, хөвөгч таслалтай тооны хувьд аравтын цэгийг хүчээр тавьна
0 (тэг)	Талбарын өмнөх дутууг 0 –ээр дүүргэнэ

printf –н хэвшүүлэлт удирдах мөрөнд флаг ашиглах

```
1. /* Ex_80 Right justifying and left justifying values */
2. #include <stdio.h>
3.
4. int main( void )
5. {
6.     printf( "%10s%10d%10c%10f\n", "hello", 7, 'a', 1.23 );
7.     printf( "%-10s%-10d%-10c%-10f\n", "hello", 7, 'a',
8.
9.     return 0; /* indicates successful termination */
10.
11.} /* end main */
```

– флаг талбар дахь тэмтэгтийг зүүн жигдрүүлнэ

```
      hello          7          a  1.230000
hello      7          a          1.230000
```

printf –н хэвшүүлэлт удирдах мөрөнд флаг ашиглах

```
1. /* Ex_81 Printing numbers with and without the + flag */
2. #include <stdio.h>
3.
4. int main( void )
5. {
6.     printf( "%d\n%d\n", 786, -786 );
7.     printf( "%+d\n%+d\n", 786, -786 );
8.
9.     return 0; /* indicates successful termination */
10.
11.} /* end main */
```

+ флаг эерэг тооны өмнө хүчээр
нэмэх тэмдэг тавьна

```
786
-786
+786
-786
```

Escape цувааг хэвлэх

- Escape цувааг хэвлэх
 - Ихэнх тэмдэгтийг хэвлэж болно
 - Анхаарлын тэмдэг зэрэгт “асуудал” гардаг
 - Тэдгээрийг escape цуваагаар дүрслэх шаардлагатай
 - Буруу ташуу (\) –гийн ард escape тэмдэгтээ бичнэ

Escape цуваа

Escape цуваа	Тайлбар
\' (дан хашилт)	Дан хашилт тэмдэгтийг гаргах
\" (давхар хашилт)	Давхар хашилт тэмдэгтийг гаргах
\? (асуултын тэмдэг)	Асуултын тэмдэгтийг гаргах
\\ (буруу ташуу)	Буруу ташуу тэмдэгтийг гаргах
\a (дохио)	Аудио дохио өгөх
\b (буцаах)	Тухайн мөрөндөө нэг байршил буцах
\f (шинэ хуудас)	Шинэ хуудас руу шилжих
\n (шинэ мөр)	Шинэ мөр рүү шилжих
\r (мөрөндөө буцах)	Тухайн мөрийн эхэнд ирэх
\t (хэвтээ таб)	Дараачийн хэвтээ таб-д очих
\v (босоо таб)	Дараачийн босоо таб-д очих

`scanf` –н оролтын хэвшүүлэлт

- **`scanf`**
 - Гаралтын нэгэн адил оролтыг хэвшүүлж болно
 - **`scanf`** –н хувиргалтын тодорхойлогч нь **`printf`** –ээс бага зэрэг ялгаатай

scanf –н хувиргалтын тодорхойлогч

Хувиргалт тодорхойлогч	Тайлбар
Бүхэл	
d	Тэмдэгтэй 10 –тын бүхлийг уншина. Харгалзах аргумент нь бүхлийн заагч байна
i	Тэмдэгтэй 8, 10, 16 –тын бүхлийг уншина. Харгалзах аргумент нь бүхлийн заагч байна
o	8 –тын бүхлийг уншина. Харгалзах аргумент нь тэмдэггүй бүхлийн заагч байна
u	Тэмдэггүй 10 –тын бүхлийг уншина. Харгалзах аргумент нь тэмдэггүй бүхлийн заагч байна
x эсхүл X	16 –тын бүхлийг уншина. Харгалзах аргумент нь тэмдэггүй бүхлийн заагч байна
h эсхүл l	Бүхэл хувиргалтын тодорхойлогчийн өмнө бичигдэнэ. Ингэснээрээ short эсхүл long бүхлийг уншихыг заана.

scanf –н хувиргалтын тодорхойлогч

Хувиргалт тодорхойлогч	Тайлбар
Хөвөгч таслалтай тоо	
e, E, f, g, G	Хөвөгч таслалтай тоог уншина. Харгалзах аргумент нь хөвөгч таслалтай хувьсагчийн заагч байна.
l L	Дурын хөвөгч таслалтай хувиргалтын тодорхойлогчийн өмнө бичигдэх ба double, long double утга оруулахыг заана. Харгалзах аргумент нь double, long double хувьсагчийн заагч байна.
Тэмдэгт, мөр	
c	Тэмдэгт уншина. Харгалзах аргумент нь char төрлийн заагч байна. NULL ('\0') тэмдэгт нэмэхгүй
s	Мөр уншина. Харгалзах аргумент нь массив буюу char төрлийн заагч байна. NULL ('\0') тэмдэгтийг төгсгөлд нь автоматаар нэмэнэ

scanf –н хувиргалтын тодорхойлогч

```
1. /* Ex_82 Reading integers */
2. #include <stdio.h>
3.
4. int main( void )
5. {
6.     int a;
7.     int b;
8.     int c;
9.     int d;
10.    int e;
11.    int f;
12.    int g;
13.
14.    printf( "Enter seven integers: " );
15.    scanf( "%d%i%i%i%o%u%x", &a, &b, &c, &d, &e, &f, &g );
16.
17.    printf( "The input displayed as decimal integers is:\n" );
18.    printf( "%d %d %d %d %d %d %d\n", a, b, c, d, e, f, g );
19.
20.    return 0; /* indicates successful termination */
21.
22.} /* end main */
```

Diagram illustrating the mapping of format specifiers in the `scanf` format string to their corresponding integer types and bit widths:

- `%d` нь 10 –тын бүхлийг оруулахыг заана
- `%i` нь бүхлийг оруулахыг заана
- `%o` нь 8 –тын бүхлийг оруулахыг заана
- `%u` нь тэмдэггүй 10 –тын бүхлийг оруулахыг заана
- `%x` нь 16 –тын бүхлийг оруулахыг заана

scanf –н хувиргалтын тодорхойлогч

```
1. /* Ex_83 Reading floating-point numbers */
2. #include <stdio.h>
3.
4. /* function main begins program execution */
5. int main( void ) {
6.     {
7.         double a;
8.         double b;
9.         double c;
10.
11.         printf( "Enter three floating-point numbers: \n" );
12.         scanf( "%le%lf%lg", &a, &b, &c );
13.
14.         printf( "Here are the numbers entered in plain\n" );
15.         printf( "floating-point notation:\n" );
16.         printf( "%f\n%f\n%f\n", a, b, c );
17.
18.         return 0; /* indicates successful termination */
19.
20. } /* end main */
```

e, f, g нь хөвөгч цэгтэй тоо оруулахыг заана

l нь double, long double -г оруулахыг заана

Enter three floating-point numbers:

1.27987 1.27987e+03 3.38476e-06

Here are the numbers entered in plain

floating-point notation:

scanf –н хувиргалтын тодорхойлогч

```
1. /* Ex_84 Reading characters and strings */
2. #include <stdio.h>
3.
4. int main( void )
5. {
6.     char x;
7.     char y[ 9 ];
8.
9.     printf( "Enter a string: " );
10.    scanf( "%c%s", &x, y );
11.
12.    printf( "The input was:\n" );
13.    printf( "the character \"%c\" ", x );
14.    printf( "and the string \"%s\"\n", y );
15.
16.    return 0; /* indicates successful termination */
17.
18.} /* end main */
```

с нь тэмдэгт оруулахыг заана

s нь мөр оруулахыг заана

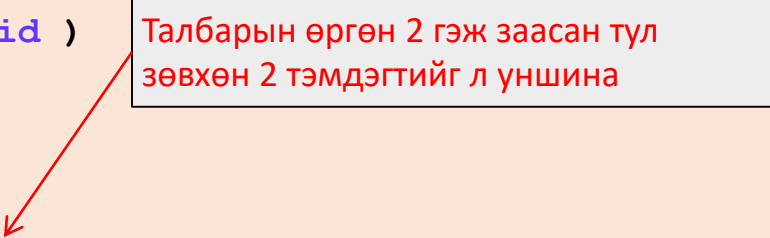
Enter a string: Sunday

The input was:

the character "s" and the string "unday"

scanf –н хувиргалтын тодорхойлогч

```
1. /* Ex_85 inputting data with a field width */
2. #include <stdio.h>
3.
4. int main( void )
5. {
6.     int x;
7.     int y;
8.
9.     printf( "Enter a six digit integer: " );
10.    scanf( "%2d%d", &x, &y );
11.
12.    printf( "The integers input were %d and %d\n", x, y );
13.
14.    return 0; /* indicates successful termination */
15.
16.} /* end main */
```



Талбарын өргөн 2 гэж заасан тул
зөвхөн 2 тэмдэгтийг л уншина

```
Enter a six digit integer: 123456
The integers input were 12 and 3456
```

```

1. /* Ex_86 Reading and discarding characters from the input stream */
2. #include <stdio.h>
3.
4. int main( void )
5. {
6.     int month1; /* define month1 */
7.     int day1;   /* define day1 */
8.     int year1;  /* define year1 */
9.     int month2; /* define month2 */
10.    int day2;   /* define day2 */
11.    int year2;  /* define year2 */
12.
13.    printf( "Enter a date in the form mm-dd-yyyy: " );
14.    scanf( "%d%c%d%c%d", &month1, &day1, &year1 );
15.
16.    printf( "month = %d  day = %d  year = %d\n\n", month1, day1, year1 );
17.
18.    printf( "Enter a date in the form mm/dd/yyyy: " );
19.    scanf( "%d%c%d%c%d", &month2, &day2, &year2 );
20.
21.    printf( "month = %d  day = %d  year = %d\n", month2, day2, year2 );
22.
23.    return 0; /* indicates successful termination */
24.
25.} /* end main */

```

* -г ашигласан бол scanf 2 оролтын
хоорондохыг хэрэгсэхгүй алгасна

Enter a date in the form of mm-dd-yyyy: 11-18-2003