Лекція 1: Базові типи Сі. Бітові та булеві операції. Розгалудження. Цикли

Повторення

```
#include <stdio.h> // заголовки
#include <math.h>
int main(){ // головна функція
   int a, b=3; // оголошення та ініціалізація
   Scanf(«%d»,&a); //введення
   int z=a*(a+b) / b; // вираз, ділення цілих чисел
   double z2 = a*(a+b+.0) / b; // ділення
   printf(«\n z=%d, z2=%f»,z,z2); // виведення
```

повторення:

```
Вираз:
float A = (x-1)/(y^*y+1);
A += 2;
Цілий тип:
int B = 23;
B = (B+2)/3;
B++; // B=B+1
B--; //B=B-1
```

Lecture 1

Лекція 1. Вигляд мінімальної програми на Сі. Програмні блоки. Як створити проект та компіляція програми. Введення-виведення для стандартного Сі на Сі++. Базові операції. Лінійні програми. Дійсні типи даних. Використання опцій для різного типу вводу/виводу та точності вхідних і вихідних параметрів.

Лекція 2. Оператори керування на Сі та Сі++. Типи циклів та їх реалізація. Цілі типи даних. Логічні операції. Бітові операції.

Базові типи Сі

- -Порожній тип void (empty type) неповний тип
- -Цілі (integer)
- -- символи(char): can be set as integer (0..255) signed char, unsigned char, char
- ➤Знакові цілі (signed integer types):
- •standard: signed char, short, int, long, long long (since C99),
- •extended: визначені додатково, e.g. __int128,
- ≻Натуральні цілі (unsigned integer types):
- •standard: Bool (since C99), unsigned char, unsigned short, unsigned int, unsigned long, unsigned long long (since C99),
- •extended: визначені додатково, e.g. __uint128

- Дійсні float

- ▶Дісні (float types): float, double, long double
- ➤Комплексні (complex types): float _Complex, double _Complex, long double _Complex,
- ▶Уявні (imaginary types): float Imaginary, double Imaginary, long double Imaginary

Цілі типи

- **Символи**(**char**): can be set as integer (0..255) – *signed char, unsigned char, char, wchar t(C95), wschar t(C95)*

>Знакові цілі (signed integer types):

•standard: signed char, short, int, long, long long (since C99),

•extended: визначені додатково, e.g. int128, int8 t, int32 t etc

≻ Натуральні цілі (unsigned integer types):

•standard: _Bool (bool) (since C99), unsigned char, unsigned short, unsigned int, unsigned long, unsigned long long (since C99),

•extended: визначені додатково, e.g. __uint128, int8_t, int32_t etc

Звичайні цілі типи

Специфікатор типу	Еквівалент	Специфікатор типу	Еквівалент
short		long	
short int	short int	long int	long int
signed short		signed long	
signed short int		signed long int	
unsigned short		unsigned long	unsigned long int
unsigned short int	unsigned short int	unsigned long int	
int		long long	long long int (C99)
signed	int		
signed int		long	
unsigned	unsigned int	long int	long int
unsigned int		signed long	
unsigned int		signed long int	

Специфікатори

Суфікс	Основа числення 10	Основи числення 8 та 16
		int
no suffix	int	unsigned int
	long int	long int
	unsigned long int (until C99)	unsigned long int
	long long int (since C99)	long long int(since C99)
		unsigned long long int(since C99)
и або U	unsigned int	unsigned int
	unsigned long int	unsigned long int
	unsigned long long int(since C99)	unsigned long long int(since C99)
I або L	long int unsigned long int(until C99)	long int
		unsigned long int
		long long int(since C99)
	long long int(since C99)	unsigned long long int(since C99)
I/L та u/U	unsigned long int	unsigned long int
	unsigned long long int(since C99)	unsigned long long int(since C99)
II 050 I I	long long int/since COO)	long long int(since C99)
II або LL	long long int(since C99)	unsigned long long int(since C99)
II/LL та <u>u/</u> l	J unsigned long long int(since C99)	unsigned long long int(since C99)

Цілі типи

```
int x=1; // integer (usually 4 bytes)
char y=0, z='a'; //character (1 byte)
short (short int) t =1;
long (long int) z=1L; // long integer
long long (long long int) z=1L;
```

```
Для того щоб зробити беззнаковий додається unsigned: unsigned xx=1; unsigned char yy=-1; ... і т.д.
```

Приклади застосувань цілих типів

```
unsigned d = 69; // десяткове представлення натурального типу
unsigned int o = 0103; // вісімкове представлення
unsigned x = 0x43; // шістнадцяткове представлення
unsigned int X = 0X43; // шістнадцяткове
printf("%u,%o,%x,%X",d,o,x,X);
signed long long int d = 69L; // десяткове представлення довгого цілого
signed long long o = 0103; // вісімкове, довгий тип
long long int x = 0x43; // шістнадцяткове, довгий
long long X = 0X43; // шістнадцятковий, довгий цілий
printf("%lld,%LLo,%LLx,%llX",d,o,x,X);
```

Введення/виведення цілих типів

```
printf("x=%d",x); // цілий тип
printf("x=%i",x); // цілий тип
printf("x=%u",x); // натуральний тип
printf("x=%Lu",x); // довгий натуральний
printf("x=%ld",x); // довгий цілий
printf("x=%x",x); //шістнадцяткова система
printf("x=%o",x); //вісімкова система
```

Межі та розміри

В модулі limits.h> зберігаються справжні межі цілих типів

```
INT_MAX, UNSIGNED_MIN etc
```

Для того, щоб вивести довжину кожного типу (необовязково цілого) використовується:

```
size_t m = sizeof(int);
long x;
size_t m = sizeof(x);
size_t m = sizeof('a');
```

Оператор sizeof

#include <stdio.h>

```
int main(void){
short x;
  // type argument:
printf("sizeof(float) = %zu\n", sizeof(float));
printf("sizeof(void(*)(void)) = %zu\n", sizeof(void(*)(void)));
printf("sizeof(char[10]) = %zu\n", sizeof(char[10]));
//printf("sizeof(void(void)) = %zu\n",sizeof(void(void)));//Error:function type
//printf("sizeof(char[]) = %zu\n", sizeof(char[])); // Error: incomplete type
// expression argument:
printf("sizeof 'a'
                  = %zu\n", sizeof 'a'); // type of 'a' is int
//printf("sizeof main
                            = %zu\n", sizeof main); // Error: Function type
printf("sizeof &main
                            = %zu\n", sizeof &main);
printf("sizeof \"hello\"
                            = %zu\n", sizeof "hello"); // type is char[6]
                         = %zu\n", sizeof x); // type of x is short
printf("sizeof x
                           = %zu\n", sizeof(x+1)); // type of x+1 is int
printf("sizeof (x+1)
```

Ранг	Операції	Напрямок виконання
1	() (виклик функції), [], ->, "."	>>>
2	!, ~, +, - (унарні), ++,, *, (тип), sizeof,	<<<
3	.*, ->* - Ci++	>>>
4	*, /, % (бінарні)	>>>
5	+, - (бінарні)	>>>
6	<<,>>>	>>>
7	<, <=, =>, >	>>>
8	==, !=	>>>
9	& (порозрядна)	>>>
10	^	>>>
11	(порозрядна)	>>>
12	&& (логічна)	>>>
13	(логічна)	>>>
14	?: (тернарна)	<<<
15	=, +=, -=, *=, /=, %=, ^=, =, <<=,>>=	<<<
16	"," (кома)	>>>

Оператор ділення(/): особливості Сі

```
int x,y; // цілі числа
scanf("%d %d",&x, &y); // введення: наприклад 7 та 3
float z = x/y; // Увага!!!!: результат 7/3 =2
printf("z=%f",z); //z=2.0
unsigned m = x\%y; // m = 1
printf("m=%u", m); // m=1
z = (float)x / y; // перетворення типу (float) x потім ділення
printf("z=%f",z); // z=3.3333
z = (x+0.0)/y; // перетворення типу x + 0.0 - дійсне потім ділення
printf("z=%f",z);
z = x/(float)y; // перетворення типу y- дійсне потім ділення
printf("z=%f",z);
```

Умови та булевий тип

Старий С (до С99) не мав булевого типу. Однак булевий тип можна зобразити цілим (0— false, 1 (та всі інші натуральні)— true). Так в С можна визначити:

int x = (1>0); int y = (1<0);// x — true, y — false

Булевий тип та умови можна визначити:

int a = (x>1); a = (x<=y); a = (y==1); a = (x!=0);

Приклад: булевий тип

```
#include <stdio.h>
#include <stdbool.h> // 3 С99 зявився булевий тип
int main(){
  bool a = true, b = (1 < 0); // ініціалізація напряму та через
  УМОВУ
  bool c = a&&b; // c==false, операція множення
  int d = a||b|; // операція додавання d = = ?
  char e = !b; // заперечення e = ?
  printf("(%d %i %u",c,d,e); //результат може бути будь
  який цілий тип
```

Операції з умовами

```
!A; //логічне заперечення(NOT, найпріорітетніший)
А && В; // логічне множення (AND, середній пріорітет)
А | В; // логічне додавання (OR, найнижчий пріорітет)
Examples: //x=1; y=1; y=
1) (x>0)&&(y>=0);
2) (x==0) || (y <= 0)
3) !(x==y) && ((z!=x)|| !(z1=y))
```

Операції з бітами

Можна робити булеві операції безпосредньо з бітами чисел (краще з типом unsigned)

```
unsigned A=0x0001, B=0x0010;
```

- 1) A & B = 0x0; // побітове множення
- 2) А | В = 0х11; // побітове додавання
- 3) A ^ B = 0xFF00; // виключне або
- 4) ~A = 0xFFFE; // заперечення

Бітови зсуви

Бітовий зсув дозволяє рухатись по двійковому представленню числа:

```
unsigned z=8,z2=23; //z=00100, z2=10111
unsigned t = z>>2; t2=t>>2; //зсув вліво
// t = 2(00010); t2 = 5(00101)
unsigned r = (1 << 7); r2 = (t2 << 5);
//r=128, r2=1011100000 зсув вправо
z<<=1;//z =16 // унарний зсув вліво
r>>=1; //r =64 // унарний зсув вправо
```

Розгалудження в С

C/C++ має сінтаксис, що схожий на аналогічні в C#, Java, Javascript

Тіла циклів та розгалуджень виділяються в фігурні дужки {} (аналог begin...end в Pascal або табуляції в Python)

Розгалудження (IF..ELSE)

```
if (<умова>)
    ***// код
// тіло умови
else
  *** //код
2) if (<умова>) {
    ***// код
```

Умова, це **булевий** вираз у скобках, як:

```
a) (x==0)
b) (x>=1)
c) (x!=0)
d) (x<10)
```

Тіло розгалудження це будь-яка послідовність команд С:

```
1) if(x==0) { printf(«x is equal to 0»); }
2) if(x!=0) { printf(«x is not equal to 0»); y=1/x; z=x+y; }
else {printf(«x is equal 0»);}
3) if(x>=0) { printf(«x is not less then 0»); y=sqrt(x);}
else {printf(«x is less then 0»);x=-x; y=sqrt(x);}
```

```
if (<умова1>)
                                        Умови можуть бути
                                        додани послідовно
                                        та/або
     ***// КОД
                                            вкладені
                                 1) if(x>=0) { printf(«x is not less then 0»); }
                                 else if (x==0) {printf(«x is 0»);}
else if (<ymoba2>)
                                 else { printf(«x is less then 0»); } // послідовно
                                 2) if(x>0) { printf(«x is greater then 0»);
                                          if (x<1){
  if (<умова3>) {
                                              printf(«x is less then 1»;) // вкладено
  *** //some code
                                 else if (x==0) { printf(«x is 0»);} //послідовно
       if(<умова4>)
                                 else {
                                    if (x>=-1){ printf(«x is greater then -1»);
                                        // ще вкладено
```

C vs Python

```
if x > = 2:
if (x>=2) {
   y = sqrt(x);
                                          y = sqrt(x)
else{
                                      else:
  y=0;
                                          y=0
                                                      If x > = 2:
         if (x>=2) {
                                                        y = sqrt(x);
            y = sqrt(x);
                                                      Elif x==0:
                                                        y = sqrt(x)+1;
         else if(x==0){
                                                      Else:
           y = sqrt(x)+1;
                                                         y=0
         else{ y=0;}
```

Цикли: з передумовою

```
while(<yмова>)
 *** // код
Приклад 1:
i=1;
while(i<n)
F = F^*i;
 j++;
```

```
Python
while <умова>:
 *** // код
Приклад 1:
i=1
while i<n:
  F = F*i
  i+=1
```

Цикли: з післяумовою

```
do
{
    *** // код
}while(<умова>);
Приклад 2:
i=1;
do
{
    F = F*i;
i++;
}while(i<n);
```

```
Python
 while True:
  *** // код
  if not <умова> break
Приклад 2:
i=1;
while True
 F = F*i;
 j++;
 if i>n: break
```

Цикл for

```
for(<початок>; <умова продовження>; <шаг>)
{ ***//код (тіло циклу)
Приклади:
1)int i; for( i=0;i<n;i++) {f=f*i; } // факторіал
2) for(int i=n;i>0;i--) {f=f*i;} // факторіал
3) for(int p=1;p<n;p*=2) \{printf("i=%d",i);\}
 //1,2,4,8,...2^{\log(n)}
```

Керування break, continue

```
for(i=0;;i++){ printf("i=%d",i); if(i==n) break; }
for(i=-n;i<n;i++) {
    printf("i=%d",i); if(i==0) continue;
}</pre>
```

Exit, abort

```
#include <stdio.h>
#include <stdlib.h>
int main(){
int n; scanf("%d",&n);
if(n==0) abort();
for(int i=n; i>=0; i-=2){
  if (i==0) exit(1);
```

Література

- 1. Річі К. Мова програмування Сі
- 2. Керниган, Б. Язык программирования Си. Задачи по языку Си / Б. Керниган, Д. Ритчи, А. Фьюэр. М. : Финансы и статистика, 1985. 279 с.
- 3. С у задачах і прикладах : навчальний посібник із дисципліни "Інформатика та програмування" / А.П. Креневич, О.В. Обвінцев. К. : Видавничо-поліграфічний центр "Київський університет", 2011. 208 с.
- 4. Уэйт М. Язык Си / М. Уэйт, С. Прата, Д. Мартин. М. : Мир, 1988. 512 с