Міністерство освіти і науки України

Національний університет “Львівська Політехніка”

Лабораторна робота №3А

Виконав:
Студент групи АП-11

Білий Анатолій Іванович

Прийняв:

Чайковський І.Б.

Львів – 2024

Тема роботи: Логічні і бітові операції та вирази мови С.

Мета роботи: Дослідження властивостей операцій порівняння, логічнихі бітових мови програмування С.

Попередні відомості.

Операції порівняння – бінарні, причому обидва операнди повинні бути арифметичного типу, або вказівниками. Результат цілочисельний: 0 (хибність) або 1 (істинність). Тип результату int.

вираз < вираз
вираз > вираз
вираз <= вираз
вираз >= вираз

Операції рівності і нерівності відносять до цієї ж групи. Важливо правильно витримувати синтаксис знаку «логічне дорівнює» - ця операція не виконує присвоювання:

вираз == вираз
вираз != вираз

Результатом цих операцій є 0, якщо задане відношення хибне, і 1, якщо істинне. Тип результату int. Ці операції мають нижчий пріоритет, ніж операції попередньої групи, наприклад, у виразі a < b = = c < d спочатку здійснюються порівняння a < b та c < d, результати кожного з них мають значення 0 або 1, після чого операція = = дає результат 0 або 1.

Для логічних операцій характерне те, що і операнди, і результат мають цілий тип і трактуються як логічні (“Так” – 1, “Ні” – 0).

Умовна тримісна операція (тернарна).

На відміну від унарних і бінарних операцій умовна тернарна операція використовується з трьома операндами. В зображенні умовної операції використовуються два символи ‘?’ і ‘:’ і три вирази:

вираз1 ? вираз2 : вираз3.

Першим обчислюється виразу1. Якщо воно істинне, тобто не дорівнює нулю, то обчислюється значення виразу2, яке стає результатом. Якщо при обчисленні виразу 1 отримується 0 (нуль), то в якості результату приймається значення виразу3.

Приклад:

x<0? – x : x;

Вираз повертає абсолютну величину змінної x.

Коли виникає необхідність роботи з величинами, що записані як біти, в частинах машинного слова, застосовуються «бітові операції». До таких відносяться:

1. Операції зсуву (визначені тільки для цілочисельних операндів): операнд лівий операція зсуву операнд правий

<< – зсув ліворуч бітового представлення лівого цілочисельного операнда на кількість розрядів, що дорівнює значенню правого цілочисельного операнда.

>> – зсув праворуч бітового представлення лівого цілочисельного операнда на кількість розрядів, що дорівнює значенню правого цілочисельного операнда.

2. Доповнення (бітове НЕ):

~ операнд

Це унарна операція, яка доповнює значення біту кожного розряду операнду до 1. Операнд повинен мати тип int.

3.Бітове І: вираз & вираз

Результатом є бітова функція І операндів. Результат обчислюється як бітовий – для кожного розряду операндів згідно таблиці істинності операції логічне І і записується у відповідний розряд. Операція застосовується тільки до операндів типу int.

ОПЕРАЦІЯ	**АСОЦІАТИВНІСТЬ**	
( ) [ ] -> .	ЗЛІВА НАПРАВО	→
! ~ ++ -- - (ТИП) * & SIZEOF	ЗЛІВА НАПРАВО	→
* / %	ЗЛІВА НАПРАВО	→
+ -	ЗЛІВА НАПРАВО	→
<< >>	ЗЛІВА НАПРАВО	→
< <= > >=	ЗЛІВА НАПРАВО	→
== !=	ЗЛІВА НАПРАВО	→
&	ЗЛІВА НАПРАВО	→
^	ЗЛІВА НАПРАВО	→
|	ЗЛІВА НАПРАВО	→
&&	ЗЛІВА НАПРАВО	→
||	ЗЛІВА НАПРАВО	→
?:	ЗПРАВА НАЛІВО	←
= += -= і.т.д.	ЗПРАВА НАЛІВО	←
, (кома)	ЗЛІВА НАПРАВО	→

Завдання

Приклад 1

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

```
void main(void)
{
float var1, var2;
printf("Введіть перше число (var1): ");
scanf("%f", &var1);
printf("Введіть друге число (var2): ");
scanf("%f", &var2);
printf("var1 > var2 дає %d\n", var1 > var2);
printf("var1 < var2 дає %d\n", var1 < var2);
printf("var1 == var2 дає %d\n", var1 == var2);
printf("var1 >= var2 дає %d\n", var1 >= var2);
printf("var1 <= var2 дає %d\n", var1 <= var2);
printf("var1 != var2 дає %d\n", var1 != var2);
printf("!var1 дає %d\n", !var1);
printf("!var2 дає %d\n", !var2);
printf("var1 || var2 дає %d\n", var1 || var2);
printf("var1 && var2 дає %d\n", var1 && var2);
}
```

```
Введіть перше число (var1): 23
Введіть друге число (var2): 11
var1 > var2 дає 1
var1 < var2 дає 0
var1 == var2 дає 0
var1 >= var2 дає 1
var1 <= var2 дає 0
var1 != var2 дає 1
!var1 дає 0
!var2 дає 0
var1 || var2 дає 1
```

var1 && var2 дає 1

Приклад 2

```
#include <stdio.h>
#define TRUE "ІСТИНА"
#define FALSE "ХИБНІСТЬ"
void main(void)
{
float var1, var2;
printf("Введіть перше число (var1): ");
scanf("%f", &var1);
printf("Введіть друге число (var2): ");
scanf("%f", &var2);
printf("var1 > var2 це %s\n", var1 > var2 ? TRUE : FALSE);
printf("var1 < var2 це %s\n", var1 < var2 ? TRUE : FALSE);
printf("var1 == var2 це %s\n", var1 == var2 ? TRUE : FALSE);
printf("var1 >= var2 це %s\n", var1 >= var2 ? TRUE : FALSE);
printf("var1 <= var2 це %s\n", var1 <= var2 ? TRUE : FALSE);
printf("var1 != var2 це %s\n", var1 != var2 ? TRUE : FALSE);
printf("var1 || var2 це %s\n", var1 || var2 ? TRUE : FALSE);
printf("var1 && var2 це %s\n", var1 && var2 ? TRUE : FALSE);
printf("!var1 це %s\n", !var1 ? TRUE : FALSE);
printf("!var2 це %s\n", !var2 ? TRUE : FALSE);
}
```

```
Введіть перше число (var1): 36
Введіть друге число (var2): 18
var1 > var2 це ІСТИНА
var1 < var2 це ХИБНІСТЬ
var1 == var2 це ХИБНІСТЬ
```

var1 >= var2 це ІСТИНА

var1 <= var2 це ХИБНІСТЬ

var1 != var2 це ІСТИНА

var1 || var2 це ІСТИНА

var1 && var2 це ІСТИНА

!var1 це ХИБНІСТЬ

!var2 це ХИБНІСТЬ

Приклад 3

```
#include <stdio.h>
int main() {
    int x, y, z;
    x = 2;
    y = 1;
    z = 0;
    x = x && y || z;
    printf("%d\n", x);  // Виведе результат першого виразу
    printf("%d\n", x || !y && z);  // Виведе результат другого виразу
}
```

```
// 1
// 1

// Приклад 4
#include <stdio.h>
#include <conio.h>
void main()
{
    int a=0, b=3,c;
    c = b%2 || (a >= 0) && (++b/2*a)==0;
```

```
    printf("a=%d, c=%d\n",a,c); //a=0,c=1
    getch();
}
```

```
// a=0, c=1

// Приклад 5
#include <stdio.h>
#include <conio.h>
void main()
{
    int a = 1, b = 0,c;
    c = b*2 || (a >= 0) && (++b*a) == 0;
    printf(" c=%d\n",c); //c=0
    getch();
}
```

```
// c=0

// Приклад 6
#include<stdio.h>
#include<conio.h>
void main()
{
    int x=1, y=2,z;
    z=(x/2*7 <= 0) && (y<0) || (y%x==0);
    printf("z=%d\n",--z); //z=0
    getch();
}
```

```
// z=0

// Приклад 7
#include<stdio.h>
#include<conio.h>
void main()
{
   int x=1,z,b=0,y=2;
   z=(x++*y >= 0) || b++ || (x/y*3==0);
   printf("z=%d\n",z); //z=1
   getch();
}
```

```
// z=1

// Приклад 8
#include<stdio.h>
#include<conio.h>
void main()
{
   int x=1,y=0,z=2; int a=0;
   z=((a=x++)*y==0 || a<0 && z);
   printf("z=%d\n",z); //z=1
   getch();
}
```

```
// z=1

// Приклад 9
#include<stdio.h>
```

```
#include<conio.h>
void main()
{
   int x=2,z,y=0;
   z=(x==0) && (y=x) || (y>0);
   printf("z=%d\n",z); //z=0
   getch();
}
```

```
// z=0

// Приклад 10
#include<stdio.h>
#include<conio.h>
void main()
{
   int x=0,y=3,z;
   z=(++x>y || y-- && y>0);
   printf("z=%d\n",z); //z=1
   getch();
}
```

```
// z=1

// Приклад 11
#include <stdio.h>
#include <conio.h>
void main()
{
unsigned int x=2,y = 1, z=3,res;
```

```
 char chx = 0xAF;
 printf("%u\n",x&y|z); /*3*/
 x=y=z=2;
 printf("%u\n",x|y&z); /*2*/
 x=3; y=0; z=1;
 printf("x^y|~z=%hu\n",x^y|~z); /*65535*/
 printf("3|0^~1=%hu\n",x|y^~z); /*65535==11111111*/
 x=1;y=2;z=0;
 printf("1&2|0=%u\n",x&y|z); /*0*/
 printf("~1^2&0=%hu\n",~x^y&z); /*65534==11111110*/
 printf("2|0&1=%u\n",y|z&x); /*2*/
 printf("2++&~0|~1=%hu\n",y++&~z|~x);/*65534==11111110*/
 printf("~3|1&++0=%hu\n",~y|x&++z); /*65533==11111101*/
 x = 0xAF; printf("%X\n",x>>4); /*A*/
 chx<<=7; printf("0x=%X\n",chx); /*(FF)80 ==10000000*/
 getch();
 }
```

```
// 3
// 2
// x^y|~z=65535
// 3|0^~1=65535
// 1&2|0=0
// ~1^2&0=65534
// 2|0&1=2
// 2++&~0|~1=65534
// ~3|1&++0=65533
// A
// 0x=FFFFFF80
```

```
// Приклад 12
#include <stdio.h>
int main() {
    unsigned char x = 255, y = 0177;
    printf("%hhu\n", x & y); // 127
    x = 40; // '(tm)' в ASCII коді відповідає 40
    y = 017;
    printf("%hhu\n", x & ~y); // 240
    y = 127;
    printf("%hhu\n", x & y); // 40
    y = 128;
    printf("%hhu\n", x | y); // 168
}
```

```
// 127
// 32
// 40
// 168
```

Контрольні запитання

1. Призначення операторів порівняння та тип результату:

- Оператори порівняння використовуються для порівняння значень. Тип результату – логічне значення (true або false), що вказує на те, чи вірне порівняння.

2. Особливість оператора "логічне дорівнює":

- Оператор "логічне дорівнює" (===) порівнює значення обох операндів і їх типи даних. Це означає, що не тільки значення повинні бути однакові, але й типи даних повинні бути однакові.

3. Відрізняються операнди в логічних операціях від операндів в операціях порівняння:

- Операнди в логічних операціях це логічні значення (true або false), тоді як

операнди в операціях порівняння - це значення, що порівнюються.

4. Пріоритети операцій:

- Порядок виконання операцій може залежати від пріоритету. Наприклад, зазвичай арифметичні операції мають вищий пріоритет, ніж логічні.

. Таблиця істинності логічного І:

-

A	B	A I B
0	0	0
0	1	0
1	0	0
1	1	1

Висновок: під час виконання даної лабораторної роботи я дослідив властивості операцій порівняння, логічних і бітових мови програмування С.