

Міністерство освіти і науки України Національний університет “Львівська Політехніка”



Лабораторна робота №7
з дисципліни «Програмування частина № 2»

Виконав:

Студент групи АП-11

Іщенко Василь

Прийняв:

Чайковський І.Б.

«Арифметичні операції та вирази мови С»

Мета роботи: ознайомитися з синтаксисом арифметичних операцій, їх пріоритетом застосувань, навчитися їх використовувати для обчислень математичних виразів.

Теоретичні відомості

Мова С була розроблена в процесі створення операційної системи UNIX, тому можна зрозуміти, які принципові можливості в ній реалізовані: це максимальна гнучкість при діалоговому режимі роботи комп'ютера, представлення повідомлень системи і користувача в максимально простій і зрозумілій формі і, водночас, спроможність вибору адекватної реакції в найскладніших ситуаціях. Мова С поєднує в собі можливості прямої адресації і побітових операцій, як в Ассемблері, з використанням великої кількості (декілька сотень) функцій найвищого рівня. При використанні бібліотеки графічних функцій мова С отримала практично необмежені можливості для розробки діалогових програмних засобів. Проте, мова С має суттєвий недолік з точки зору потреб розробки радіотехнічних задач: тут недостатньо розвинені операції арифметики, зокрема, повністю відсутня комплексна арифметика, і її імітація призводить до генерування недостатньо ефективних кодів, що значно збільшує потреби часу при проведенні значних за обсягом математичних обчислень. Фірма Microsoft розробила власну версію мови С з інтерфейсом подібним до мови ФОРТРАН, найбільш пристосованою для математичних розрахунків і генеруючою найефективніші машинні коди. Паралельно на фірмі Borland велась розробка іншої версії мови С, перші варіанти якої мали назву "Turbo C", а пізніші - "Borland C", "C++", причому в версіях "C++" комплексну арифметику реалізують за допомогою класу об'єкта. Сервісна оболонка мови С призначена для розробки та відладки програм і включає в себе засоби роботи з файлами, їх редагування, запуску виконуваних файлів, а також різноманітні режими компіляції і збірки виконаного модуля, розвинуті засоби відлагоджування програми - детальна діагностика помилок, 2 можливість виконання по кроках з переглядом проміжних результатів, можливість прослідкувати вміст певних змінних тощо.

Приклад 1

```
#include <stdio.h>
int main() {
int a=67;
int b=33;
int c = a+b+7;
printf("a+b+7 =%d\n", c);
return 0;
}
```

a+b+7 =107

Приклад 2

```
#include <stdio.h>
int main() {
int a=38;
int b=7;
int c=a+5*b;
printf("c = %d \n", c);
}
```

c = 73

Приклад 3

```
#include <stdio.h>
int main() {
int a=8;
int b=7;
int c=(a+5)*b;
printf("c = %d \n", c);
}
```

c = 91

Приклад 4

```
#include <stdio.h>
int main() {
int a=8;
int b=++a;
printf{("a=2%d \n", a);
printf("b=%d",b);
}
```

a=29

b=9

Приклад 5

```
#include <stdio.h>
int main() {
int a=8;
int b=a++;
printf("a=%d\n", a);
printf("b=%d", b);
}
```

a=9

b=8

Приклад 6

```
#include <stdio.h>
int main() {
int a=8;
int b=a--;
printf("a=%d\n", a);
printf("b=%d", b);
}
```

a=7

b=8

Приклад 7

```
#include <stdio.h>
int main() {
int a=8;
int b=--a;
printf("a=%d\n", a);
printf("b=%d", b);
}
```

a=7

b=7

Завдання 3

```
#define PRAISE "О, яке чудове ім'я!"
#include <stdio.h>
#include <string.h>
#include <windows.h>
int main() {
    SetConsoleCP(65001);
    SetConsoleOutputCP(65001);
    char name[50];
    printf("Як Вас звати?\n"); scanf("%s",name);
    printf("Привіт, %s. %s\n",name,PRAISE);
    printf("Ваше ім'я складається з %d літер і займає %d комірок пам'яті\n",
    strlen(name)/2, sizeof name);
    printf("Вітальна фраза складається з %d літер", strlen (PRAISE)/2);
    printf(" і займає %d комірок пам'яті\n", sizeof PRAISE); }
```

Як Вас звати?

Василь

Привіт, Василь. О, яке чудове ім'я!

Ваше ім'я складається з 6 літер і займає 50 комірок пам'яті

Вітальна фраза складається з 15 літер і займає 32 комірок пам'яті

Завдання 4

```
#include <stdio.h>
#include <conio.h>
void main() {
    float x=1.4, y=2.0; int z;
    z = x/2*7 + y/4 - 1;
    printf("z=%d\n",--z);
    getch();
}
```

z = 3

```
#include <stdio.h>
#include <conio.h>
void main() {
    int x=2,z;
    float y;
    z=0.5*(y = 2.3*x) + x++/3*y;
    getch();
}
```

z = 2

```
#include <stdio.h>
#include <conio.h>
void main() {
int x,y=3;
float z;
z=1.1*(x++y/2.)+ 0.3*x;
printf("z=%4.1f\n",z);
getch();
}
```

$z = 2.8$

Висновок: Я ознайомився з синтаксисом арифметичних операцій, їх пріоритетом застосувань, навчився їх використовувати для обчислень математичних виразів