

Міністерство освіти і науки України Національний університет “Львівська Політехніка”



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

Виконав:

Студент групи АП-11

Іщенко Василь

Прийняв:

Чайковський І.Б.

Тема роботи: Арифметичні операції та вирази мови С.

Мета роботи: Дослідження принципів створення математичних виразів при складанні програм для виконання обчислень за допомогою різних операцій мови програмування С.

Попередні відомості.

Елементарною коміркою машинної пам'яті є біт. Біт – це елемент інформації, який може приймати значення 1 або 0. Фізично це означає наявність або відсутність електричного струму в певній ділянці електричного кола. Такий спосіб представлення елементу інформації пристосований для двійкової системи числення, яка використовується в ЕОМ. Група з восьми біт утворює байт. В одному байті можна записати беззнакове ціле число від 0 до 255 (2^8 - восьмий степінь числа 2) або знакове від 0 до 127. Звичайно одного байту недостатньо для запису більш складних даних, тому з двох (або чотирьох) байт утворюється машинне слово - вектор бітів, який розглядається апаратною частиною ЕОМ як єдине ціле. Число бітів у слові називається довжиною слова, залежить від апаратної реалізації комп'ютера і, як правило, буває довжиною 16 або 32 біти. Пам'ять обчислювальної машини поділяється логічно на слова. Слово має довжину, достатню для розміщення в ньому команди або цілого числа. Всі дані, якими оперує мова С, підрозділяються на типи. Кожен тип даних має свій спосіб запису в пам'яті ЕОМ, і, отже, займає чітко визначену ділянку. Компілятор мови С вимагає попереднього визначення типів абсолютно всіх даних, які використовуються в програмі, для того, щоб визначити спосіб і місце розміщення їх у пам'яті. При не дотриманні цих вимог робота компілятора припиняється. Дані розрізняються числові та текстові. Згадане вище однобайтне представлення цілого числа може бути використане як за прямим призначенням, так і для ідентифікації коду символу текстових даних. Існують стандартні (ASCII, а також декілька альтернативних) таблиці символів, які містять 256 цілих кодів, що відповідають найбільш поширеним текстовим символам. Можна вважати, що текстові дані представляються в пам'яті посимвольно послідовністю цілих однобайтових чисел.

ЗАВДАННЯ

1. Здійснити виконання програми VALUES.C:

```
#include <stdio.h>
#include <conio.h>
#include <windows.h>
int main() {
    SetConsoleCP(65001);
    SetConsoleOutputCP(65001);
    printf("Числа типу int займають %d байт.\n", sizeof(int));
    printf("Числа типу char займають %d байт.\n", sizeof(char));
    printf("Числа типу float займають %d байт.\n", sizeof(float));
    printf("Числа типу double займають %d байт.\n", sizeof(double));
    getch();
}
```

Числа типу int займають 4 байт.

Числа типу char займають 1 байт.

Числа типу float займають 4 байт.

Числа типу double займають 8 байт.

2. Створити і виконати програми дослідження властивостей арифметичних операцій із різними типами величин.

Префіксний та постфіксний інкремент ++ і декремент --

```
#include <stdio.h>
#include <conio.h>
int main()
{ int n = 1;
  printf("n=%d \n",n);
  // n++;
  printf("prefix: ++n=%d\n",++n);
  printf("postfix: n++=%d\n",n++);
  printf("after-postfix: n=%d\n",n);
  // n--;
  printf("prefix: --n=%d\n",--n);
  printf("postfix: n--=%d\n",n--);
  printf("after-postfix: n=%d\n",n);
}
```

n=1

prefix: ++n=2

postfix: n++=2

after-postfix: n=3

prefix: --n=2

postfix: n--=2

after-postfix: n=1

3. Виконати завдання згідно варіанта і пояснити результат при n=1, m=1.

Зразок програми.(Варіант 8)

```
int main() {  
    int n=1,m=1,res1,res2;  
    res1=n++-m;  
    printf("res1=%d\n",res1);  
    res2= m-- *n;  
    printf("res2=%d",res2);  
    return 0;  
}
```

res1=0
res2=2

4. Виконати приклади і пояснити результати

4.1

```
#include <stdio.h>  
int main()  
{ int a, b=3;  
float c;  
c = b%2 + (a = ++b/2) + 1.1;  
printf("a=%d, c=%4.1f\n",a,c); }
```

a=2, c= 4.1

4.2

```
#include <stdio.h>  
int main()  
{  
int x=2,z;  
float y = 2.1;  
z = x++*y + y/x*3;  
printf("x=%d z=%d\n",x,z);}
```

x=3 z=6

4.3

```
#include <stdio.h>  
int main()  
{  
float x = 1.1, y = 0, z;  
int a;  
z = (a=x++)*y + 3*x;  
printf("z=%4.1f\n",z);}
```

z= 6.3