

Міністерство освіти і науки України
Національний університет “Львівська Політехніка”



Лабораторна робота №6

Виконав:
Студент групи АП-11
Білий Анатолій Іванович

Прийняв:
Чайковський І.Б.

Львів – 2024

“Загальна структура програми на мові C”

Мета роботи: ознайомитися із загальною структурою побудови програм на мові C, навчитися використовувати функції введення та виведення даних.

Теоретичні відомості

Програма на мові C складається з однієї або більше функцій і хоча б одна з них повинна називатися **main()**.

Опис функції складається з заголовку та тіла. Заголовок у свою чергу містить директиви препроцесора типу **#include** тощо, що під’єднують бібліотечні файли та специфікують перетворення тексту програми перед компіляцією; а також ім’я функції.

Ознакою імені функції служать круглі дужки. Тіло функції поміщається в фігурні дужки та є набором операторів (команд), кожен із яких закінчується символом “ ; ” - крапка з комою.

Елементом програми є коментар - частина тексту програми для пояснення окремих операторів, що входять до її складу. Коментар не впливає на виконання операторів і записується таким чином : **//текст коментарю** або так: **/* текст коментарю*/** .

В першому випадку коментар має бути єдиним у рядку або в кінці рядка. Другий спосіб дозволяє записувати коментар будь-де в тексті програми

Оголошення змінної задає ім’я та атрибути змінної. Визначення змінної, крім задання імені та її атрибутів, приводить до виділення для неї пам’яті.

Програма може містити довільне число директив, вказівок компілятору, оголошень та визначень. Їх синтаксис розглядатиметься нижче. Порядок появи цих елементів у програмі є важливий.

Хід роботи

1. Ознайомитися з теоретичними відомостями.
2. Здійснити виконання усіх прикладів, представлених у теоретичних відомостях, після чого представити їх скріни та результати їх виконання у звіті.
3. Оформити звіт.

Приклад 1

```
#include<stdio.h>
void main(){
int z,w;
int x = 1;
int y = 2;
z = x + y;
w = y - x;
printf("z = %d, w = %d",z,w);
}
```

z = 3, w = 1

Приклад 2

```
#define PI 3,14159
#include<stdio.h>
#include<windows.h>
void main(){
SetConsoleCP(65001);
SetConsoleOutputCP(65001);
int a = 5;
float b = 23.5;
int c = 31000;
printf("%d метрів тканини коштувало %.2f гривень\n",a,b);
printf("Значення числа PI %f\n",a,b);
printf("IBM сумісні комп'ютери набули широкого розповсюдження.\n");
printf("%c%d",&c);
}
```

5 метрів тканини коштувало 23.50 гривень
Значення числа PI 0.000000
IBM сумісні комп'ютери набули широкого розповсюдження.
&31000

Приклад 3

```
#include<stdio.h>
void main(){
printf("/%d\n",557);
printf("/%10d\n",557);
printf("/%-10d\n",557);
}
```

/557/
/ 557/
/557 /

Приклад 4

```
#include<stdio.h>
void main(){
```

```
printf("%d\n",557);
printf("%o\n",557);
printf("%x\n",557);
printf("%d\n",-557);
}
```

```
557
1055
22d
-557
```

Приклад 5

```
#include<stdio.h>
#include<windows.h>
void main(){
    SetConsoleCP(65001);
    SetConsoleOutputCP(65001);
    int vik;
    char name[30];
    printf("Ваш вік? ");
    scanf("%d",&vik);
    printf("Введіть ваше ім'я ");
    scanf("%s",&name);
    printf("Привіт %s якому %d років",name,vik);
}
```

```
Ваш вік? 18
Введіть ваше ім'я Анатолій
Привіт Анатолій якому 18 років
```

Приклад 6

```
#define STOP '*'
#include<stdio.h>
void main(){
    char ch;
    ch = getchar();
    if (ch != STOP){
        putchar(ch);
        main();}}
}
```

```
-----
dfg
dfg
*
```

Висновок: під час виконання цієї лабораторної роботи я ознайомився із загальною структурою побудови програм на мові C, навчився використовувати функції введення та виведення даних.

Контрольні питання

1. Структура програми на мові C:

Програма на мові C складається з таких основних частин:

- Заголовкові файли (підключення бібліотек через `#include`).
- Функція `main()`, яка є точкою входу в програму.
- Оголошення змінних.
- Інструкції та вирази.
- Інші функції, якщо необхідно.

2. Ідеологія організації операцій уведення-виведення в мові C:

Уведення-виведення в мові C організовано через бібліотеку `stdio.h`, яка забезпечує функції для роботи з потоками даних. Основні ідеї включають роботу з файлами і стандартними потоками (`stdin`, `stdout`, `stderr`) через функції типу `printf()`, `scanf()`, `getchar()`, `putchar()`, `fopen()`, `fclose()`, `fread()`, `fwrite()`.

3. Синтаксис функцій `printf()` і `scanf()`:

- `printf(format, аргумент1, аргумент2, ...)` - використовується для виведення даних на екран.
- `scanf(format, &аргумент1, &аргумент2, ...)` - використовується для введення даних з клавіатури.

4. Основні типи форматів при звертанні до функцій `printf()` і `scanf()`:

- `%d` для цілих чисел.
- `%f` для чисел з плаваючою комою.
- `%c` для символів.
- `%s` для рядків.
- `%x` для шістнадцяткових чисел.

5. Модифікатори форматів при звертанні до функцій `printf()` і `scanf()`:

- `%.nf` для обмеження кількості знаків після коми у числах з плаваючою комою.
- `%ld` для довгих цілих чисел.
- `%lf` для чисел з плаваючою комою типу `double`.
- `%%s` для пропуску заданої кількості символів у рядках.

6. Відмінності при застосуванні функцій `printf()` і `scanf()`:

- `printf()` використовується для виведення даних і може працювати з багатьма аргументами одночасно.
- `scanf()` використовується для введення даних і потребує адреси змінних для запису введених значень.

7. Застосування функцій `getchar()` і `putchar()`:

- `getchar()` зчитує один символ з стандартного потоку введення.
- `putchar(int char)` виводить один символ на стандартний потік виведення.

8. Пояснити зміст і обґрунтувати результати виконаних прикладів:

Приклади програм і використання функцій ``printf()``, ``scanf()``, ``getchar()``, ``putchar()`` демонструють основні операції введення та виведення в мові C. Результати виконання таких прикладів можна пояснити розглядом конкретних значень змінних та форматів введення-виведення. Наприклад, ``printf("Hello, %s!", name);`` виведе "Hello, [ім'я користувача]!" на екран, якщо ``name`` містить відповідне значення. Аналіз результатів допомагає зрозуміти правильність використання форматів і функцій.