

Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут телекомунікацій, радіоелектроніки та електронної техніки кафедра
«Радіоелектронні пристрої та системи»



Звіт з лабораторної роботи №6
з дисципліни «Програмування»

Підготував:

ст. групи АП-11

Фостик Віталій

Прийняла:

Гордійчук-Бублівська О. В.

Тема: Загальна структура програми на мові C

Мета: 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*/`

Завдання:

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.

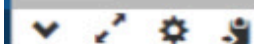
Виконання:

```
1  #include <stdio.h>
2  #include <conio.h>
3  #define STOP '*'
4  void main()
5  {
6      char ch;
7      ch=getchar();
8      m1: if(ch!=STOP){
9          putchar(ch);
10         ch=getchar(); goto m1;
11     }
12 }
13
```



```
..Program finished with exit code 42
Press ENTER to exit console.
```

```
1 #include <stdio.h>
2 void main()
3 {
4     int vik;
5     char name[30];
6     printf("Vash vik?\n");
7     scanf("%d", &vik);
8     printf("Vvedit vashe imya\n");
9     scanf("%s", name);
10    printf("Pryvit %s jakomy(iy) %d rokiv", name, vik);
11 }
12
```



```
Vash vik?
18
Vvedit vashe imya
Vlad
Pryvit Vlad jakomy(iy) 18 rokiv

...Program finished with exit code 0
Press ENTER to exit console.
```

```
1 #include <stdio.h>
2 void main()
3 {
4     printf("%d\n",557);
5     printf("%o\n",557);
6     printf("%x\n",557);
7     printf("%d\n",-557);
8 }
9
```

557
1055
22d
-557

...Program finished with exit code 5
Press ENTER to exit console.

```
1 #include <stdio.h>
2 #define riadok "Чудова погода"
3 void main()
4 {
5     printf("/%2s/\n", riadok);
6     printf("/%15s/\n", riadok);
7 }
8
```

Чудова погода/
/

...Program finished with exit code 18
Press ENTER to exit console.

4

```
1 #include <stdio.h>
2 void main()
3 {
4     printf("/%f/\n", 6543.21);
5     printf("/%e/\n", 6543.21);
6     printf("/%4.2f/\n", 6543.21);
7     printf("/%3.1f/\n", 6543.21);
8     printf("/%10.3f/\n", 6543.21);
9     printf("/%10.3e/\n", 6543.21);
10 }
11
```

6543.210000/
6.543210e+03/
6543.21/
6543.2/
6543.210/
6.543e+03/

...Program finished with exit code 13
Press ENTER to exit console.

5

```
1 #include <stdio.h>
2 void main()
3 {
4     printf("/%8d/\n", 21949);
5     printf("/%8d/\n", -1611);
6     printf("/%8d/\n", -8380);
7     printf("/%8d/\n", 2143);
8 }
9 |
```

▼ ↗ ⚙ 🐞

```
/ 21949/
/ -1611/
/ -8380/
/ 2143/

...Program finished with exit code 11
Press ENTER to exit console. □
```

```
1 #include <stdio.h>
2 void main()
3 {
4     printf("/%d/\n",557);
5     printf("/%10d/\n",557);
6     printf("/%-10d/\n",557);
7 }
8
```

✓ ↗ ⚙ 🗑

```
/557/
/          557/
/557          /

...Program finished with exit code 13
Press ENTER to exit console.
```



```

1 #define PI 3.14159
2 #include <stdio.h>
3 void main()
4 {
5     int a =5;
6     float b=23.5;
7     int c=31000;
8     printf("%d метрів тканини коштувало %f гривень.\n", a,b);
9     printf("Значення числа pi рівне %f.\n", PI);
10    printf("IBM сумісні комп'ютери набули широкого розповсюдження.\n");
11    printf("%c%d\n", '$', c);
12 }
13

```

5 метрів тканини коштувало 23.500000 гривень.
 значення числа pi рівне 3.141590.
 IBM сумісні комп'ютери набули широкого розповсюдження.
 \$31000

Program finished with exit code 7
 Press ENTER to exit console.

```
1  #include <stdio.h>
2  void main()
3  {
4      int z;
5      int w;
6      int x=1;
7      int y=2;
8      z=y+x;
9      w=y-x;
10     printf("z=%d , w=%d", z,w);
11 }
12
```

z=3 , w=1

..Program finished with exit code 9
Press ENTER to exit console.

Контрольні запитання:

1. Структура програми на мові C:

Програма на мові C зазвичай складається з принаймні одного файлу з розширенням `.c`. Кожна програма має функцію `main()`, яка є точкою входу в програму. Програма може містити інші функції, які можуть бути визвані з `main()` або інших функцій. Основні складові програми на C включають директиви препроцесора (наприклад, `#include`), оголошення змінних, виклики функцій і управляючі конструкції (наприклад, умовні оператори і цикли).

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

Мова C має просту, але потужну систему уведення-виведення. Операції вводу-виводу в мові C базуються на бібліотечних функціях, таких як `printf()` і `scanf()`, які забезпечують форматований вивід і введення відповідно. Користувач може використовувати ці функції для взаємодії з користувачем через консоль.

3. Синтаксис функцій `printf()` і `scanf()` Синтаксис `printf()`: ``c printf("формат_строки", список_аргументів); ``

Синтаксис `scanf()`: ``c

scanf("формат_введення", &змінні); ``

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

- `%d`: цілі числа.
- `%f`: числа з плаваючою комою.
- `%c`: символи. - `%s`: рядки.

5. Модифікатори форматів при звертанні до функцій `printf()` і `scanf()`: - `*`: вказує на ігнорування зчитуваного значення в `scanf()`.

- `l`: для вказівника на `long` ціле або подвійне число.
- `h`: для короткого цілого або короткого беззнакового цілого.

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

- `printf()` використовується для виведення даних на екран, тоді як `scanf()` використовується для отримання введених даних від користувача.
- `printf()` має лише один аргумент - рядок формату, тоді як `scanf()` може мати більше аргументів для зчитування значень у відповідні змінні.

7.- `getchar()`: зчитує один символ зі стандартного вводу. - `putchar()`: виводить один символ у стандартний вивід.

8. Пояснення змісту і обґрунтування результатів виконаних прикладів: Відповідно до ваших потреб, якщо ви надаєте конкретні приклади, я можу допомогти пояснити їх зміст і результати. Ви можете надати додаткові деталі або конкретні сценарії, щоб я зміг краще розібратися у вашому запиті.