**Практична робота № 13** Модифікація власних програм з додаванням нових функцій

**Мета:** навчитися складати програми з власними функціями**.**

**Завдання**

1. Ознайомтеся з теоретичною частиною.
2. Модифікуйте свою програму з лабораторної роботи №12-1 таким чином, щоб дії, які виконують при виборі (в середині оператору case) виконувались окремою функцією, а в операторі case встановіть виклик цієї функції. **Нові функції** беремо з ПР08 та ПР10
3. Результати у вигляді текстового файлу надсилати на електронну адресу викладача [**t.i.lumpova@gmail.com**](mailto:t.i.lumpova@gmail.com)

Файл повинен мати назву в такому форматі:

**ОП+АМ <Номер групи><Номер лабораторної><Прізвище англійською>**

Наприклад, 21-01Ivanov.cpp.

Для відповідей формується текстовий файл з відповідною назвою.

**Тему в заголовку листа записати**

**ОП+АМ <Номер групи>-><Номер лекції / практичної / лабораторної [літера позначення типу роботи L – лекція, P – практична, R – лабораторна]<Прізвище англійською>**

Всі запитання, що виникнуть, надсилайте на електронну адресу викладача, тему в заголовку листа записати

Всі запитання, що виникнуть, надсилайте на електронну адресу викладача, тему в заголовку листа записати

**ОП+АМ-Запитання-<Номер групи>-<Прізвище >**.

.

**Строк відсилки ЛР для ІПЗ-31 14.11.2024**

**ІПЗ-32 16.11.2024**

**ІПЗ-33 15.11.2024**

*Контрольні запитання для самоперевірки*.

1. Що ми називаємо функцією?
2. Які змінні називаються локальними?
3. Як задаються в функціях значення, що повертаються?
4. Які параметри називаються формальними, а які фактичними?
5. Розкажіть про порядок дій, які виконуються під час виклику функції.
6. Чи можна у викликаємій функції змінювати параметри при **виклику функції з передачею значень? Чи вплине така зміна значень параметрів на дії в головній функції, якщо ці параметри потім там використовуються?**
7. **Які правила встановленні для формування списку параметрів, коли** передача даних виконується за замовчуванням?

**ТЕОРЕТИЧНА ЧАСТИНА**

Рядок – це послідовність символів, яка закінчується нульовим байтом *‘\0’*. Дана послідовність символів записується в подвійних лапках.

Послідовності в С++ представляються масивами або покажчиками, тому стандартно програми мовою C++ зберігають рядки як масиви типу *char*. Для оголошення строкової змінної необхідно всередині програми оголосити масив типу *char* з кількістю елементів, достатньою для зберігання необхідного рядка символів.

Головна відмінність між символьними рядками та іншими типами масивів полягає в тому, що C++ визначає останній елемент масиву. Програми мовою C++ представляють кінець символьного рядка за допомогою символу NULL, який в C++ зображується як спеціальний символ '\0', або нульовий байт.

Створення рядків, наприклад, відбувається наступним чином:

char str2[10]="Hello";

Для стандартної роботи з символьними рядками в стилі С++ призначена бібліотека string.h, яка містить наступні основні функції:

– **char \**strcat*(char \*dest, const char \*src);** – дописує рядок *src* в кінець *dest*;

– **char \**strncat*(char \*dest, const char \*src, size\_t n);** – дописує не більше *n* початкових символів рядка *src* (або весь *src*, якщо його довжина менше) у кінець *dest*;

– **char \**strchr*(const char \*str, int ch);** – шукає символ *ch* у рядку *str*, починаючи з голови, і повертає його адресу, або NULL якщо символ не знайдено;

– **char \**strrchr*(const char \*str, int ch);** – шукає символ *ch* у рядку *str*, починаючи з хвоста і повертає його адресу, або NULL якщо символ не знайдено;

– **int *strcmp*(const char \*str1, const char \*str2);** – лексикографічне порівняння рядків;

– **int *strncmp*(const char \*str1, const char \*str2, size\_t n);** – лексикографічне порівняння перших *n* байтів рядків;

**– char \**strcpy*(char \*dest, const char \*src);** – копіює рядок з *src* у *dest*;

– **char \**strncpy*(char \*dest, const char \*src, size\_t n);** – копіює до *n* байт рядку з *src* у *dest*;

– **size\_t *strlen*(const char \*str);** – повертає довжину рядка;

– **size\_t *strspn*(const char \*str, const char \*alp);** – визначає максимальну довжину початкового підрядка з *str*, що складається виключно з байтів, перерахованих в *alp*;

– **char \**strpbrk*(const char \*str, const char \*alp);** – знаходить перше входження будь-якого символу, перерахованого в *alp*, у *str*;

– **char \**strstr*(const char \*str, const char \*pstr);** – знаходить перше входження рядка *pstr* у *str*.