**Практична робота 1**

**Бурлаченко Єгор**

**Функції**

**Код задач:**

#include <iostream>

#include <sstream>

using namespace std;

void printNumberWithSpaces(int number) {

if (number >= 1 && number <= 32767) {

stringstream ss;

ss << number;

string numberStr = ss.str();

for (char digit : numberStr) {

cout << digit << " ";

}

cout << endl;

}

else {

cout << "Number is out of the range from 1 to 32767." << endl;

}

}

int printRandomNumber()

{

srand((unsigned)time(NULL));

int random = 100 + (rand() % 101);

return random;

}

int main() {

int number;

cin >> number;

printNumberWithSpaces(number);

printRandomNumber();

return 0;

}

**Контрольні запитання:**

1. Що називають функцією? Що таке опис функції?

- Функція - це набір інструкцій, які виконують певну операцію або завдання в програмі. Функції використовуються для організації програм у більш маневровий та читабельний спосіб.

- Опис функції включає в себе ім'я функції, тип повернення (якщо є), список параметрів, та тіло функції, яке містить інструкції для виконання.

2. Що таке прототип функції? Де його розміщують у програмі?

- Прототип функції - це опис функції без її фактичного визначення. Прототипи функцій розміщують у верхній частині програми або в окремому файлі заголовка, щоб інші функції або частини програми могли знати про існування цієї функції.

3. Які символи оточують тіло функції? Які символи оточують аргументи функції?

- Тіло функції оточують фігурні дужки `{}`.

- Аргументи функції перераховуються в круглих дужках `()` і розділяються комами.

4. Чи обов’язково в прототипах функцій вказувати ідентифікатори змінних?

- Ні, в прототипах функцій не обов'язково вказувати ідентифікатори змінних. Важливо вказати тип і порядок параметрів, але ідентифікатори змінних можна опускати.

5. Як відбувається звернення до функції?

- Звернення до функції включає виклик функції за її ім'ям з відповідними аргументами. Після виклику функції, виконання програми переходить до визначення функції, виконує там інструкції та повертає результат до місця виклику.

6. Яке ключове слово використовується в заголовку функції, щоб показати, що функція не повертає нічого і немає жодного параметра?

- `void` використовується в заголовку функції для показу, що функція не повертає значення і не приймає жодних параметрів. Наприклад, `void functionName()`.

7. Чи кожна функція повинна мати оператор повернення?

- Ні, не кожна функція повинна мати оператор повернення. Функції, які не повертають значення, мають тип `void` і не мають оператора `return`. Функція, яка повертає значення, повинна мати оператор `return`.

8. Назвіть специфікатори класу пам’яті.

- До специфікаторів класу пам'яті входять `auto`, `register`, `static`, та `extern`.

9. Що таке локальні змінні? Що означає специфікатор пам’яті `auto`?

- Локальні змінні - це змінні, оголошені в межах блоку, функції або методу, і вони видимі лише в межах цього контексту.

- Специфікатор пам'яті `auto` вказує, що змінна автоматично живе та ініціалізується при вході в область видимості і знищується при виході з неї.

10. Що таке глобальні змінні? Що означає специфікатор пам’яті `extern`?

- Глобальні змінні - це змінні, оголошені поза будь-якими функціями або методами, і вони доступні для всіх частин програми.

- Специфікатор пам'яті `extern` вказує, що змінна не визначена в поточному файлі, і вона використовується з зовнішнього файлу або модулю.

11. Що означає специфікатор пам’яті `static`? Чи можуть статичні змінні бути локальними?

- Специфікатор пам'яті `static` вказує, що змінна зберігається на протязі всього виконання програми і її значення зберігається між викликами функції.

- Статичні змінні можуть бути як глобальними, так і локальними. У випадку локальних статичних змінних, вони обмежені областю видимості функції, але зберігають своє значення між викликами цієї функції.

12. Чи можна у функцію передавати параметри за замовчуванням?

- Так, ви можете вказати значення параметрів за замовчуванням, що означає, що при виклику функції, параметри, які не передаються, приймають значення за замовчуванням, визначені при оголошенні функції.

13. Що таке фактичні параметри функції?

- Фактичні параметри (або аргументи) функції - це значення, які передаються у функцію при її виклику. Вони використовуються всередині функції для обробки і виконання відповідних операцій.

14. Що таке формальні параметри?

- Формальні параметри - це змінні, оголошені у визначенні функції, які приймають значення фактичних параметрів під час виклику функції.

15. Чим відрізняються фактичні параметри від формальних?

- Фактичні параметри - це значення, які передаються у функцію при виклику. Формальні параметри - це змінні у визначенні функції, які приймають ці значення під час виклику.

16. Чи можуть ідентифікатори фактичних і формальних параметрів співпадати?

- Так, ідентифікатори фактичних та формальних параметрів можуть співпадати. Однак формальні параметри мають пріоритет у межах функції.

17. Чи обов’язково кількість фактичних і формальних параметрів повинні співпадати?

- Так, кількість фактичних і формальних параметрів повинна співпадати. В іншому випадку ви отримаєте помилку компіляції.

18. Що таке рекурсивна функція? Навести приклад.

- Рекурсивна функція - це функція, яка викликає сама себе в тілі функції. Рекурсія використовується для розв'язання завдань, які можна розділити на менші аналогічні завдання.

- Приклад рекурсивної функції:

int factorial(int n) {

if (n <= 1) {

return 1;

}

return n \* factorial(n - 1);

}

19. Що означає, що параметри передаються у функцію за значенням?

- Параметри передаються у функцію за значенням означає, що функція отримує копії параметрів, і зміни внутрішньої копії параметрів не впливають на оригінальні значення.

20.Що означає, що параметри передаються у функцію за покажчиком?

- Параметри передаються у функцію за покажчиком означає, що функція отримує адреси (покажчики) параметрів, і зміни внутрішніх копій параметрів впливають на оригінальні значення.

21. Як передати у функцію масив? Навести приклад.

- Масиви передаються в функцію як покажчики на перший елемент масиву. Нижче наведено приклад передачі масиву:

void printArray(int arr[], int size) {

for (int i = 0; i < size; i++) {

std::cout << arr[i] << " ";

}

}

int main() {

int myArray[] = {1, 2, 3, 4, 5};

int size = sizeof(myArray) / sizeof(myArray[0]);

printArray(myArray, size);

return 0;

}

22. Як передати у функцію рядок символів? Навести приклад.

- Рядки символів у C++ передаються як масиви символів (строки C-style). Ось приклад передачі рядка символів:

void printString(const char\* str) {

std::cout << str;

int main() {

const char\* myString = "Hello, World!";

printString(myString);

return 0;

}

23. Що означають параметри головної функції програми?

- Параметри головної функції програми (`main`) представляють аргументи, які можна передати програмі при запуску з командного рядка. Зазвичай, це масив параметрів та їх кількість, які можна використовувати для налаштування програми.

24. Чи може глобальна змінна бути розташована у тілі програми?

- Глобальна змінна розташовується поза функціями та класами у програмі, і вона доступна в усьому файлі. Вона не може бути розташована в тілі програми, оскільки глобальні змінні повинні бути оголошені на глобальному рівні, поза функціями.

25. Чи можна у середині однієї функції оголошувати іншу функцію?

- Так, ви можете оголосити одну функцію всередині іншої функції, і ця функція буде локальною для зовнішньої функції.

26. Що таке вбудовані функції? Коли вони використовуються?

- Вбудовані функції - це функції, які вже включені в стандартну бібліотеку C++ та можуть використовуватися без додаткового оголошення або включення. Наприклад, `printf`, `scanf`, `strlen`, тощо. Вони використовуються для виконання різних операцій, таких як вивід на екран, введення, обробка рядків тощо.