

Міністерством освіти та науки України
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій
Кафедра « Основи програмування »

Звіт

Про виконання лабораторної роботи № 3
« МОВА ПРОГРАМУВАННЯ CІ, КОМПІЛЯЦІЯ, ФУНКЦІЇ ВВОДУ/ВИВОДУ »

виконав

Студент групи ФЕП-11с

Гарбіч Данило

Перевірено:

Асистент Кужій Ю. І.

Львів 2024 р.

1. Написати програму на мові програмування Cі, котра ілюструє роботу функції printf() з використанням основних специфікаторів. Вивести ціле число у десятковому, двійковому, вісімковому і шістнадцятковому форматах. Вивести дійсне число (з дробовою частиною) у формі з плаваючою комою, в експоненційній формі і в гнучкій формі (специфікатор g). Вивести символ, стрічку і вказівник.
2. Написати програму котра реалізує введення кількох стрічок символів та виведення їх у складі форматowanego тексту (наприклад, таблиця списку групи з колонками: No п/п; Прізвище, Ініціали; Ел.пошта; Улюблений колір тощо).

https://github.com/Harbich-Danulo/Labs_OP/tree/main

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
int main() {
```

```
    // Example of using basic format specifiers
```

```
    int num = 255;
```

```
    float fnum = 123.456;
```

```
    char character = 'A';
```

```
    const char* str = "Hello, World!";
```

```
    void* ptr = &num;
```

```
// Output integer in various formats
```

```
printf("Decimal format: %d\n", num);
```

```
printf("Octal format: %o\n", num);
```

```
printf("Hexadecimal format: %x\n", num);
```

```
// Binary format (manually printed)
```

```
printf("Binary format: ");
```

```
for (int i = sizeof(num) * 8 - 1; i >= 0; i--) {
```

```
    printf("%d", (num >> i) & 1);
```

```
}
```

```
printf("\n");
```

```
// Output floating-point number in various formats
```

```
printf("Floating-point format: %f\n", fnum);
```

```
printf("Exponential format: %e\n", fnum);
```

```
printf("Flexible format: %g\n", fnum);
```

```
// Output character, string, and pointer
```

```
printf("Character: %c\n", character);
```

```
printf("String: %s\n", str);
```

```
printf("Pointer: %p\n", ptr);
```

```
// Input and output of multiple strings (table)
```

```
int n;
```

```
printf("\nEnter the number of students: ");
```

```
scanf_s("%d", &n); // scanf_s used here
```

```
getchar(); // Reads the leftover newline after entering the number
```

```

// Dynamically allocate memory for the arrays
char** names = (char**)malloc(n * sizeof(char*));
char** emails = (char**)malloc(n * sizeof(char*));

for (int i = 0; i < n; i++) {
    names[i] = (char*)malloc(50 * sizeof(char));
    emails[i] = (char*)malloc(50 * sizeof(char));

    printf("\nStudent %d\n", i + 1);
    printf("Enter name and initials: ");
    fgets(names[i], 50, stdin);

    printf("Enter email: ");
    fgets(emails[i], 50, stdin);

}

// Output formatted table
printf("\n%-5s %-30s %-30s\n", "No", "Name, Initials", "Email");
printf("-----\n");

for (int i = 0; i < n; i++) {
    // Remove newline after input
    names[i][strcspn(names[i], "\n")] = 0;
    emails[i][strcspn(emails[i], "\n")] = 0;
}

```

```
printf("%-5d %-30s %-30s\n", i + 1, names[i], emails[i]);
```

```
// Free allocated memory for each student entry
```

```
free(names[i]);
```

```
free(emails[i]);
```

```
}
```

```
// Free the main arrays
```

```
free(names);
```

```
free(emails);
```

```
}
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  int main() {
6      int num = 255;
7      float fnum = 123.456;
8      char character = 'A';
9      const char* str = "Hello, World!";
10     void* ptr = &num;
11
12     // Output integer in various formats
13     printf("Decimal format: %d\n", num);
14     printf("Octal format: %o\n", num);
15     printf("Hexadecimal format: %x\n", num);
16
17     printf("Binary format: ");
18     for (int i = sizeof(num) * 8 - 1; i >= 0; i--) {
19         printf("%d", (num >> i) & 1);
20     }
21     printf("\n");
22
23     printf("Floating-point format: %f\n", fnum);
24     printf("Exponential format: %e\n", fnum);
25     printf("Flexible format: %g\n", fnum);
26
27     printf("Character: %c\n", character);
28     printf("String: %s\n", str);
29     printf("Pointer: %p\n", ptr);
30
31     int n;
32     printf("\nEnter the number of students: ");
33     scanf_s("%d", &n);
34     getchar();
35
36     char** names = (char**)malloc(n * sizeof(char*));
37     char** emails = (char**)malloc(n * sizeof(char*));
```

```

35
36 char** names = (char**)malloc(n * sizeof(char*));
37 char** emails = (char**)malloc(n * sizeof(char*));
38
39 for (int i = 0; i < n; i++) {
40     names[i] = (char*)malloc(50 * sizeof(char));
41     emails[i] = (char*)malloc(50 * sizeof(char));
42
43
44     printf("\nStudent %d\n", i + 1);
45     printf("Enter name and initials: ");
46     fgets(names[i], 50, stdin);
47
48     printf("Enter email: ");
49     fgets(emails[i], 50, stdin);
50
51
52 }
53
54 printf("\n%-5s %-30s %-30s\n", "Number", "Name, Initials", "Email");
55 printf("-----\n");
56
57 for (int i = 0; i < n; i++) {
58     names[i][strcspn(names[i], "\n")] = 0;
59     emails[i][strcspn(emails[i], "\n")] = 0;
60
61
62     printf("%-5d %-30s %-30s\n", i + 1, names[i], emails[i]);
63
64     free(names[i]);
65     free(emails[i]);
66
67 }
68
69 free(names);
70 free(emails);

```

Консоль отладки Microsoft Visual Studio

Floating-point format: 123.456001

Exponential format: 1.234560e+02

Flexible format: 123.456

Character: A

String: Hello, World!

Pointer: 000000CB1C2FF974

Enter the number of students: 3

Student 1

Enter name and initials: one

Enter email: one.email

Student 2

Enter name and initials: two

Enter email: two.email

Student 3

Enter name and initials: three

Enter email: three.email

Number Name, Initials

Email

1	one	one.email
2	two	two.email
3	three	three.email