

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут телекомунікації, радіоелектроніки та електронної техніки

кафедра «Радіоелектронні пристрої та системи»
З дисципліни «Програмування частина 2»
Лабораторна робота №10-11
«Умовні оператори та оператори переходу у мові С»

Мета роботи: навчитися використовувати умовні оператори та оператори переходу під час програмування на мові С.

Підготував
ст.групи АП-11
Василюк Ростислав

Прийняв:
Чайковський І.Б.

Львів 2024р

\

Теоретичні відомості:

Для розробки алгоритмів розв'язку задач (якщо задачі не зовсім примітивні) потрібна умовна конструкція, або розгалуження: в деякій точці алгоритму перевіряється певна умова, і в залежності від неї обирається один з двох можливих подальших шляхів. Оператор `if` у загальному випадку має наступний вигляд: Слова `if` та `else` є ключовими словами мови C, тобто вони не можуть використовуватися для інших цілей, крім умовного оператора (наприклад, не можна називати так змінні або функції).

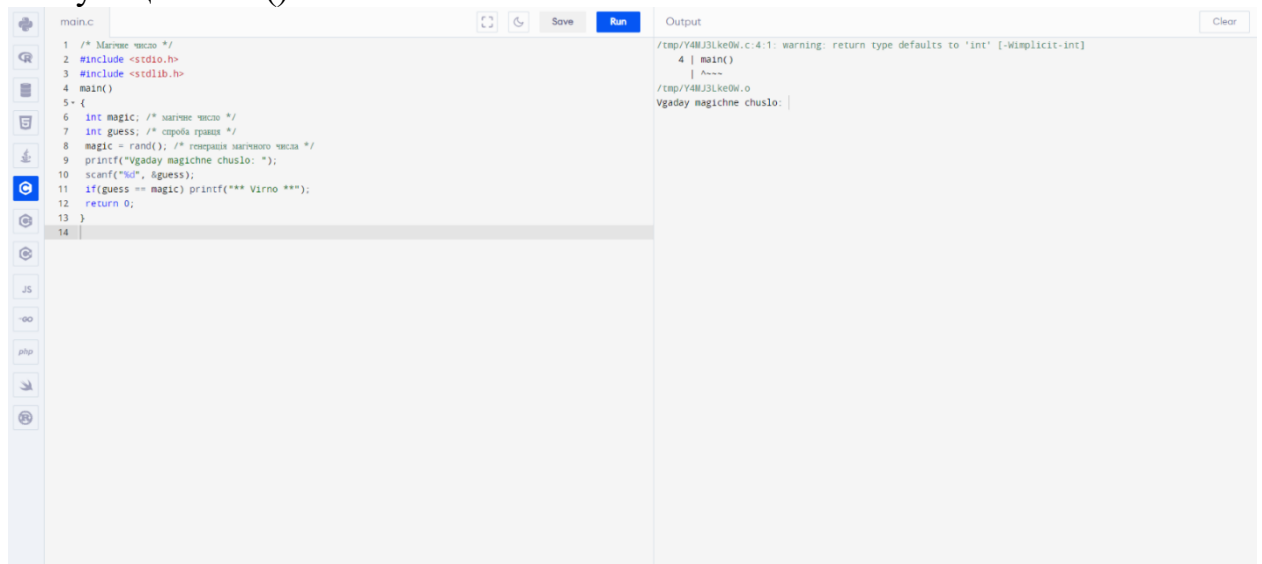
Виконання умовного оператора відбувається таким чином. Спочатку обчислюється значення виразу в дужках. Якщо це значення є логічною істиною (не дорівнює 0), то виконується перший оператор, в протилежному випадку (логічна хиба, число 0) виконується другий оператор.

Обидві гілки умовного оператора (як слово `if`, так і слово `else`) зв'язують лише один наступний оператор. Якщо ж треба включити до умовної конструкції не один, а кілька операторів, то їх треба об'єднати у блок, взявши у фігурні дужки `{ . . . }`.

Нерідко також виникає потреба виконати деякі дії, якщо умова виконується, але не робити нічого (тобто переходити далі до виконання наступного оператора), якщо вона не виконується.

Хід роботи

2. Функція `rand()` оголошена в `<stdlib.h>`



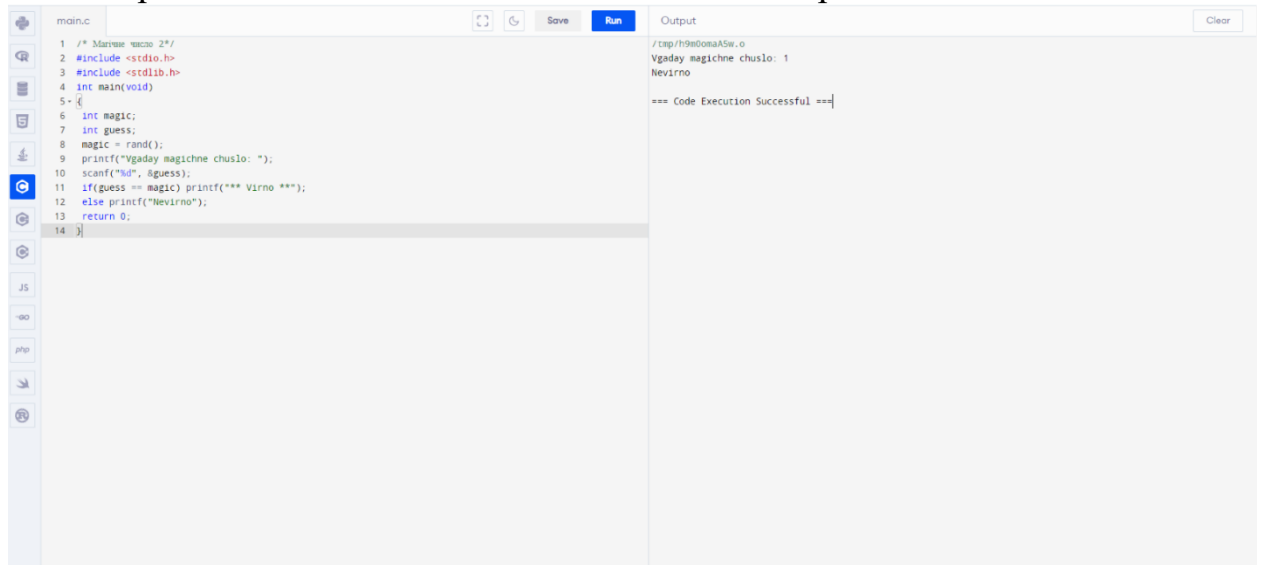
The screenshot shows a C program in a code editor. The code is as follows:

```
1 /* Математичне число */
2 #include <stdio.h>
3 #include <stdlib.h>
4 main()
5 {
6     int magic; /* математичне число */
7     int guess; /* спроба границі */
8     magic = rand(); /* генерація математичного числа */
9     printf("Vgaday magicne chuslo: ");
10    scanf("%d", &guess);
11    if(guess == magic) printf("*** Varno ***");
12    return 0;
13 }
14
```

The output window shows the following text:

```
/tmp/Y4Mj3lke0W.c:4:1: warning: return type defaults to 'int' [-Wimplicit-int]
4 | main()
  | ^~~~~
/tmp/Y4Mj3lke0W.o
Vgaday magicne chuslo: |
```

У цій версії виводиться додаткове повідомлення в разі помилкової відповіді



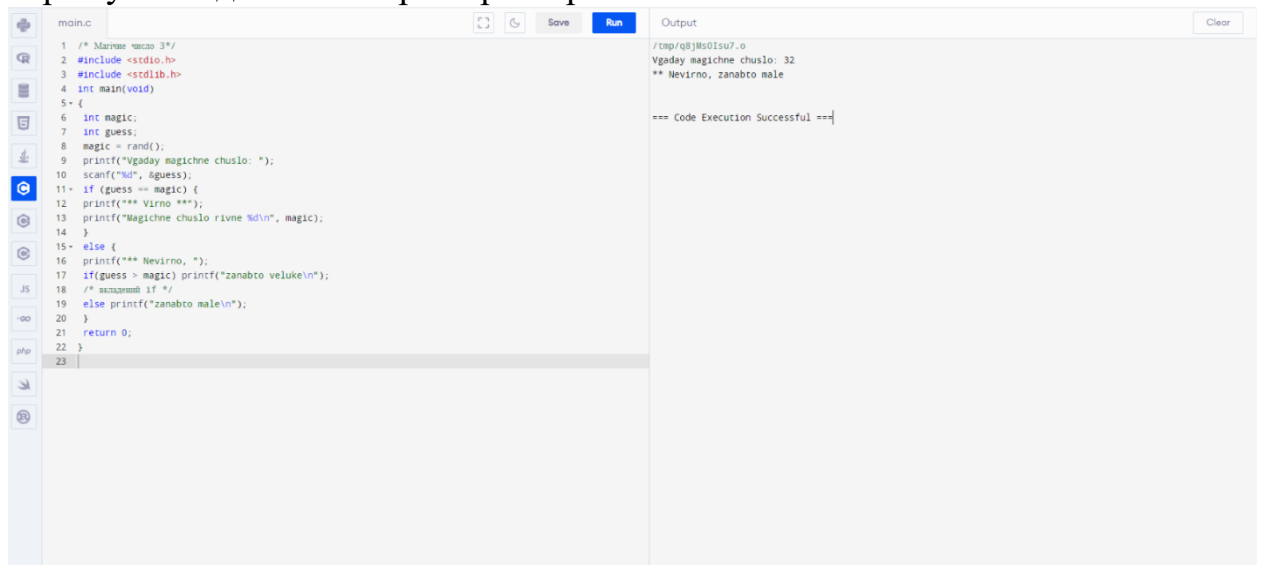
```
1 /* Marvinne chuslo 2 */
2 #include <stdio.h>
3 #include <stdlib.h>
4 int main(void)
5 {
6     int magic;
7     int guess;
8     magic = rand();
9     printf("Vgaday magichne chuslo: ");
10    scanf("%d", &guess);
11    if(guess == magic) printf("*** Virno ***");
12    else printf("Nevirno");
13    return 0;
14 }
```

Output

```
/tmp/h9e0naASw.o
Vgaday magichne chuslo: 1
Nevirno

=== Code Execution Successful ===
```

У наступному прикладі вкладений оператор if використовується в модернізованій програмі для гри в магічне число. З його допомогою гравець отримує повідомлення про характер помилки:



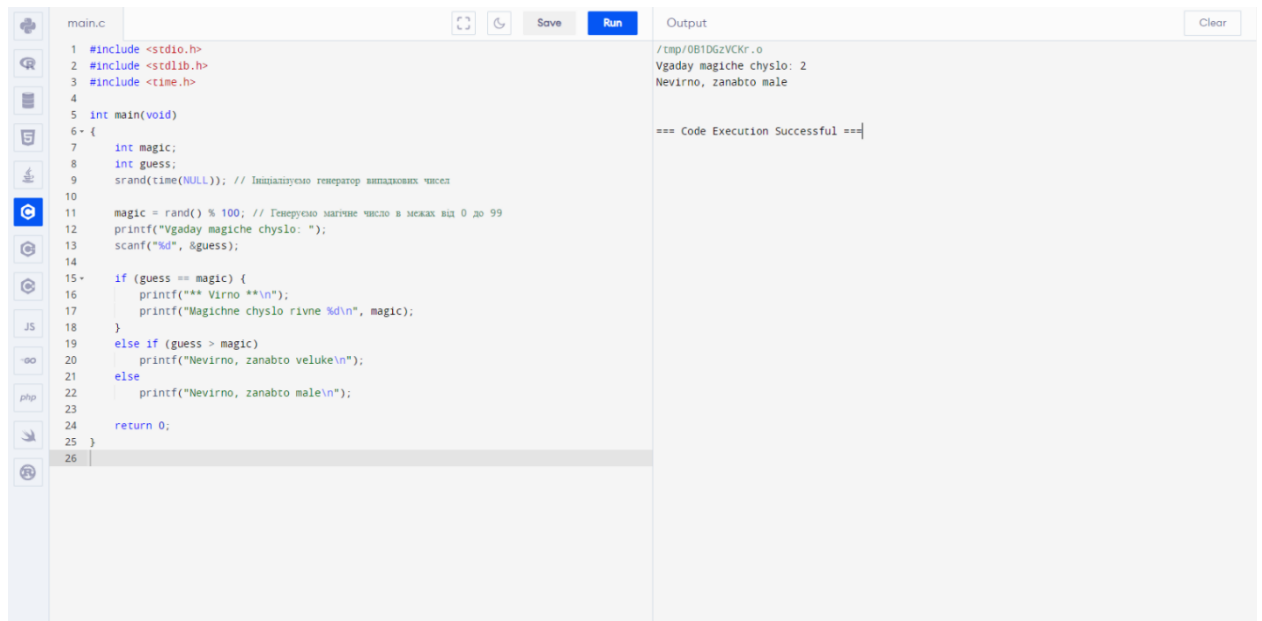
```
1 /* Marvinne chuslo 3 */
2 #include <stdio.h>
3 #include <stdlib.h>
4 int main(void)
5 {
6     int magic;
7     int guess;
8     magic = rand();
9     printf("Vgaday magichne chuslo: ");
10    scanf("%d", &guess);
11    if(guess == magic) {
12        printf("*** Virno ***");
13        printf("Magichne chuslo rivne %d\n", magic);
14    }
15    else {
16        printf("*** Nevirno, ");
17        if(guess > magic) printf("zanabto veluke\n");
18        /* zazachem if */
19        else printf("zanabto male\n");
20    }
21    return 0;
22 }
23
```

Output

```
/tmp/q8j8u0isu7.o
Vgaday magichne chuslo: 32
** Nevirno, zanabto male

=== Code Execution Successful ===
```

Використовуючи сходи if-else-if, програму для гри в "магічне число" можна записати так:



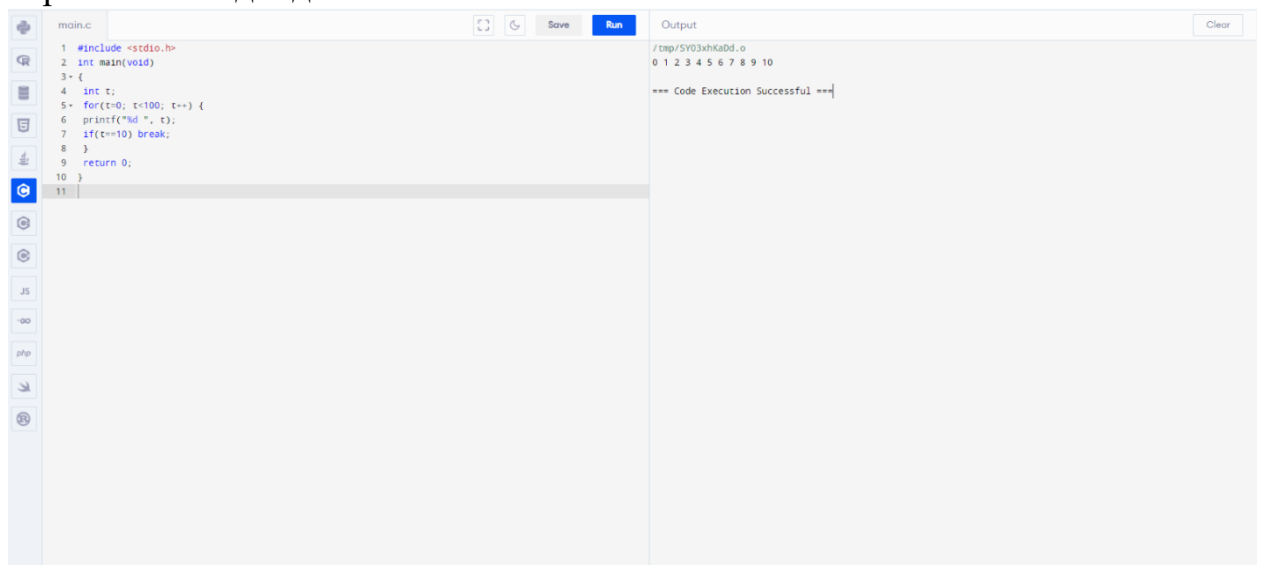
```
main.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main(void)
6 {
7     int magic;
8     int guess;
9     srand(time(NULL)); // Ініціалізуємо генератор випадкових чисел
10
11     magic = rand() % 100; // Генеруємо магічне число в межах від 0 до 99
12     printf("Vgaday magiche chyslo: ");
13     scanf("%d", &guess);
14
15     if (guess == magic) {
16         printf("*** Virno **\n");
17         printf("Magichne chyslo rivne %d\n", magic);
18     }
19     else if (guess > magic)
20         printf("Nevirno, zanabto veluke\n");
21     else
22         printf("Nevirno, zanabto male\n");
23
24     return 0;
25 }
26
```

Output

```
/tmp/0B1DGzVCKr.o
Vgaday magiche chyslo: 2
Nevirno, zanabto male

=== Code Execution Successful ===
```

Коли всередині циклу зустрічається оператор break, виконання циклу безумовно (тобто без перевірки будь-яких умов) зупиняється і керування передається оператору, наступному за ним. Наприклад, програма виводить на екран числа від 0 до 10.



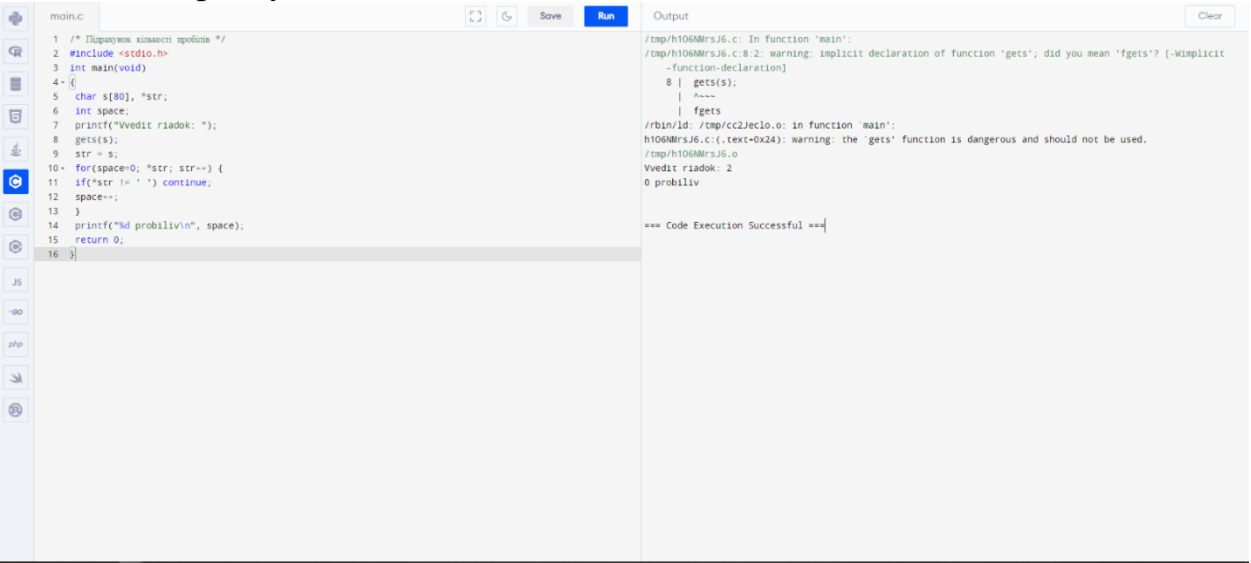
```
main.c
1 #include <stdio.h>
2 int main(void)
3 {
4     int t;
5     for(t=0; t<100; t++) {
6         printf("%d ", t);
7         if(t==10) break;
8     }
9     return 0;
10 }
11
```

Output

```
/tmp/SY03xhKaDd.o
0 1 2 3 4 5 6 7 8 9 10

=== Code Execution Successful ===
```

У наступному прикладі програма підраховує кількість пробілів в рядку, введеним користувачем:



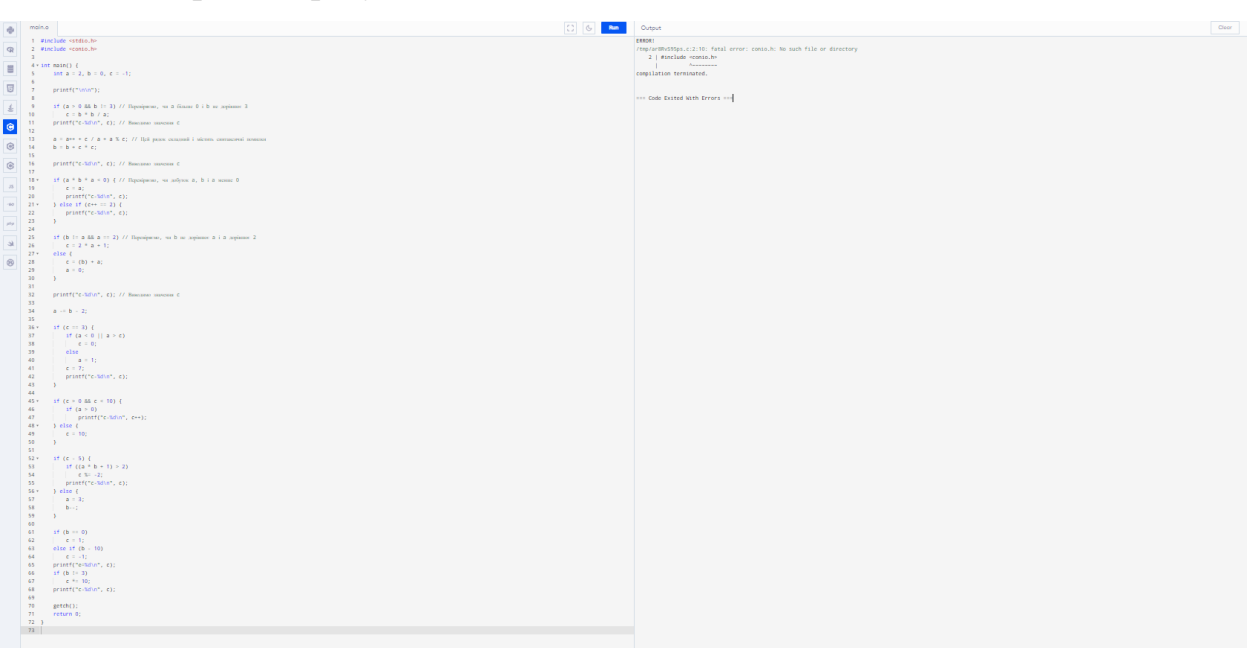
```
main.c
1  /* Підрахунок кількості пробілів */
2  #include <stdio.h>
3  int main(void)
4  {
5      char s[80], *str;
6      int space;
7      printf("Введіть рядок: ");
8      gets(s);
9      str = s;
10     for(space=0; *str; str++) {
11         if(*str == ' ') continue;
12         space++;
13     }
14     printf("Кількість пробілів: %d", space);
15     return 0;
16 }
```

Output

```
/tmp/h106MrsJ6.c: In function 'main':
/tmp/h106MrsJ6.c:8:2: warning: implicit declaration of function 'gets'; did you mean 'fgets'? [-Wimplicit-function-declaration]
8 | gets(s);
  | ^~~~~
  | fgets
/rbin/ld: /tmp/cc2Jeclo.o: in function 'main':
h106MrsJ6.c:(.text+0x24): warning: the 'gets' function is dangerous and should not be used.
/tmp/h106MrsJ6.o
Введіть рядок: 2
0 пробілів

=== Code Execution Successful ===
```

3. Виконати програму, яка ілюструє розгалуження умовними операторами та пояснити отримані результати:



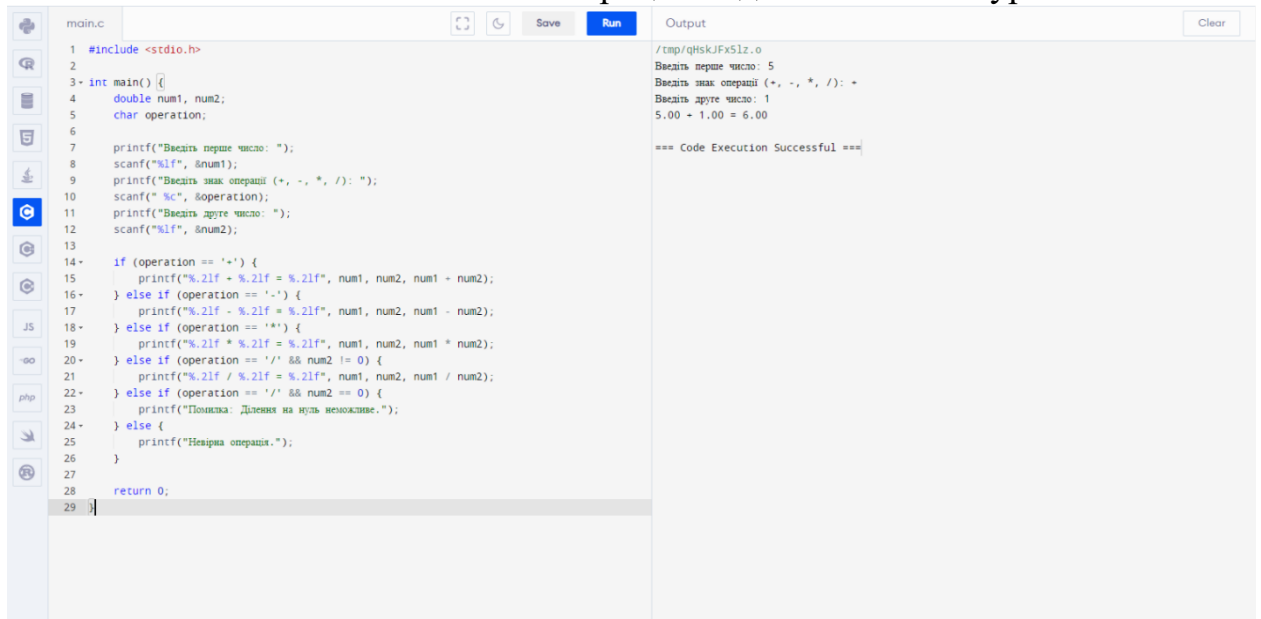
```
main.c
1  #include <stdio.h>
2  #include <math.h>
3
4  int main() {
5      int a, b, c;
6      printf("Введіть a: ");
7      scanf("%d", &a);
8      printf("Введіть b: ");
9      scanf("%d", &b);
10     printf("Введіть c: ");
11     scanf("%d", &c);
12
13     // Перевірка, чи a більше b і b більше c
14     if (a > b && b > c) {
15         printf("a > b > c");
16     }
17
18     // Перевірка, чи a більше b, або b більше c, або c більше a
19     if (a > b || b > c || c > a) {
20         printf("a > b || b > c || c > a");
21     }
22
23     // Перевірка, чи a більше b і c, або b більше a і c, або c більше a і b
24     if (a > b && c > b || b > a && c > a || c > a && b > a) {
25         printf("a > b && c > b || b > a && c > a || c > a && b > a");
26     }
27
28     // Перевірка, чи a більше b і c, або b більше a і c, або c більше a і b
29     if (a > b && c > b || b > a && c > a || c > a && b > a) {
30         printf("a > b && c > b || b > a && c > a || c > a && b > a");
31     }
32
33     // Перевірка, чи a більше b і c, або b більше a і c, або c більше a і b
34     if (a > b && c > b || b > a && c > a || c > a && b > a) {
35         printf("a > b && c > b || b > a && c > a || c > a && b > a");
36     }
37
38     // Перевірка, чи a більше b і c, або b більше a і c, або c більше a і b
39     if (a > b && c > b || b > a && c > a || c > a && b > a) {
40         printf("a > b && c > b || b > a && c > a || c > a && b > a");
41     }
42
43     // Перевірка, чи a більше b і c, або b більше a і c, або c більше a і b
44     if (a > b && c > b || b > a && c > a || c > a && b > a) {
45         printf("a > b && c > b || b > a && c > a || c > a && b > a");
46     }
47
48     // Перевірка, чи a більше b і c, або b більше a і c, або c більше a і b
49     if (a > b && c > b || b > a && c > a || c > a && b > a) {
50         printf("a > b && c > b || b > a && c > a || c > a && b > a");
51     }
52
53     // Перевірка, чи a більше b і c, або b більше a і c, або c більше a і b
54     if (a > b && c > b || b > a && c > a || c > a && b > a) {
55         printf("a > b && c > b || b > a && c > a || c > a && b > a");
56     }
57
58     // Перевірка, чи a більше b і c, або b більше a і c, або c більше a і b
59     if (a > b && c > b || b > a && c > a || c > a && b > a) {
60         printf("a > b && c > b || b > a && c > a || c > a && b > a");
61     }
62
63     // Перевірка, чи a більше b і c, або b більше a і c, або c більше a і b
64     if (a > b && c > b || b > a && c > a || c > a && b > a) {
65         printf("a > b && c > b || b > a && c > a || c > a && b > a");
66     }
67
68     // Перевірка, чи a більше b і c, або b більше a і c, або c більше a і b
69     if (a > b && c > b || b > a && c > a || c > a && b > a) {
70         printf("a > b && c > b || b > a && c > a || c > a && b > a");
71     }
72 }
```

Output

```
main.c:2:10: fatal error: math.h: No such file or directory
2 | #include <math.h>
  |          ^~~~~~
compilation terminated.

=== Code Ended with Errors ===
```

4. Написати програму для здійснення базових арифметичних операцій (додавання, віднімання, множення, ділення) над двома числами, використовуючи 13 умовний оператор if. Врахувати, що на нуль ділити неможна. Значення чисел та знак операції вводяться з клавіатури.



```
main.c
1 #include <stdio.h>
2
3 int main() {
4     double num1, num2;
5     char operation;
6
7     printf("Введіть перше число: ");
8     scanf("%lf", &num1);
9     printf("Введіть знак операції (+, -, *, /): ");
10    scanf(" %c", &operation);
11    printf("Введіть друге число: ");
12    scanf("%lf", &num2);
13
14    if (operation == '+') {
15        printf("%.2lf + %.2lf = %.2lf", num1, num2, num1 + num2);
16    } else if (operation == '-') {
17        printf("%.2lf - %.2lf = %.2lf", num1, num2, num1 - num2);
18    } else if (operation == '*') {
19        printf("%.2lf * %.2lf = %.2lf", num1, num2, num1 * num2);
20    } else if (operation == '/' && num2 != 0) {
21        printf("%.2lf / %.2lf = %.2lf", num1, num2, num1 / num2);
22    } else if (operation == '/' && num2 == 0) {
23        printf("Помилка: Ділення на нуль неможливе.");
24    } else {
25        printf("Невірна операція.");
26    }
27
28    return 0;
29 }
```

Output

```
/tmp/qHskJFx5l2.o
Введіть перше число: 5
Введіть знак операції (+, -, *, /): +
Введіть друге число: 1
5.00 + 1.00 = 6.00

=== Code Execution Successful ===
```

Контрольні питання:

1. Назвіть умовні оператори у мові C.

if ; if-else; switch-case; Тернарний оператор ?:

2. Назвіть оператори переходу у мові C.

break

continue

goto

return

3. Охарактеризуйте умовний оператор if.

Умовний оператор if використовується для виконання певної частини коду, якщо задана умова є істинною (true). Частина else є необов'язковою.

Оператор if може бути вкладеним в інший оператор if для створення більш складних умов.

4. Охарактеризуйте призначення оператора break.

Оператор break використовується для передчасного виходу з циклу (while, for, do-while) або оператора switch. Коли оператор break зустрічається всередині циклу, він припиняє виконання циклу, і управління передається на наступний оператор після циклу. Коли break зустрічається в операторі switch, він виводить управління з поточного case та закінчує виконання оператора switch.

5. Поясніть призначення функції rand.

Функція rand() в мові C використовується для генерації псевдовипадкових чисел. Вона визначена в заголовному файлі stdlib.h. Функція rand() повертає послідовність псевдовипадкових чисел в діапазоні від 0 до RAND_MAX (константа, визначена в stdlib.h).