

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ

до практичних занять з дисципліни

«Основи програмування»

для студентів спеціальності

121 «Інженерія програмного забезпечення»

всіх форм навчання

2017

Методичні вказівки до практичних занять з дисципліни «Основи програмування» для студентів спеціальності 121 «Інженерія програмного забезпечення» всіх форм навчання/ Укладач: Л.Ю. Дейнега. – Запоріжжя: ЗНТУ, 2017. – 47 с.

Укладач: Л. Ю. Дейнега

Рецензент: А.В. Притула

Відповідальний за випуск: Л. Ю. Дейнега

Затверджено на
засіданні кафедри ПЗ
Протокол № 5 від 17.11.16

ЗМІСТ

ВСТУП	4
НАЛАГОДЖЕННЯ ПРОГРАМ	5
Точки переривання	6
Трасування і покрокове налагодження	6
Інспектор	7
Спостерігач (Watches)	8
Трасування програми (Kultin67.cpp)	9
Покрокове виконання програми (Kultin80.cpp)	15
Покрокове виконання програми (Kultin119.cpp)	18
Покрокове виконання програми (Kultin135.cpp)	23
Покрокове виконання програми (Kultin162.cpp)	28
Покрокове виконання програми (Kultin163.cpp)	34
ЛІТЕРАТУРА	47

ВСТУП

Дані методичні вказівки розраховані на студентів першого курсу спеціальності 121 «Інженерія програмного забезпечення» для практичної допомоги при вивченні дисципліни "Основи програмування".

Під час вивчення мов програмування важливо розуміти послідовність роботи програми, яка розробляється. Для цього та для пошуку помилок роботи програми може бути дуже корисним використання відладчика (відлагоджувача). Сучасні середовища розробки містять такі вбудовані засоби.

В даних методичних вказівках продемонстровано роботу відладчика середовища розробки Borland C++. Отримані за наданими прикладами практичні навички роботи будуть корисними при роботі в інших середовищах розробки.

В якості завдань та приладів програм для демонстрації роботи відладчика використані матеріали з підручника : Культин М.Б. C/C++ в задачах и примерах. – СПб.: БХВ – Петербург.

НАЛАГОДЖЕННЯ ПРОГРАМ

Відлагоджувач – це інструмент, який дозволяє виконувати програму рядок за рядком, переглядаючи, як змінюються змінні, області пам'яті і навіть регістри процесора.

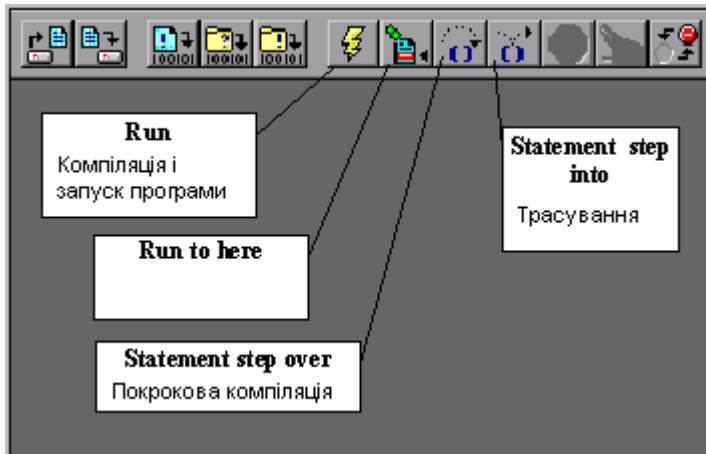


Рисунок 1

У процесі налагодження ви можете користуватися деякими спеціальними засобами.

Breakpoints (Точки переривання). За їх допомогою можна зупиняти відлагоджувач на деякому рядку. Точки переривання використовуються для того, щоб програма виконувалася без зупинки до певного рядка, починаючи з якого потрібно покрокове виконання.

Stepping and tracing (Покрокове виконання і відстежування). Дозволяють виконувати програму рядок за рядком і бачити результат кожної дії, тому можна помітити, коли станеться що-небудь непередбачене.

Inspectors (Інспектори). Виводять на екран значення змінних. Коли компілятор зупиняється в точці переривання або проводить виконання програми рядок за рядком, вікна інспекції дозволяють проглянути значення усіх використовуваних змінних.

Watches (Спостережіачі). Дозволяють виводити на екран значення змінних під час виконання програми рядок за рядком. За допомогою вікон спостереження можна отримати динамічну картину зміни змінних. Вікно спостереження рекомендується розташовувати поряд з вікном редагування.

Точки переривання

Тепер трохи детальніше про ці засоби налагодження. Точка переривання використовується у тому випадку, коли треба зупинитися на певному рядку. Щоб встановити точку переривання у вибраному рядку, клацніть на відступі (області сірого кольору) зліва від цього рядка. Щоб прибрати точку переривання, клацніть там ще раз. З рядка знімається виділення червоним кольором, зникає піктограма зі знаком "стоп". Слід звернути увагу, що клацання правою кнопкою миші викликає контекстне меню, за допомогою якого можна встановлювати і прибрати точки переривання. Якщо ви не пам'ятаєте точно, чи є у вашій програмі точки переривання, можна скористатися командою View→Breakpoint. Вона виводить на екран діалогове вікно, в якому показані усі активні точки переривання.

Трасування і покрокове налагодження

Після того, як буде досягнута точка переривання, можна досліджувати роботу кожного рядка коду. Це робиться двома способами. Перший полягає в запуску програми по одному рядку і називається ***покроковим виконанням***. Якщо в рядку міститься звернення до якої-небудь функції, ця функція виконується, але без зупинки.

Трасування – різновид покрокового виконання. У цьому режимі програма виконується рядок за рядком. Але, якщо під час трасування відбувається звернення до функції, відлагоджувач переходить до першого рядка функції, що викликається.

Трасування використовується в ситуаціях, коли необхідно досліджувати функції, що не входять у ваш початковий код.

Трасування дозволяє переглядати кожен рядок функції як окремий блок.

При покроковому виконанні, як і при трасуванні програми, той рядок, який виконуватиметься наступним, висвічується блакитним, а на полі відступу біля нього з'являється зелена стрілка.

Щоб приступити до покрокового виконання, слід натиснути <F8> чи клацнути на піктограмі (рис. 1). Для трасування, потрібно натиснути <F7> чи клацнути по відповідній кнопці SpeedBar (рис. 1). Для виконання кожного кроку необхідно повторно натискати F7 або F8. Для припинення покрокового виконання натисніть <Ctrl-F2>.

Інспектор

Інспектор – незамінний інструмент покрокового налагодження. Він показує значення змінних і виразів. Тому, якщо вас цікавить, чи правильний результат застосування формули, що зберігається в тій або іншій змінній, або які значення елементів цього масиву, досліджуйте потрібні змінні за допомогою інспектора. Щоб зробити це, необхідно виконати наступне:

1) виберіть в редакторі ім'я об'єкту, встановивши на нього курсор;

2) клацніть правою кнопкою миші, щоб викликати оперативне меню SpeedMenu;

3) виберіть у ньому команду Inspect (Інспектувати).

З'являється інспектор і показує значення об'єкту. Приклад вікна показаний на рис. 2. По мірі виконання програми вікно інспектора оновлюється.

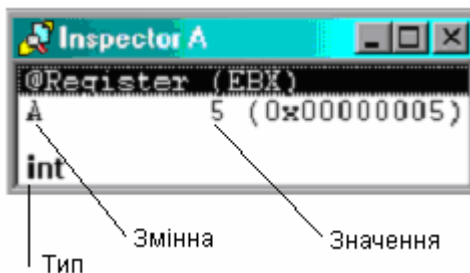


Рисунок 2

Існує можливість зміни змінних під час їх вивчення. Робиться це так.

1. У вікні інспектора клацніть правою кнопкою миші на імені змінної.
2. Виберіть команду Change.
3. Наберіть нове значення.
4. Підтвердіть зміну.

Спостерігач (Watches)

Якщо необхідно подивитися, як змінюються ті або інші елементи при виконанні програми, до кожного з них можна застосувати вікно спостереження. Особливо корисні вікна спостереження, коли необхідно переглядати декілька значень одночасно. Приклад вікна спостереження наведений на рис. 3.

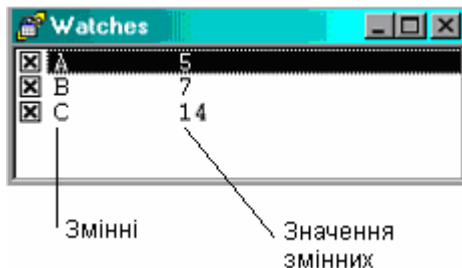


Рисунок 3

Для того, щоб за допомогою вікна спостереження проглянути значення об'єкту, необхідно виконати наступні дії.

1. Виконайте команду View→Watch.
2. Клацніть правою кнопкою по вікну Watches, щоб викликати підменю.
3. Виберіть у ньому Add Watch..., для додавання до списку спостереження нової змінної.

Трасування програми (Kultin67.cpp)

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

char* Rus (const char*);

void main(){
    float r, h, v; //радиус основания, высота и объём
цилиндра
    printf(Rus("Вычисление объёма цилиндра\n"));
    printf(Rus("Введите исходные данные:\n"));
    printf(Rus("Радиус основания (см) -> "));
    scanf("%f", &r);
    printf(Rus("Высота цилиндра (см) -> "));
    scanf("%f", &h);
    v = 2 * 3.1415926 * r * r * h;
    printf(Rus("\nОбъём цилиндра %6.2f куб.см\n"), v);
    printf(Rus("\nДля завершения нажмите <Enter>"),
v);
    getch();
}

char* Rus (const char* text){
    char bufRus[256];
    CharToOem(text, bufRus);
    return bufRus;
}
```

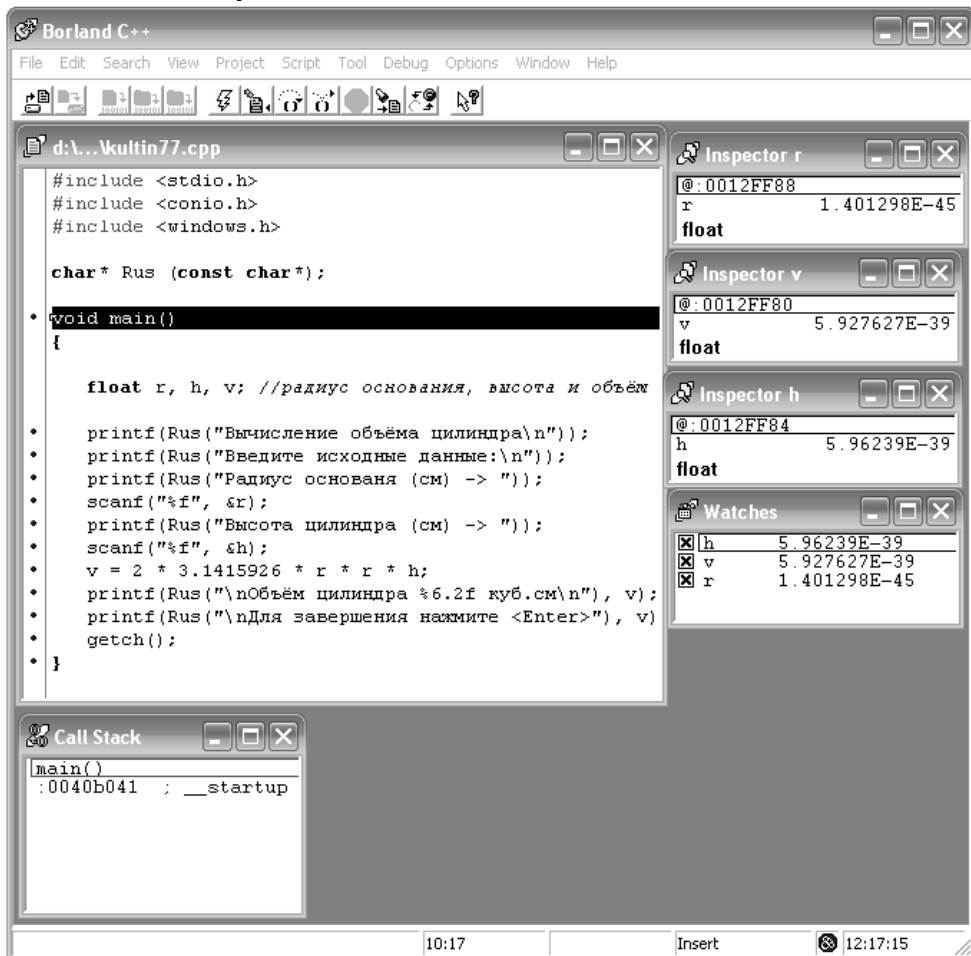
1. Натискаємо клавішу <F7>. У вікні редактора висвічується блакитним рядок *void main ()*. Відкривається консольне вікно. Дослідимо змінні за допомогою інспектора. Для цього для кожної змінної (*r*, *h*, *v*) виконаємо наступне:

1) виберемо в редакторі ім'я змінної, встановивши на неї курсор;

2) натиснемо праву кнопку миші, щоб викликати оперативне меню SpeedMenu;

3) виберемо у ньому команду Inspect (Інспектувати).

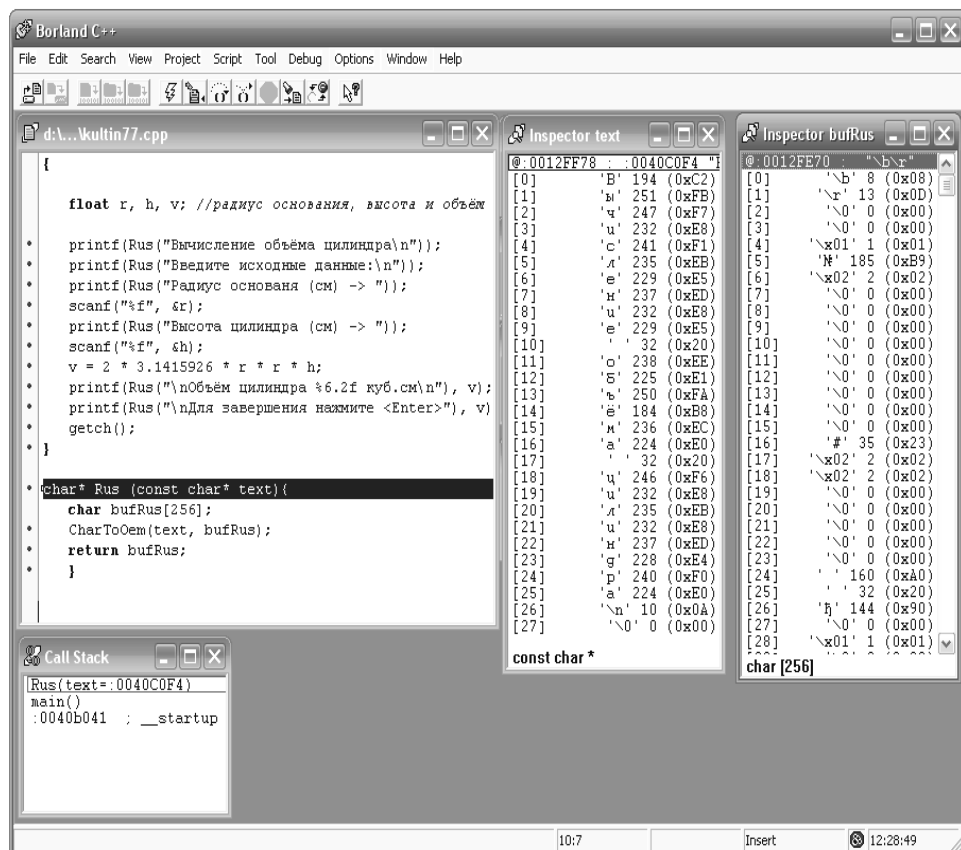
На екран буде виведено вікно інспектора з вмістом змінною, що нас цікавить. Змінні r , h , v містять безглузді значення ("сміття"), оскільки вони не були ініціалізовані.



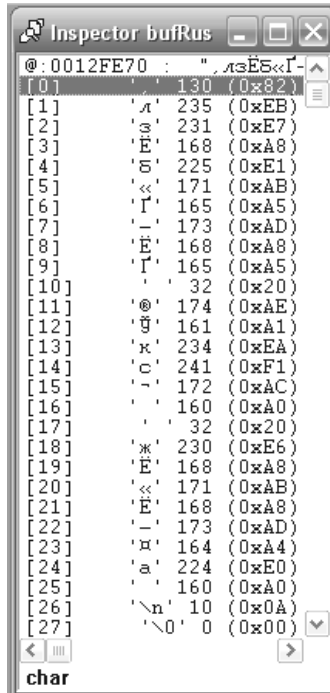
2. Натискаємо клавішу <F7>. У вікні редактора висвічується блакитним рядок `printf(Rus("Вычисление объёма цилиндра\n"))`.

4. Натискаємо клавішу <F7>. У вікні редактора висвічується блакитним рядок *CharToOem(text, bufRus)*; – перший здійснимий оператор у тілі функції. Проінспектуємо змінні *text* і *bufRus*. Оскільки змінні *text* і *bufRus* є масивами, у вікнах інспектора з'являються поточні значення усіх їх елементів. Масив *bufRus* заповнений "сміттям".

4. Натискаємо клавішу <F7>. У вікні редактора висвічується блакитним рядок *CharToOem(text, bufRus)*; – перший здійснимий оператор у тілі функції. Проінспектуємо змінні *text* і *bufRus*. Оскільки змінні *text* і *bufRus* є масивами, у вікнах інспектора з'являються поточні значення усіх їх елементів. Масив *bufRus* заповнений "сміттям".



5. Натискаємо клавішу <F7>. У вікні редактора висвічується блакитним рядок *return bufRus;*. Подивимося, як змінився вміст вікна інспектора з масивом *bufRus*. Масив містить набір символів OEM – перетворений рядок, що міститься у змінній *text*. Набір символів OEM відповідає природному для цієї апаратури набору.



6. Натискаємо клавішу <F7>. У вікні редактора висвічується блакитним рядок *}*.

7. Натискаємо клавішу <F7>. У консольне вікно виводиться текст *Вычисление объёма цилиндра* і курсор переводиться на новий рядок. У вікні редактора висвічується блакитним рядок *printf(Rus("Введите исходные данные:\n"));*.

8-11. Аналогічно пунктам 3–6.

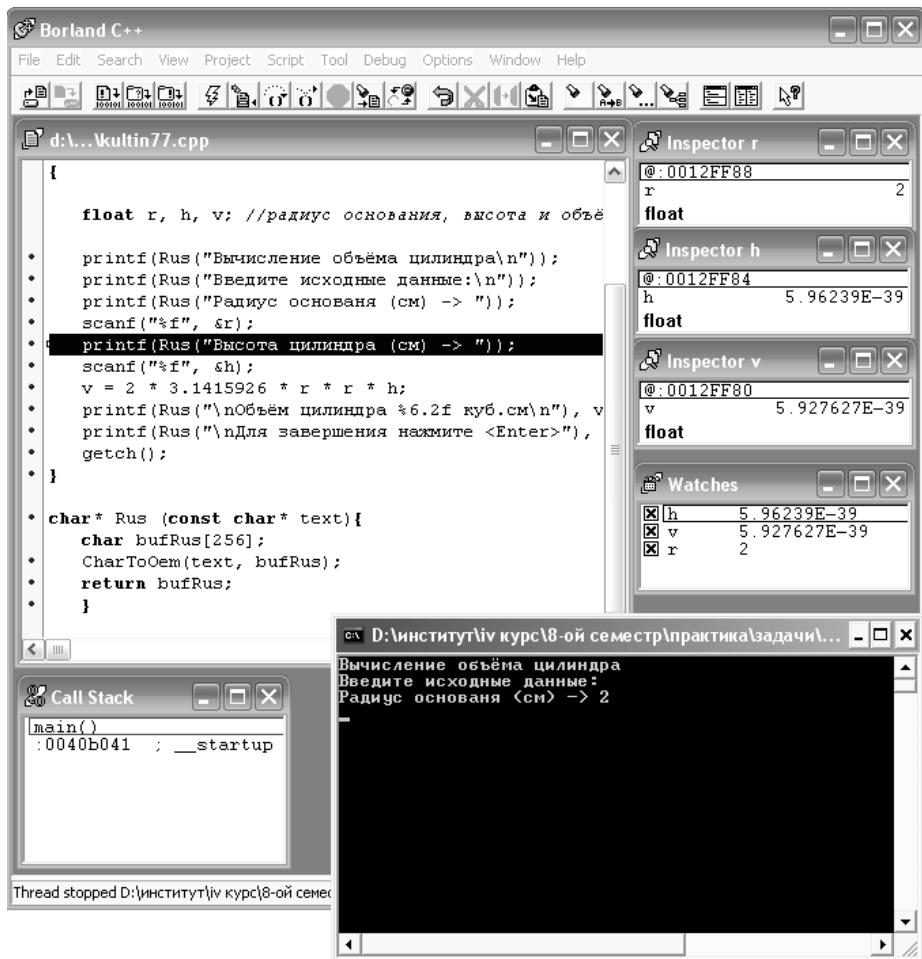
12. Натискаємо клавішу <F7>. У консольне вікно виводиться текст *Введите исходные данные:* і курсор переводиться на новий

рядок. У вікні редактора висвічується блакитним рядок `printf(Rus("Радіус основания (см) ->"));`.

13-16. Аналогічно пунктам 3–6.

17. Натискаємо клавішу <F7>. У консольне вікно виводиться текст *Радіус основания (см) ->*. У вікні редактора висвічується блакитним рядок `scanf("%f", &r);`.

18. Натискаємо клавішу <F7>. На передній план виступає консольне вікно з миготливим курсором. Вводимо 2 і натискаємо <Enter>. Подивимося, як змінився вміст вікна інспектора із змінною *r*: значення змінної стало дорівнювати 2. У вікні редактора висвічується блакитним рядок `printf(Rus("Высота цилиндра (см) -> "));`.



19-22. Аналогічно пунктам 3–6.

23. Натискаємо клавішу <F7>. У консольне вікно виводиться текст *Высота цилиндра (см) ->*. У вікні редактора висвічується блакитним рядок *scanf("%f", &h);*.

24. Натискаємо клавішу <F7>. На передній план виступає консольне вікно з миготливим курсором. Вводимо 10 і натискаємо <Enter>. Подивимося, як змінився вміст вікна інспектора із змінною *h*:

значення змінної стало дорівнювати 10. У вікні редактора висвічується блакитним рядок $v = 2 * 3.1415926 * r * r * h;$.

25. Натискаємо клавішу <F7>. Подивимося, як змінився вміст вікна інспектора із змінною v : значення змінної стало дорівнювати 251.3274.



У вікні редактора висвічується блакитним рядок `printf(Rus("\nОбъём цилиндра %6.2f куб.см\n"), v);`.

26-29. Аналогічно пунктам 3–6.

30. Натискаємо клавішу <F7>. У консольне вікно виводиться текст *Объём цилиндра 251.33 куб.см* і курсор переводиться на новий рядок. У вікні редактора висвічується блакитним рядок `printf(Rus("\nДля завершения нажмите <Enter>"));`.

30-34. Аналогічно пунктам 3–6.

35. Натискаємо клавішу <F7>. У консольне вікно виводиться текст *Для завершения нажмите <Enter>* і курсор переводиться на новий рядок. У вікні редактора висвічується блакитним рядок `getch();`.

36. Натискаємо клавішу <F7>. На передній план виступає консольне вікно з миготливим курсором. Натискаємо <Enter>. Консольне вікно закривається. Трасування програми виконане.

Покрокове виконання програми (Kultin80.cpp)

```
// Проверять, является ли год високосным
#include <stdio.h>
#include <conio.h>
#include <windows.h>

char* Rus (const char*);

void main(){
```

```

int year,
    r;           // остаток от деления на 4

printf(Rus("Введите год, например 2001, и нажмите
<Enter>"));
printf("->");
scanf("%i", &year);
r = year % 4;

if (r)
    printf(Rus("\nГод           не           является
високосным.\n"));
else
    printf(Rus("\nГод является високосным.\n"));

printf(Rus("\nДля завершения нажмите <Enter>"));
getch();
}

char* Rus (const char* text){
    char bufRus[256];
    CharToOem(text, bufRus);
    return bufRus;
}

```

1. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *void main ()*. Відкривається консольне вікно. За допомогою вікна спостереження проглянемо значення змінних *year* і *r*:

- 1) виконаємо команду View→Watch;
- 2) натиснемо праву кнопку миші у вікні Watches, щоб викликати підменю;
- 3) виберемо у ньому Add Watch... для додавання до списку спостереження нової змінної.



Змінні містять безглузді значення ("сміття"), оскільки вони не були ініціалізовані.

2. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок `printf(Rus("Введіть год, наприклад 2001, і натисніть <Enter>"))`;

3. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *Введіть год, наприклад 2001, і натисніть <Enter>*. У вікні редактора висвічується блакитним рядок `printf("->")`;

4. Натискаємо клавішу <F8>. У консольне вікно виводиться *->*. У вікні редактора висвічується блакитним рядок `scanf("%i", &year)`;

5. Натискаємо клавішу <F8>. На передній план виступає консольне вікно з миготливим курсором. Вводимо 2012 і натискаємо <Enter>. У вікні редактора висвічується блакитним рядок `r = year % 4`; Подивимося, як змінився вміст вікна Watches:



6. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:



У вікні редактора висвічується блакитним рядок *if (r)*.

7. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *printf(Rus("\nГод является високосным.\n"))*;

8. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *Год является високосным*. У вікні редактора висвічується блакитним рядок *printf(Rus("\nДля завершения нажмите <Enter>"))*;

9. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *Для завершения нажмите <Enter>*. У вікні редактора висвічується блакитним рядок *getch()*;

10. Натискаємо клавішу <F8>. На передній план виступає консольне вікно з миготливим курсором. Натискаємо <Enter>. У вікні редактора висвічується блакитним рядок *}*.

11. Натискаємо клавішу <F8>. Консольне вікно закривається.

Покрокове виконання програми (Kultin119.cpp)

```
// вычисление среднего арифметического случайных
// последовательностей
#include <stdio.h>
#include <conio.h>
#include <stdlib.h> // для доступа к srand и rand
#include <time.h>
#include <windows.h>

#define L 10 // длина последовательности
#define N 3 // количество последовательностей

char* Rus (const char*);
```

```

void main (){
    int r,           // случайное число
        sum,        // сумма чисел последовательности
        i,j;        // счетчики циклов
    float sred; // среднее арифметическое
    time_t t;   // текущее время - для инициализации

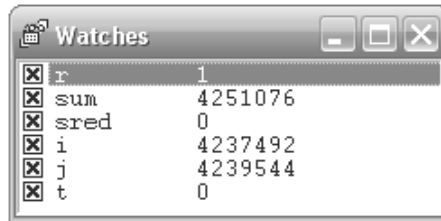
    // генератора случайных чисел
    srand((unsigned) time(&t)); // инициализация
генератора
    // случайных чисел
    for (i = 1; i <= N; i++){
        // генерируем последовательность
        printf(Rus("\nСлучайные числа: "));
        sum=0 ; //не забыть обнулить !
        for (j = 1; j <= L; j++){
            r=rand()%10+1;
            printf("%i ", r);
            sum+=r;
        }
        sred=(float)sum/L; // чтобы не было
усечения
        printf(Rus("\nСред.арифм.: %3.2f\n"), sred);
    }
    printf(Rus("\nДля завершения нажмите <Enter>"));
    getch();
}

char* Rus (const char* text){
    char bufRus[256];
    CharToOem(text, bufRus);
    return bufRus;
}

```

1. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *void main ()*. Відкривається консольне вікно. За

допомогою вікна спостереження переглядатимемо, як змінюються змінні *r*, *sum*, *sred*, *i*, *j* при виконанні програми.

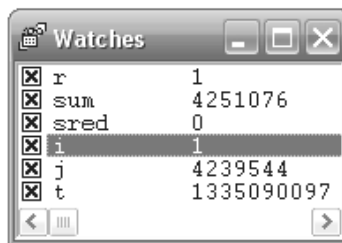


Змінні містять безглузді значення ("сміття"), оскільки вони не були ініціалізовані.

2. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *srand((unsigned) time(&t));*.

3. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *for(i=1; i<=N; i++){*.

4. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:

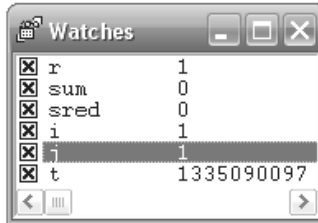


У вікні редактора висвічується блакитним рядок *printf(Rus("\nСлучайные числа: "));*.

5. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *Случайные числа:*. У вікні редактора висвічується блакитним рядок *sum=0;*.

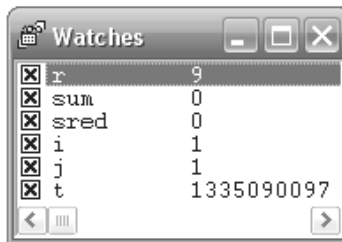
6. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *for(j=1; j<=L; j++){*.

7. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:



У вікні редактора висвічується блакитним рядок `r=rand()%10+1;`.

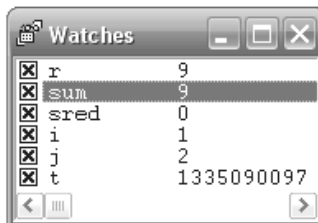
8. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:



У вікні редактора висвічується блакитним рядок `printf("%i ", r);`.

9. Натискаємо клавішу <F8>. У консольне вікно виводиться число 9. У вікні редактора висвічується блакитним рядок `sum+=r;`.

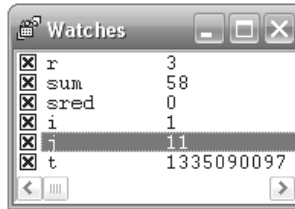
10. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:



У вікні редактора висвічується блакитним рядок `r=rand()%10+1;`.

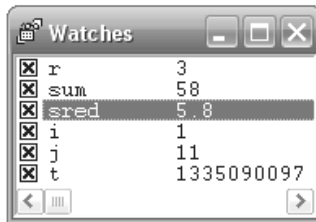
11-36. Аналогічно пунктам 8-10. У консольне вікно виведено усі 10 випадкових чисел першої послідовності в діапазоні від 1 до 10.

37. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:



У вікні редактора висвічується блакитним рядок *sred=(float)sum/L;*.

38. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:

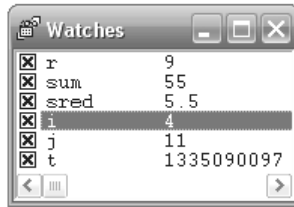


У вікні редактора висвічується блакитним рядок *printf(Rus("\nСред.арифм.: %3.2f\n"), sred);*.

39. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *Сред.арифм.: 5.80*. У вікні редактора висвічується блакитним рядок *printf(Rus("\nСлучайные числа: "));*.

40-107. Аналогічно пунктам 5-39. У консольне вікно виведені: друга послідовність з десяти випадкових чисел в діапазоні від 1 до 10, її середнє арифметичне і третя послідовність.

108. Натискаємо клавішу <F8>. У консольне вікно виводиться середнє арифметичне третьої послідовності. Подивимося, як змінився вміст вікна Watches:



У вікні редактора висвічується блакитним рядок `printf(Rus("\nДля завершення нажміть <Enter>"));`.

109. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *Для завершення нажміть <Enter>*. У вікні редактора висвічується блакитним рядок `getch();`.

110. Натискаємо клавішу <F8>. На передній план виступає консольне вікно з миготливим курсором. Натискаємо <Enter>. У вікні редактора висвічується блакитним рядок `}`.

111. Натискаємо клавішу <F8>. Консольне вікно закривається.

Покрокове виконання програми (Kultin135.cpp)

```
// Определение минимального числа в
// последовательности положительных чисел
#include <stdio.h>
#include <conio.h>
#include <windows.h>

char* Rus (const char*);

void main(){
    int a,           // очередное число
        min;        // минимальное число

    printf(Rus("\nОпределение минимального числа в
последовательности положительных чисел.\n"));
    printf(Rus("Вводите после стрелки числа. Для
завершения введите ноль.\n"));
    printf("-> ");
```

```

scanf("%i", &a);

min = a;      // пусть первое число минимальное
while (a > 0){
    if (a < min)
        min = a;
    printf("-> ");
    scanf("%i", &a);
}

    printf(Rus("Минимальное                число
последовательности: "));
printf("%i", min);

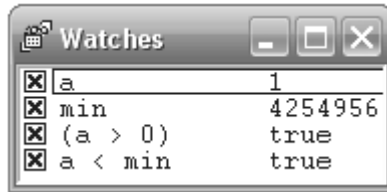
printf(Rus("\nДля завершения нажмите <Enter>"));
getch();
}

char* Rus (const char* text){
    char bufRus[256];
    CharToOem(text, bufRus);
    return bufRus;
}

```

1. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *void main ()*. Відкривається консольне вікно. За допомогою вікна спостереження проглянемо значення змінних *a* і *min* і виразів $(a > 0)$ і $(a < min)$:

- 1) виконаємо команду View→Watch;
- 2) натиснемо праву кнопку миші у вікні Watches, щоб викликати підменю;
- 3) виберемо у ньому Add Watch... для додавання до списку спостереження нової змінної.



Змінні містять безглузді значення ("сміття"), оскільки вони не були ініціалізовані.

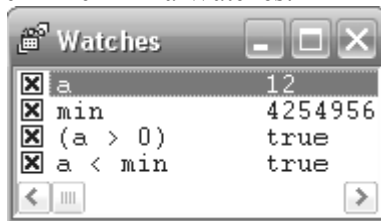
2. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок `printf(Rus("\nОпределение минимального числа в последовательности положительных чисел.\n"));`.

3. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *Определение минимального числа в последовательности положительных чисел.* У вікні редактора висвічується блакитним рядок `printf(Rus("Вводите после стрелки числа. Для завершения введите ноль.\n"));`.

4. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *Вводите после стрелки числа. Для завершения введите ноль.* У вікні редактора висвічується блакитним рядок `printf("-> ");`.

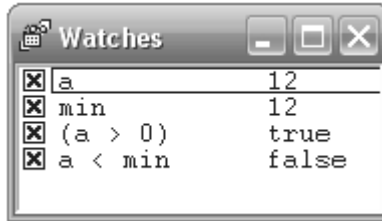
5. Натискаємо клавішу <F8>. У консольне вікно виводиться `->`. У вікні редактора висвічується блакитним рядок `scanf("%i", &a);`.

6. Натискаємо клавішу <F8>. На передній план виступає консольне вікно з миготливим курсором. Вводимо 12 і натискаємо <Enter>. У вікні редактора висвічується блакитним рядок `min = a;`. Подивимося, як змінився вміст вікна Watches:



Змінна `min` як і раніше містить "сміття".

7. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок `while(a>0){`.

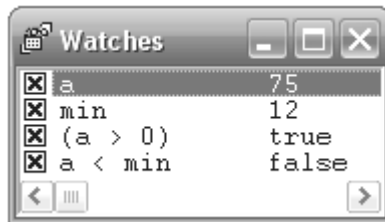


8. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок `if(a<min)`.

9. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок `printf("-> ");`.

10. Натискаємо клавішу <F8>. У консольне вікно виводиться ->. У вікні редактора висвічується блакитним рядок `scanf("%i", &a);`.

11. Натискаємо клавішу <F8>. На передній план виступає консольне вікно з миготливим курсором. Вводимо 75 і натискаємо <Enter>. У вікні редактора висвічується блакитним рядок `while(a>0){`. Подивимося, як змінився вміст вікна Watches :

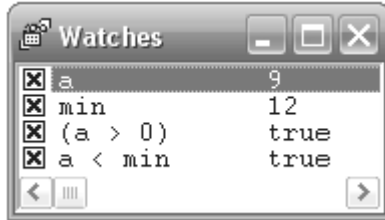


12. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок `if(a<min)`.

13. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок `printf("-> ");`.

14. Натискаємо клавішу <F8>. У консольне вікно виводиться ->. У вікні редактора висвічується блакитним рядок `scanf("%i", &a);`.

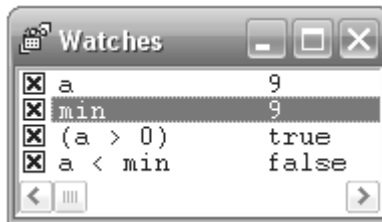
15. Натискаємо клавішу <F8>. На передній план виступає консольне вікно з миготливим курсором. Вводимо 9 і натискаємо <Enter>. У вікні редактора висвічується блакитним рядок `while(a>0){`. Подивимося, як змінився вміст вікна Watches:



16. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *if(a<min)*.

17. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *min=a;*.

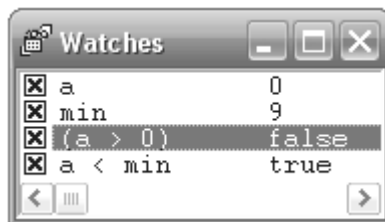
18. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:



У вікні редактора висвічується блакитним рядок *printf("-> ");*.

19. Натискаємо клавішу <F8>. У консольне вікно виводиться ->. У вікні редактора висвічується блакитним рядок *scanf("%i", &a);*.

20. Натискаємо клавішу <F8>. На передній план виступає консольне вікно з миготливим курсором. Вводимо 0 і натискаємо <Enter>. У вікні редактора висвічується блакитним рядок *while(a>0){*. Подивимося, як змінився вміст вікна Watches:



21. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок `printf(Rus("Минимальное число последовательности:"))`;

22. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *Минимальное число последовательности:*. У вікні редактора висвічується блакитним рядок `printf("%i", min)`;

23. Натискаємо клавішу <F8>. У консольне вікно виводиться 9. У вікні редактора висвічується блакитним рядок `printf(Rus("\nДля завершения нажмиме <Enter>"))`;

24. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *Для завершения нажмиме <Enter>*. У вікні редактора висвічується блакитним рядок `getch()`;

25. Натискаємо клавішу <F8>. На передній план виступає консольне вікно з миготливим курсором. Натискаємо <Enter>. У вікні редактора висвічується блакитним рядок `}`.

26. Натискаємо клавішу <F8>. Консольне вікно закривається.

Покрокове виконання програми (Kultin162.cpp)

```
// Определитель матрицы второго порядка
#include <stdio.h>
#include <conio.h>
#include <windows.h>

char* Rus (const char*);

void main(){
float a[2][2], // матрица
      det;      // определитель (детерминант)
int i;
printf(Rus("\nВведите матрицу второго
порядка.\n"));
printf(Rus("После ввода элементов строки нажмиме
<Enter>\n"));

for (i = 0; i < 2; i++){
```

```

        printf("->");
        scanf("%f %f", &a[i][0], &a[i][1]);
    }

    det = a[0][0] * a[1][1] - a[0][1] * a[1][0];

    printf(Rus("Определитель матрицы\n"));

    for (i = 0; i < 2; i++)
        printf("%4.2f %4.2f\n", a[i][0], a[i][1]);

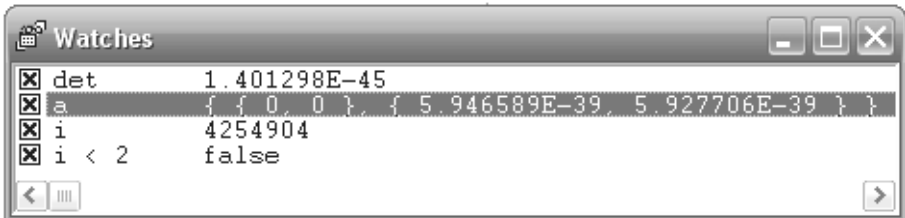
    printf(Rus("равен "));
    printf("%5.2f", det);

    printf(Rus("\nДля завершения нажмите <Enter>"));
    getch();
}

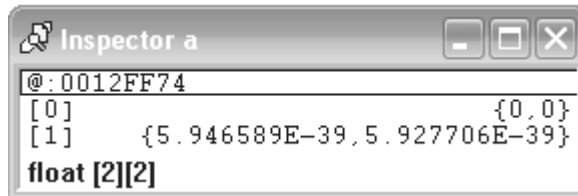
char* Rus (const char* text){
    char bufRus[256];
    CharToOem(text, bufRus);
    return bufRus;
}

```

1. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *void main ()*. Відкривається консольне вікно. За допомогою вікна спостереження переглядатимемо, як змінюються змінні *det*, *a*, *i* і вираз (*i*<2) при виконанні програми.



Змінні містять безглузді значення ("сміття"), оскільки вони не були ініціалізовані. Будемо досліджувати змінну *a* за допомогою інспектора.



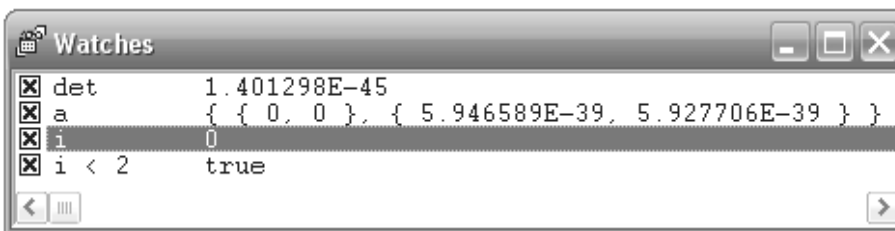
Оскільки змінна *a* є масивом, у вікні інспектора з'являються поточні значення усіх його елементів. Масив *a* заповнений "сміттям".

2. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок `printf(Rus("\nВведіть матрицю другого порядку.\n"));`

3. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *Введіть матрицю другого порядку*. У вікні редактора висвічується блакитним рядок `printf(Rus("Після введення елементів рядка натисніть <Enter>\n"));`.

4. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *Після введення елементів рядка натисніть <Enter>..* У вікні редактора висвічується блакитним рядок `for(i=0; i < 2; i++){`.

5. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна **Watches**:

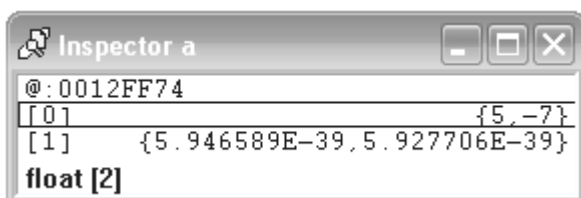
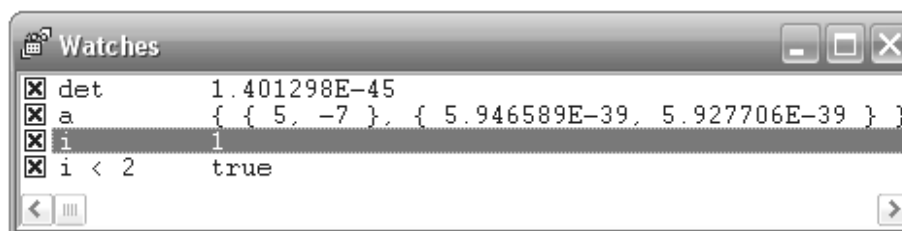


У вікні редактора висвічується блакитним рядок `printf("->");`.

6. Натискаємо клавішу <F8>. У консольне вікно виводиться *->.* У вікні редактора висвічується блакитним рядок `scanf("%f %f", &a[i][0], &a[i][1]);`.

7. Натискаємо клавішу <F8>. На передній план виступає консольне вікно з миготливим курсором. Вводимо 5, натискаємо

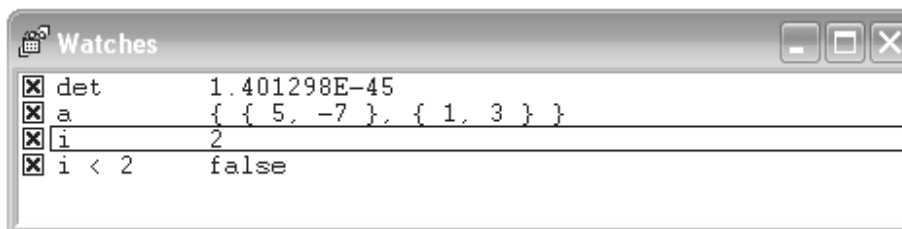
клавішу "Space", вводимо -7, натискаємо <Enter>. Подивимося, як змінився вміст вікон Watches і Inspector a:



У вікні редактора висвічується блакитним рядок `printf("->");`.

8. Натискаємо клавішу <F8>. У консольне вікно виводиться ->. У вікні редактора висвічується блакитним рядок `scanf("%f %f", &a[i][0], &a[i][1]);`.

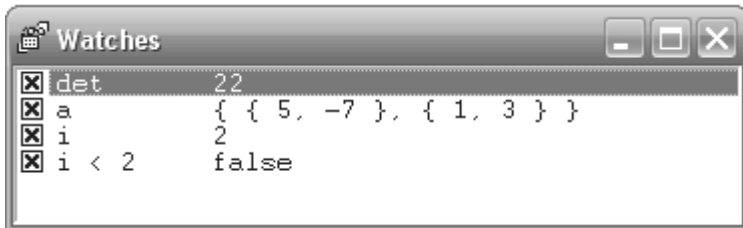
9. Натискаємо клавішу <F8>. На передній план виступає консольне вікно з миготливим курсором. Вводимо 1, натискаємо "Space", вводимо 3, натискаємо <Enter>. Подивимося, як змінився вміст вікон Watches і Inspector a:





У вікні редактора висвічується блакитним рядок $det = a[0][0] * a[1][1] - a[0][1] * a[1][0];$.

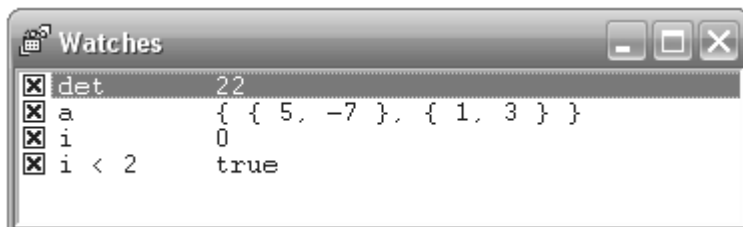
10. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:



У вікні редактора висвічується блакитним рядок `printf(Rus("Определитель матрицы\n"));`.

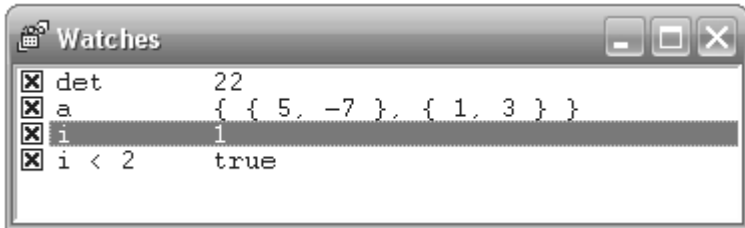
11. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *Определитель матрицы*. У вікні редактора висвічується блакитним рядок `for (i = 0; i < 2; i++)`.

12. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:



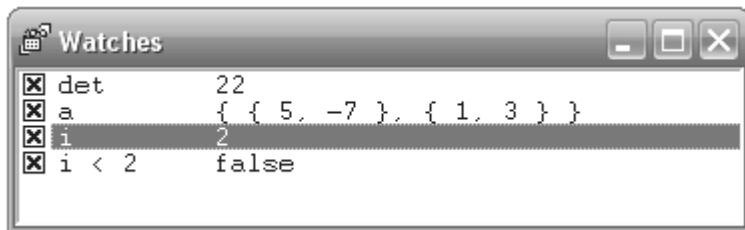
У вікні редактора висвічується блакитним рядок `printf("%4.2f\n", a[i][0], a[i][1]);`.

13. Натискаємо клавішу <F8>. У консольне вікно виводиться перший рядок матриці і курсор переводиться на новий рядок. Подивимося, як змінився вміст вікна Watches:



У вікні редактора висвічується блакитним рядок `printf("%.2f\n", a[i][0], a[i][1]);`.

14. Натискаємо клавішу <F8>. У консольне вікно виводиться другий рядок матриці і курсор переводиться на новий рядок. Подивимося, як змінився вміст вікна Watches:



У вікні редактора висвічується блакитним рядок `printf(Rus("павен "));`.

15. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *павен*. У вікні редактора висвічується блакитним рядок `printf("%.5.2f", det);`.

16. Натискаємо клавішу <F8>. У консольне вікно виводиться значення визначника матриці (22). У вікні редактора висвічується блакитним рядок `printf(Rus("\nДля завершення нажми <Enter>"));`.

17. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *Для завершення нажми <Enter>*. У вікні редактора висвічується блакитним рядок `getch();`.

18. Натискаємо клавішу <F8>. На передній план виступає консольне вікно з миготливим курсором. Натискаємо <Enter>. У вікні редактора висвічується блакитним рядок }.

19. Натискаємо клавішу <F8>. Консольне вікно закривається.

Покрокове виконання програми (Kultin163.cpp)

```
// Проверяет, является ли матрица "магическим"
квдратом
#include <stdio.h>
#include <conio.h>
#include <windows.h>

#define SZ 5          // максимальный размер матрицы

char* Rus (const char*);

void main(){
    int a[SZ][SZ], // матрица
        n,        // размер проверяемой матрицы
        ok,       // матрица - "магический" квадрат
        i, j,     // индексы матрицы
        sum,      // сумма эл-тов главной диагонали,
        temp;     // сумма эл-тов текущей строки,
        столбца или второй диагонали матрицы

    printf(Rus("*** МАГИЧЕСКИЙ КВАДРАТ ***\n"));
    printf(Rus("\nВведите размер матрицы (3..%i) -
>"), SZ);
    scanf("%i", &n);
    printf(Rus("Введите строки матрицы\n"));
    printf(Rus("После ввода строки, %i целых чисел,
"), n);
    printf(Rus("нажмите <Enter>\n"));

    for(i=0; i<n; i++){
```

```

printf("->");
for(j=0; j<n; j++)
    scanf("%i", &a[i][j]);
}

```

квадрат ok=1; // пусть матрица - "магический"

```

sum=0;
for(i=0; i<n; i++)
    sum+=a[i][i];

```

```

i=0;
do{
    temp=0;
    for (j=0; j<n; j++)
        temp+=a[i][j];
    if (temp!=sum)
        ok=0;
    i++;
} while (ok && i < n);

```

```

if (ok){
    j=0;
    do{
        temp=0;
        for(i=0; i<n; i++)
            temp+= a[i][j];
        if(temp!=sum) ok=0;
        j++;
    } while (ok && j < n);
}

```

```

if (ok){
    temp=0;
    i=n-1;
    for(j=0; j<n; j++)
        temp+=a[i--][j];
}

```

```

        if(temp!=sum) ok=0;
    }

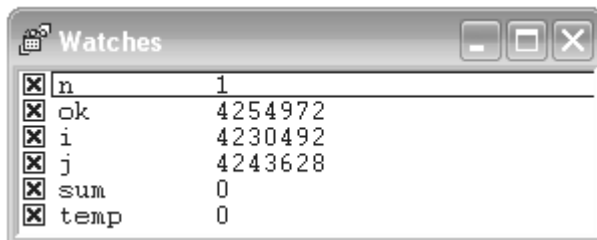
    printf(Rus("Введённая матрица "));
    if (!ok)
        printf(Rus("не "));
    printf(Rus("является \"магическим\" квадратом.
\n"));

    printf(Rus("\nДля завершения нажмите
<Enter>"));
    getch();
}

char* Rus (const char* text){
    char bufRus[256];
    CharToOem(text, bufRus);
    return bufRus;
}

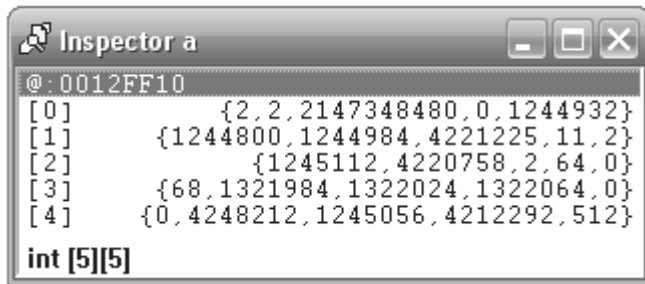
```

1. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *void main()*. Відкривається консольне вікно. За допомогою вікна спостереження переглядатимемо, як змінюються змінні *n*, *ok*, *i*, *j*, *sum*, *temp* при виконанні програми.



Змінні містять безглузді значення ("сміття"), оскільки вони не були ініціалізовані. Будемо досліджувати змінну *a* за допомогою інспектора.

Оскільки змінна *a* є масивом, у вікні інспектора з'являються поточні значення усіх його елементів. Масив *a* заповнений "сміттям".

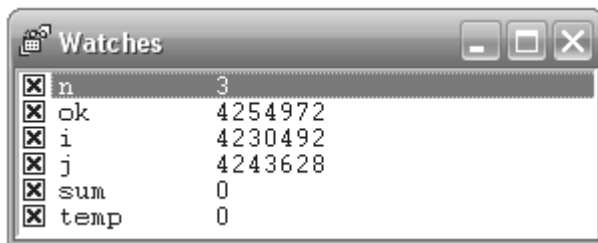


2. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок `printf(Rus("*** МАГИЧЕСКИЙ КВАДРАТ ***\n"));`.

3. Натискаємо клавішу <F8>. У консольне вікно виводиться текст `*** МАГИЧЕСКИЙ КВАДРАТ ***`. У вікні редактора висвічується блакитним рядок `printf(Rus("\nВведите размер матрицы (3..%i) ->", SZ));`.

4. Натискаємо клавішу <F8>. У консольне вікно виводиться текст `Введите размер матрицы (3..5) ->`. У вікні редактора висвічується блакитним рядок `scanf("%i", &n);`.

5. Натискаємо клавішу <F8>. На передній план виступає консольне вікно з миготливим курсором. Вводимо 3 і натискаємо <Enter>. Подивимося, як змінився вміст вікна Watches:



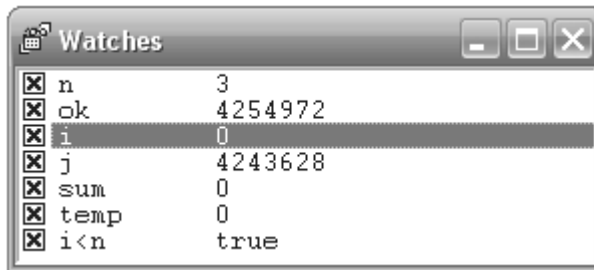
У вікні редактора висвічується блакитним рядок `printf(Rus("Введите строки матрицы\n"));`.

6. Натискаємо клавішу <F8>. У консольне вікно виводиться текст `Введите строки матрицы`. У вікні редактора висвічується блакитним рядок `printf(Rus("После ввода строки, %i целых чисел, "), n);`.

7. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *После ввода строки, 3 целых чисел.* У вікні редактора висвічується блакитним рядок `printf(Rus("нажмиме <Enter>\n"));`.

8. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *нажмиме <Enter>*. У вікні редактора висвічується блакитним рядок `for(i=0; i<n; i++){`.

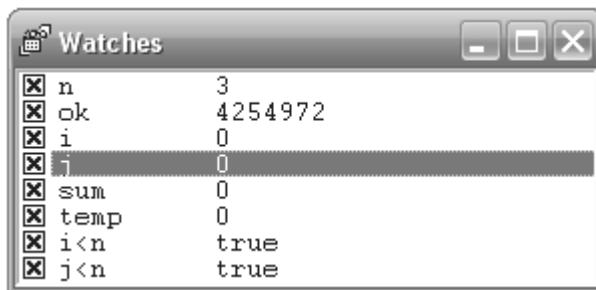
9. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches і значення вираження ($i < n$):



У вікні редактора висвічується блакитним рядок `printf("->");`.

10. Натискаємо клавішу <F8>. У консольне вікно виводиться *->*. У вікні редактора висвічується блакитним рядок `for(j=0; j<n; j++){`.

11. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches і значення вираження ($j < n$):



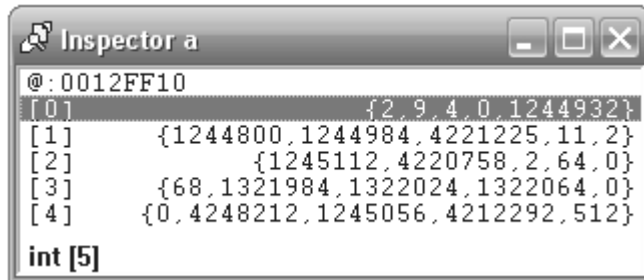
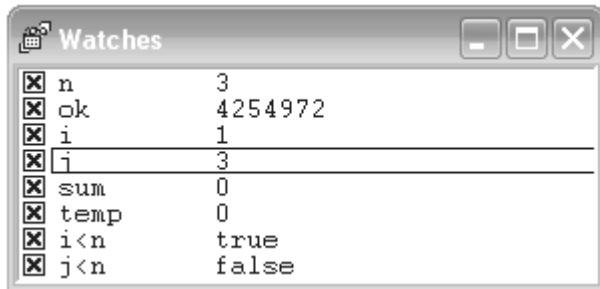
У вікні редактора висвічується блакитним рядок `scanf("%i", &a[i][j]);`.

12. Натискаємо клавішу <F8>. На передній план виступає консольне вікно з миготливим курсором. Вводимо через пробіл

елементи першого рядка матриці 2, 9, 4 і натискаємо <Enter>. У вікні редактора висвічується блакитним рядок `scanf("%i", &a[i][j]);`.

13. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок `scanf("%i", &a[i][j]);`.

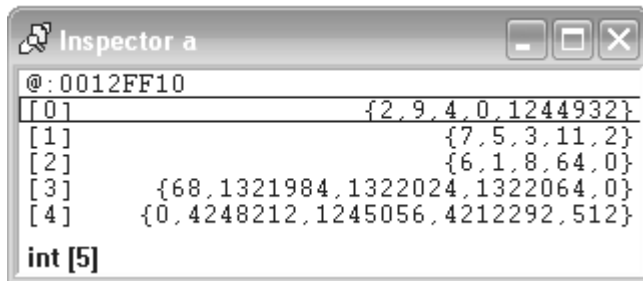
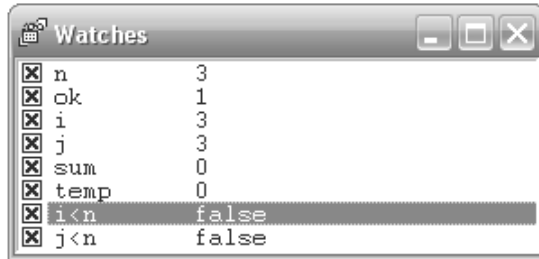
14. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікон Watches і Inspector a:



У вікні редактора висвічується блакитним рядок `printf("->");`.

15-23. Аналогічно пунктам 10-14 вводиться другий і третій рядок матриці 7, 5, 3 і 6, 1, 8 відповідно.

24. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікон Watches і Inspector a:

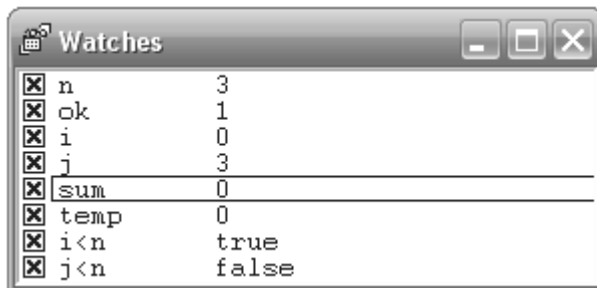


У вікні редактора висвічується блакитним рядок $ok=1$;

25. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок $sum=0$;

26. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок $for(i=0; i<n; i++)$.

27. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:

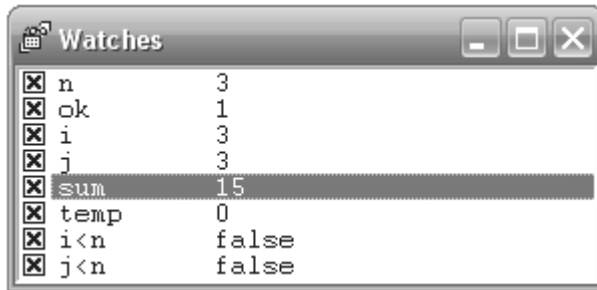


У вікні редактора висвічується блакитним рядок $sum+=a[i][i]$;

28. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок $sum += a[i][i];$.

29. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок $sum += a[i][i];$.

30. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:



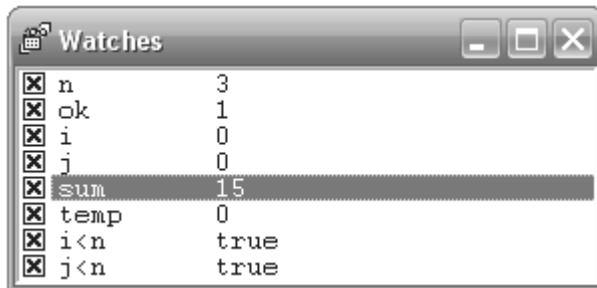
У змінній sum – сума елементів головної діагоналі матриці.

У вікні редактора висвічується блакитним рядок $i=0;$.

31. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок $temp=0;$.

32. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок $for(j=0; j<n; j++)$.

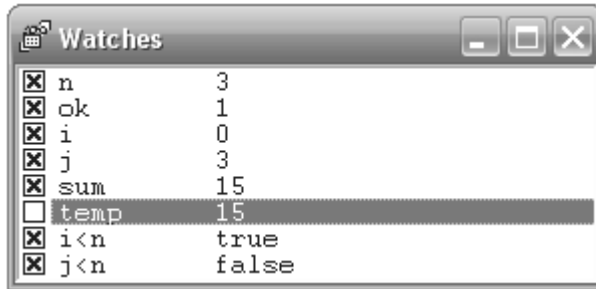
33. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:



У вікні редактора висвічується блакитним рядок $temp += a[i][j];$.

34-35. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок `temp+=a[i][j];`.

36. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:



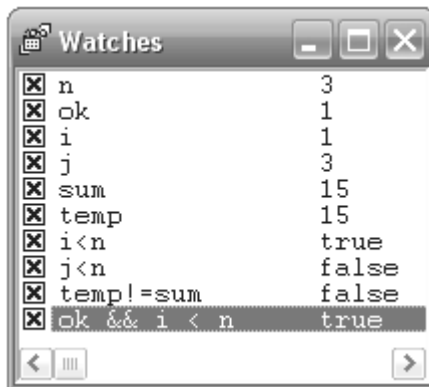
У змінній `temp` – сума елементів першого рядка матриці.

У вікні редактора висвічується блакитним рядок `if (temp!=sum).`

37. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок `i++;`.

38. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок `while(ok && i < n);`.

39. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:



У вікні редактора висвічується блакитним рядок `temp=0;`.

40-47. Аналогічно 32-39 перевіряється рівність суми елементів другого рядка матриці сумі елементів головної діагоналі.

48-54. Аналогічно 32-39 перевіряється рівність суми елементів третього рядка матриці сумі елементів головної діагоналі.

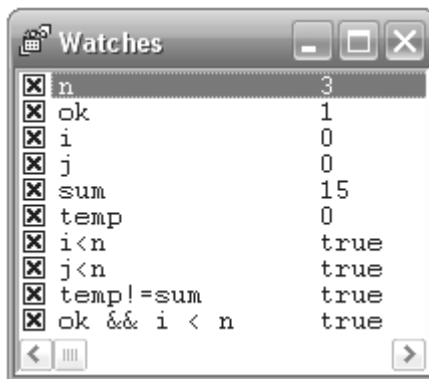
55. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *if(ok){*.

56. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *j=0;*.

57. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *temp=0;*.

58. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *for(i=0; i<n; i++)*.

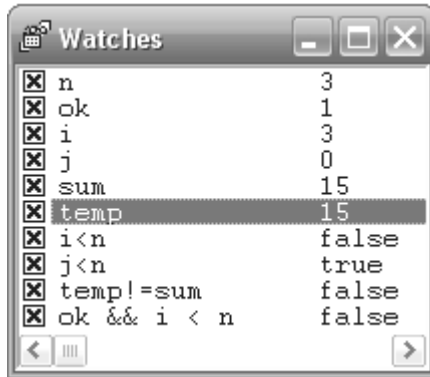
59. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:



У вікні редактора висвічується блакитним рядок *temp+=a[i][j];*.

60-61. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *temp+=a[i][j];*.

62. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:



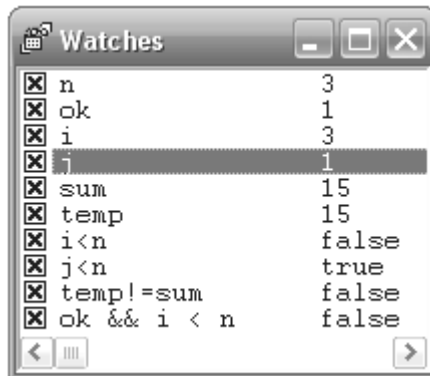
У змінній *temp* – сума елементів першого стовпця матриці.

У вікні редактора висвічується блакитним рядок *if (temp!=sum)*.

63. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *j++;*.

64. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *while(ok && j < n);*.

65. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:



У вікні редактора висвічується блакитним рядок *temp=0;*.

66-73. Аналогічно 58-65 перевіряється рівність суми елементів другого стовпця матриці сумі елементів головної діагоналі.

74-80. Аналогічно 58-65 перевіряється рівність суми елементів третього стовпця матриці сумі елементів головної діагоналі.

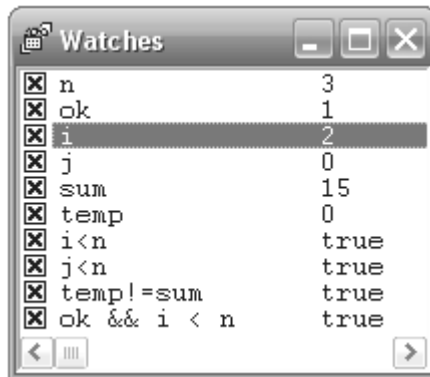
81. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *if(ok){*.

82. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *temp=0;*.

83. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *i=n - 1;*.

84. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *for(j=0; j<n; j++)*.

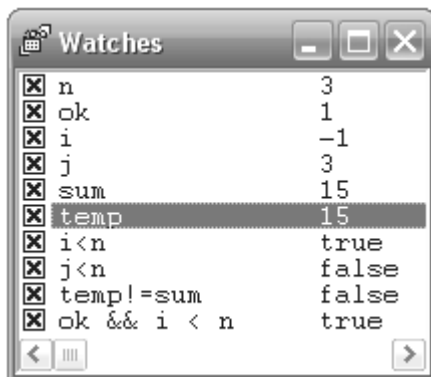
85. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:



У вікні редактора висвічується блакитним рядок *temp+=a[i--][j];*.

86-87. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *temp+=a[i--][j];*.

88. Натискаємо клавішу <F8>. Подивимося, як змінився вміст вікна Watches:



У змінній *temp* – сума елементів побічної діагоналі матриці.

У вікні редактора висвічується блакитним рядок *if (temp!=sum)*.

89. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *printf(Rus("Введённая матрица "));*.

90. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *Введённая матрица*. У вікні редактора висвічується блакитним рядок *if (!ok)*.

91. Натискаємо клавішу <F8>. У вікні редактора висвічується блакитним рядок *printf(Rus("является \"магическим\" квадратом. \n"));*.

92. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *является "магическим" квадратом*. У вікні редактора висвічується блакитним рядок *printf(Rus("\nДля завершения нажмите <Enter>"));*.

93. Натискаємо клавішу <F8>. У консольне вікно виводиться текст *Для завершения нажмите <Enter>*. У вікні редактора висвічується блакитним рядок *getch();*.

94. Натискаємо клавішу <F8>. На передній план виступає консольне вікно з миготливим курсором. Натискаємо <Enter>. У вікні редактора висвічується блакитним рядок *}*.

95. Натискаємо клавішу <F8>. Консольне вікно закривається.

ЛІТЕРАТУРА

1. Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Селюн М.И. Задачи по программированию. – М.:Наука, 1988. – 224с.
2. Павловская Т.А. С/С++. Программирование на языке высокого уровня. – СПб.: Питер, 2003. – 461с.
3. Павловская Т.А. Структурное программирование: Учеб. пособие/ Т.А. Павловская, Ю.А. Щупак. – СПб.: Питер, 2002. – 240с.
4. Керниган Б. Язык программирования С: Пер. с англ./Б. Керниган, Д. Ритчи. – 2-е изд., перераб. И доп. – М.: Вильямс, 2006. – 304с.
5. Крячков А.В., Сухинина И.В., Томшин В.К. Программирование на С и С++: Учеб. Пособие для вузов/ Пер. ред. В.К. Томшина. – 2-е изд., испр. – М.: Горячая линия – Телеком, 2006. – 344с.
6. Демидович Е.М. Основы алгоритмизации и программирования. Язык С: Учеб. пособие, - СПб.: БХВ – Петербург, 2006. – 440с.
7. Шилдт Г. Искусство программирования на С++. СПб.: БХВ – Петербург, 2006. – 496с.
8. Шилдт Г. Справочник программиста по С/С++: Пер с англ. – 2-е изд. – М.: ВШ, 2000. – 447с.
9. Культин М.Б. С/С++ в задачах и примерах. – СПб.: БХВ – Петербург, 2005. – 288с.
10. Дейтел Х.М. Дейтел П.Дж. Как программировать на С++: Пер с англ. – М.: Бином, 1999. – 1022с.