Debugowanie nie jest opcją domyślną projektów DevCpp, dlatego należy włączyć ją w opcjach projektu:

C compiler C++ compiler Code Generation Code profiling Linker Do not use standard system startup files or libraries Do not create a console window Strip executable No	General   Files   Compiler NOTE: These settings will Compiler:   Default com	Parameters   Directories   Build Opti   Makefil override the global Compiler Options affecting this p apiler	e   Version info roject only.
	C compiler C++ compiler Code Generation Code profiling Linker Optimization	Link an Objective C program Generate debugging information Do not use standard system startup files or libraries Do not create a console window Strip executable	No Yesl ▼ No No

Należy też sprawdzić, czy przypadkiem nie jest włączona optymalizacja (Optimization). Uruchomienie programu:

🚟 Dev-C++ 4.9.9.2 - [ Odplusk ] - Odplusk.dev	
<u>File Edit Search View Project Execute Debug Tools CVS Window H</u> elp	
🕲 🕲   🗋 🖳 🍇   🖴 🛹   💁 🖾   🖶   餐 🕾   🧧	
📙 🗄 📰 🔠 🧭 📔 🦿 😨 📔 New 🗐 Insert 🚱 Toggle 🔲 Goto	
Project Classes Debug [*] maindeb.c	
#include <stdio.h></stdio.h>	*
#include <stdlib.h></stdlib.h>	
int i, a;	
int main(int argc, char *argv[])	
	_
a=0;	
<b>for</b> (i=0.i<10.i++)	
rintf("%d %d)n", i, a);	
a++;	
}	
system("PAUSE");	
return 0;	
}	
🔡 Compiler 🖷 Resources 🚮 Compile Log 🗹 Debug 🖾 Find Results 🏶 Close	
Debug Backtrace Output	
[ <u>N</u> ext Step 《 Continue 《 Debug	<b>S</b> . 1
Step Into	<b>⊕</b> <u>B</u> €
13: 1 Modified Insert 19 Lines in file	
	///

odbywa się opcją 'Debug' lub 'Run to Cursor'. Miejsca zatrzymania programu określa się znacznikiem na czarnym pasku. W trakcie debagowania można podglądać wartości zmiennych. Dodanie zmiennych przedstawione jest poniżej:



New Variable Watch	×
Enter Variable Name:	
a	
OK Cancel	

Proces debagowania pozwala na śledzenie kolejnych linii wykonywanego kodu, wchodzenie w głąb lub niezagłębianie się w kolejne wywołania, czy też dalsze wykonywanie do następnego znacznika.



🔍 D:\Odp	lusk.exe	<u> </u>
0 0 1 1 2 2	Project       Classes       Debug       Maindeb.c	
	<pre>a=2 #include <stdio.h> #include <stdlib.h> int i, a; int main(int argc, char *argv[]) {     a=0;     for(i=0;i&lt;10;i++)     {         printf("%d %d\n", i, a);         a++;         }         system("PAUSE");         return 0;     } </stdlib.h></stdio.h></pre>	
	Compiler       Image: Compile Log       Image: Compile Log<	
	Image: Next Step       Image: Continue       Image: Debug         Image: Next Step Into       Image: Plant to Cursor       Image: Stop Execution	🖾 : 🖨 Br
	13: 2 Insert 19 Lines in file	li.

Przykładowa pętla była wykonywana krok po kroku – niebieska strzałka oznacza zatrzymanie się przed wykonaniem instrukcji w linii. Na bieżąco wyświetlane wartości zgadzają się z wynikami prezentowanymi przez program.