

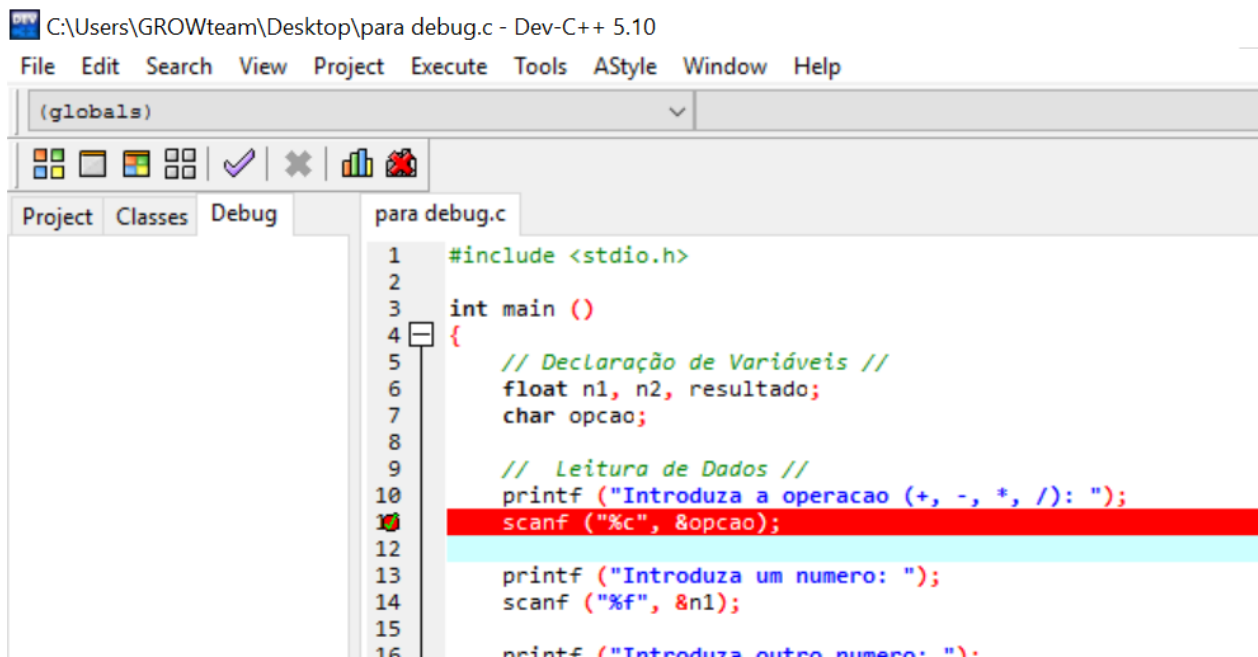
Tutorial do Depurador de Código (debugger)


O debugger (depurador) é um programa usado para testar um programa em C e fazer sua depuração, que consiste em executar instrução a instrução um programa. Ajuda a encontrar problemas do programa. Como utilizar:

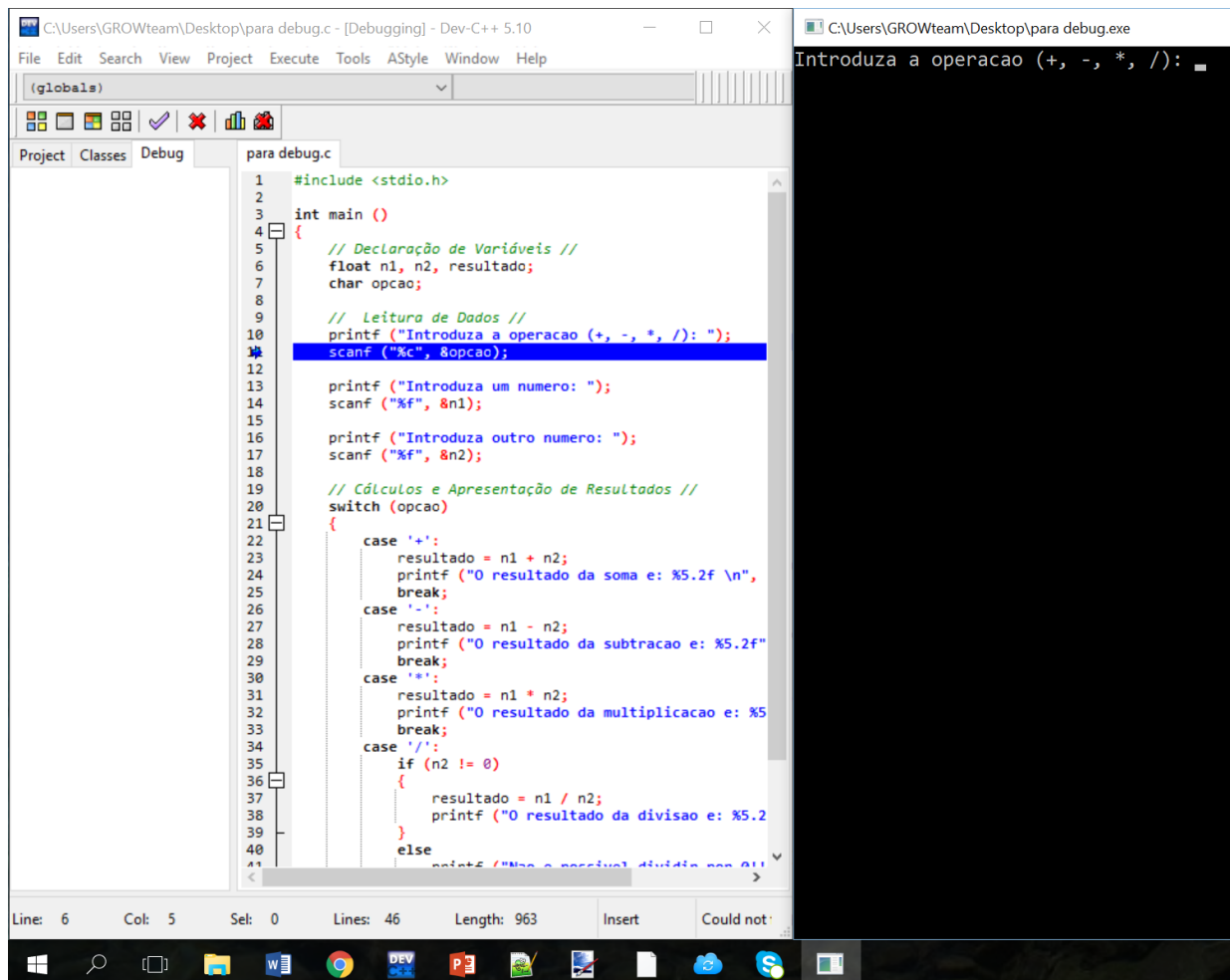
1. Ativar o compilador debug



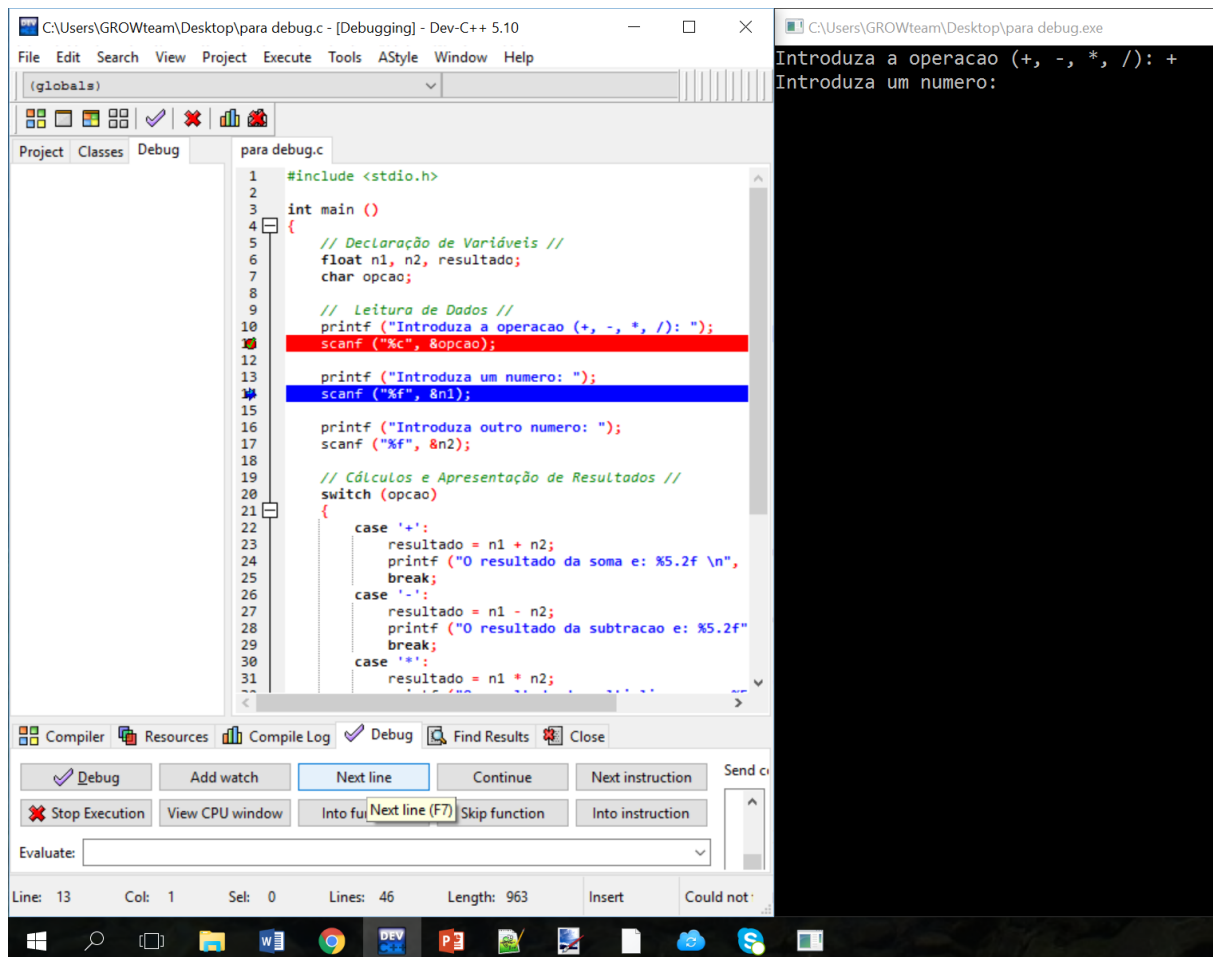
2. **Compilar** o código escrito (tecla F9). Este não deverá apresentar erros.
3. Colocar um **breakpoint** (ponto de paragem de execução) numa determinada instrução do código, clicando no numero de uma linha do código. Aparecerá uma bola vermelha sobre o número da linha e esta ficará vermelha, tal como na figura em baixo.



4. Correr o **Debug** (tecla F5 ou botão ). O programa será executado normalmente, de forma sequencial, parando no breakpoint. Uma linha a azul indica a próxima instrução a ser executada (neste caso, a linha onde foi colocado o breakpoint). Aparecerá uma janela de comando preta que se deverá colocar num dos lados do ecrã, tal como representado na figura em baixo, para permitir a correto acompanhamento do programa nas entradas de dados que requiere e nas saídas de dados que apresenta.

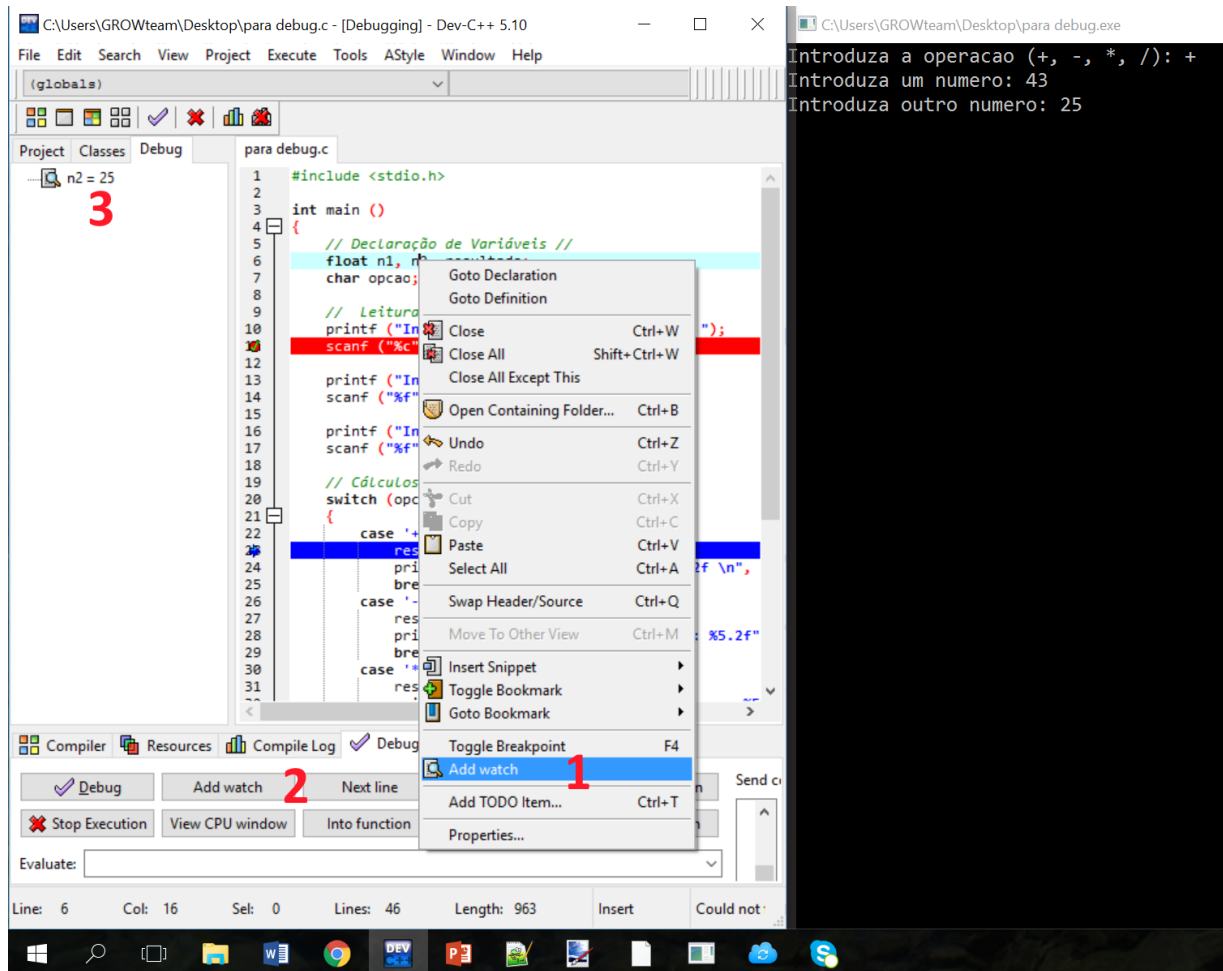


5. É possível executar passo a passo cada instrução do programa selecionando a tecla **F7** ou clicando no botão “**Next Line**” na janela Report Window que aparece em baixo (se não aparecer, ativar a sua visualização em “View\Floating Report Window”. Clicando uma segunda vez, em vez de flutuante ficará fixa, como em baixo na imagem).



6. Alternativamente, é possível introduzir outros **breakpoints**, deixando o programa executar até ao ponto de paragem seguinte.

7. É possível fazer a visualização do valor de variáveis (**Add watch**). Basta colocar o cursor numa variável, clicar no botão direito do rato e seleccionar Add Watch (como ilustrado na figura, para a variável `n2`, 1). Alternativamente, pode-se clicar diretamente no botão “Add watch” da Report Window (2). Aparecerá na janela esquerda de Debug o nome da variável juntamente com o seu valor (em 3, `n2=25`). Esta ferramenta é muito útil para acompanhar o desenrolar do programa e detectar possíveis erros.



8. É essencial saber utilizar o Debug. A sua utilização será também avaliada nos testes e exame.

Exercícios

2. Uso do debugger


Considere o seguinte programa. Copie-o para o Dev-C++, gravando-o com o nome **Lab4_ex1_NomeApelido.c** (faça semelhante com os restantes exercícios).

```
#include <stdio.h>
int main ()
{
    // Declaração de Variáveis //
    int n1, n2;
    float resultado;
    char opcao;

    // Leitura de Dados //
    printf ("Introduza um numero: ");
    scanf ("%d", &n1);
    printf ("Introduza outro numero: ");
    scanf ("%f", &n2);
    printf ("Introduza a operacao (carater +, -, *, /): ");
    scanf ("%c", &opcao);

    // Cálculos e Apresentação de Resultados //
    switch (opcao)
    {
        case '+':
            resultado = n1 - n2;
            printf ("Resultado da soma: %5.2d \n", resultado);
            break;
        case '-':
            resultado = n1 + n2;
            printf ("Resultado da subtracao: %5.2f", &resultado);
            break;
        case '*':
            resultado = n1 * n1;
            printf ("Resultado da multiplicacao: %5.2f", resultado);
            break;
        case '/':
            if (n2 != 0)
                printf ("Nao e possivel dividir por 0!!\n");
            else
                resultado = n2 / n1;
                printf ("Resultado da divisao: %5.2f", resultado);
                break;
        default:
            printf("Selecionou uma operação nao existente.");
    }
}
```

O programa compila, mas existe uma serie de erros que só se evidenciam quando o executamos. Faça os seguintes passos:

- a) Compile e corra o programa e veja se funciona como expectável.
- b) Coloque um breakpoint na instrução `printf ("Introduza um numero: ");`
- c) Corra o modo debug (tecla F5 ou botão  Debug).
- d) Adicione a visualização de todas as variáveis, fazendo “add watch” para cada uma delas.
- e) Execute o programa linha a linha com o comando “next line” (F7), conforme indicado no tutorial, tendo as janelas do Dev-C++ e a janela de comando (janela preta) lado a lado.
- f) Corra o programa várias vezes, experimente todas as operações disponíveis (+,-,*,/), conferindo os resultados.
- g) Se o programa deixa de funcionar depois de um determinado passo, é porque existe um problema numa instrução, que deverá corrigir. Introduza um comentário nessa linha indicando qual o erro detectado.
- h) Faça as correções necessárias para que o programa funcione corretamente na sua totalidade. Para tal, deverá testar todas as operações, correndo várias vezes o modo debug. Teste em particular a divisão por 0.
- i) Submeta a versão corrigida, com os comentários em cada linha que alterou.

1. Características de um número

Escreva um programa que solicite um número inteiro ao utilizador. Em relação ao número inserido, deve avaliar:

- se é par.
- se está entre 0 e 20 (inclusivé).
- se é múltiplo de 5 mas não é múltiplo de 2.
- Se está entre -20 e -10 (inlcusivé) ou entre 10 e 20 (inlcusivé) e é um número par
- Se é um múltiplo de 7, não é negativo e tem 3 dígitos (uma forma de ver se tem 3 dígitos é se a divisão por 100 dá entre 1 e 9).

Para cada um dos requisitos, deve construir duas condições diferentes mas equivalentes (recorrendo entre outros às regras de Morgan), e imprimir sempre o resultado das duas. Por exemplo, se se perguntar se o número é positivo, pode avaliar das seguintes formas, imprimindo sempre duas vezes se for correto:

```
if (n > 0)
    printf("%d é positivo\n", n);
if (!(n <= 0))
    printf("%d é positivo\n", n);
```

2. Nome do mês

Crie um programa que, utilizando um switch:

- solicita a introdução do número de um mês.
- apresenta o nome desse mês por extenso.
- Caso o número não corresponda a um mês válido, deverá mostrar a mensagem “Mês Inválido”.

Exemplo de interação com o utilizador:

```
Insira um numero do mes: 2
O mes 2 corresponde a fevereiro
```

3. Identificador de carater

Escreva um programa que solicite ao utilizador para inserir um carater. Indique o código ASCII do carater inserido. Indique também se se trata de uma letra ou um número, e se a letra é maiúscula ou minúscula.

4. Pedido insistente

Escreva um programa que solicite ao utilizador para inserir dois numeros, o primeiro maior que o segundo. Enquanto esta condição não se verificar, imprima uma mensagem a indicar que não respeitou o requisito e volte a pedir.