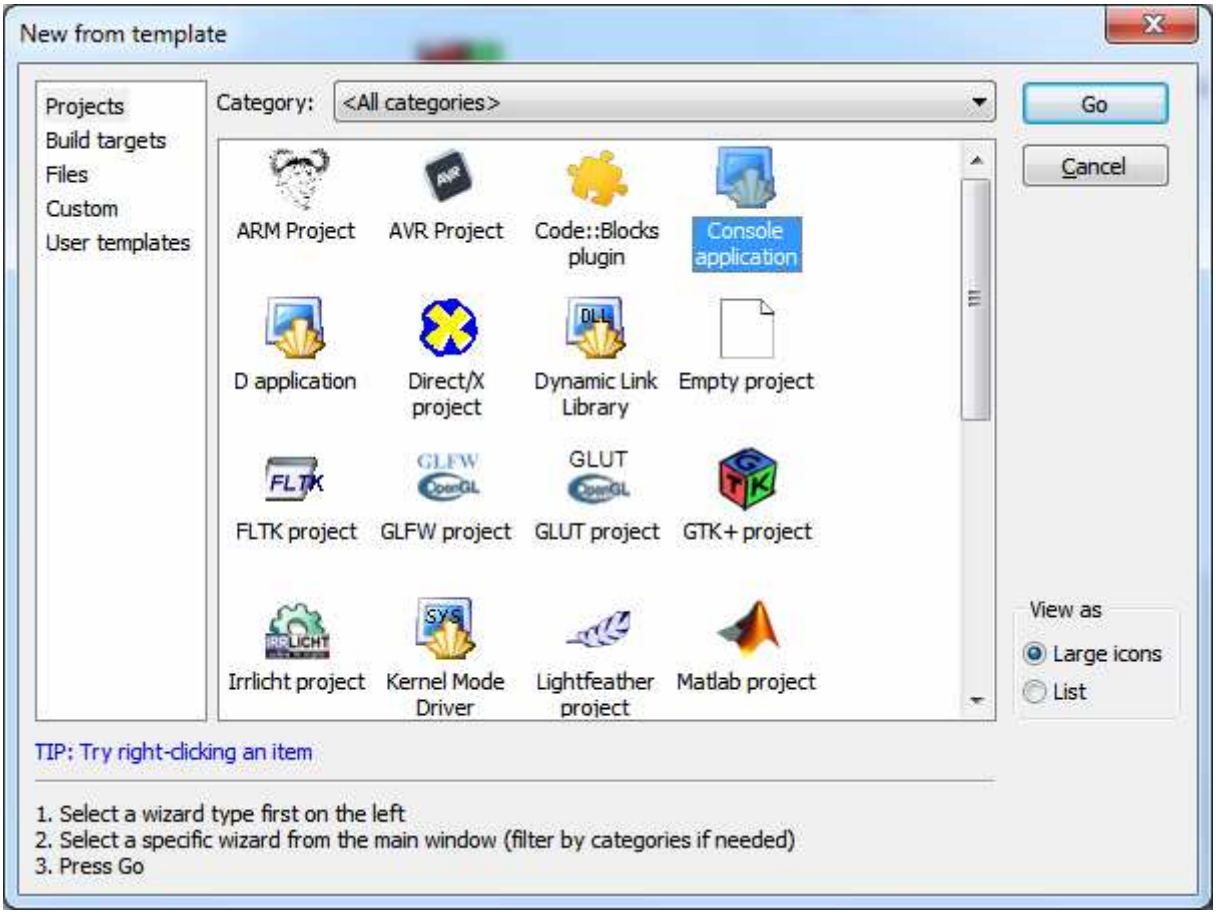
Język C i C++. Podstawy

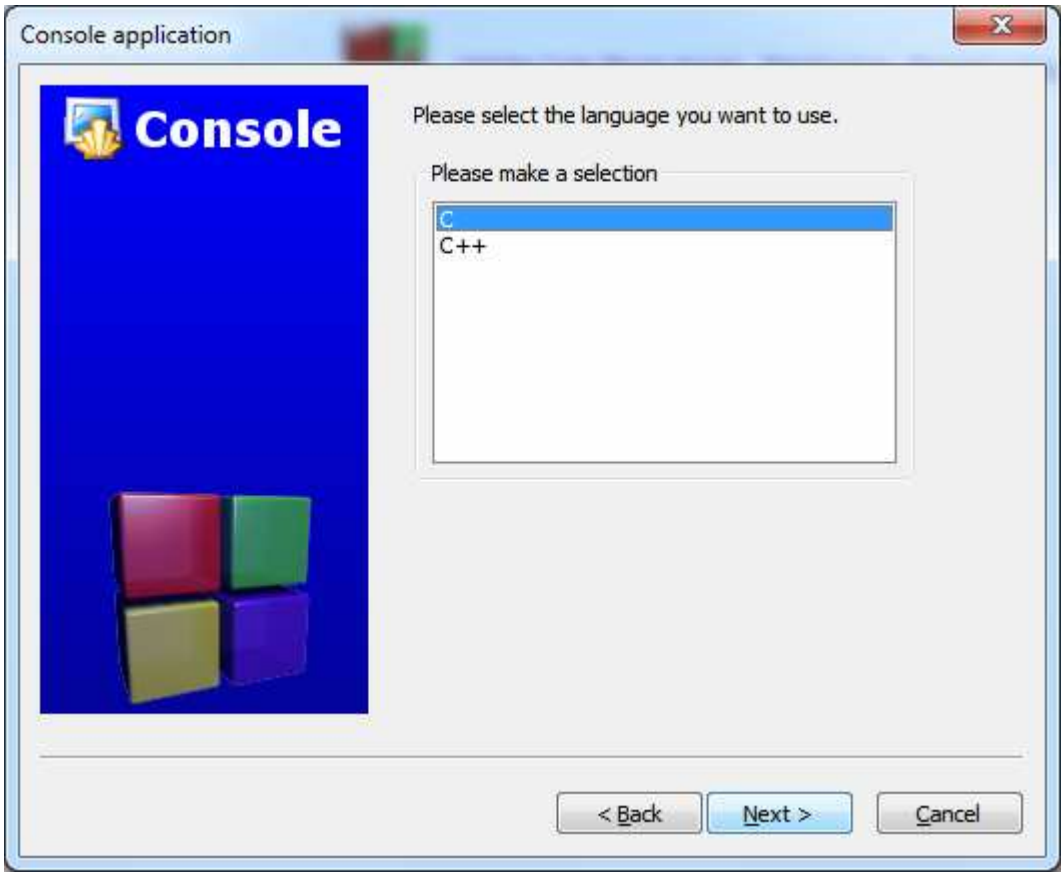
Zagadnienia do opanowania

1) Czym jest projekt C/C++, jakie pliki wchodzą w jego skład?
2) W jaki sposób z pliku źródłowego (C, C++) powstaje kod maszynowy (plik wykonywalny)?
3) Jaką rolę pełni Preprocesor, Kompilator, Linker? Za co odpowiada? W którym momencie jest używany?
4) Czym się różnią polecenia Compile, Build, Run?
5) Do czego służy Clean?
6) Kiedy warto używać dystrybucji Debug, a kiedy Release?

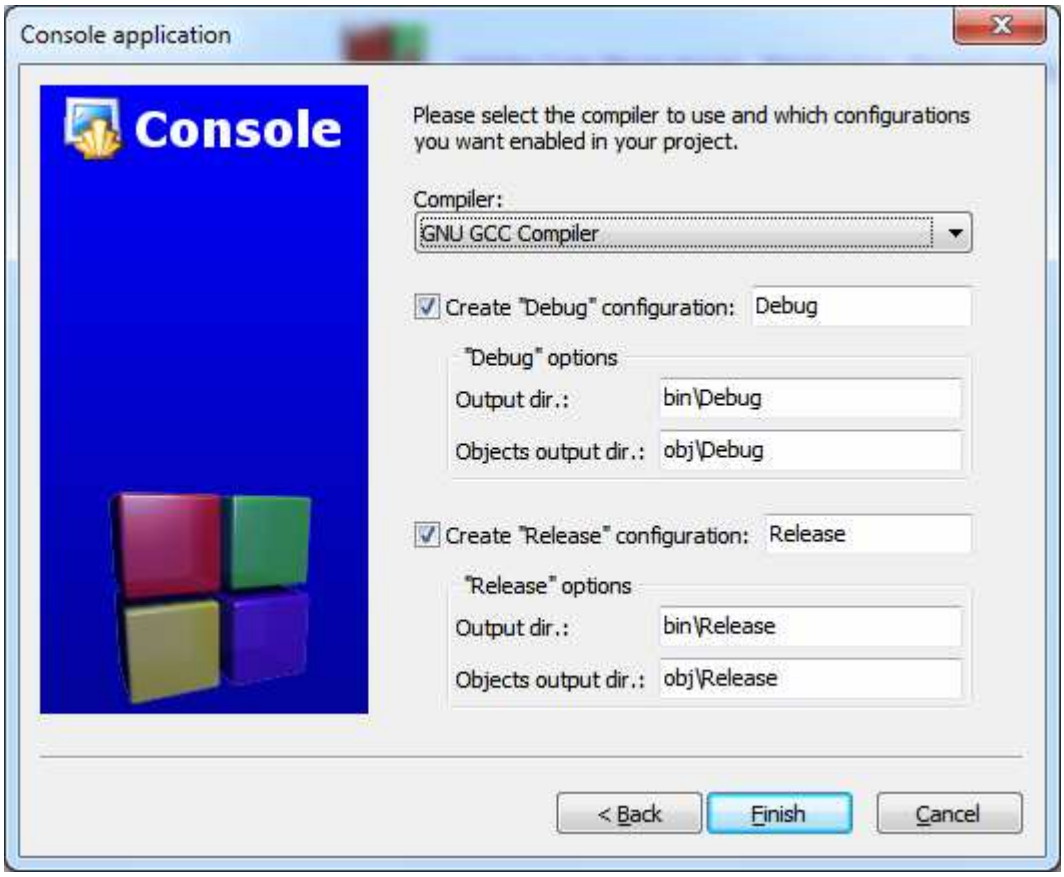
Zakładamy projekt w Code::Blocks

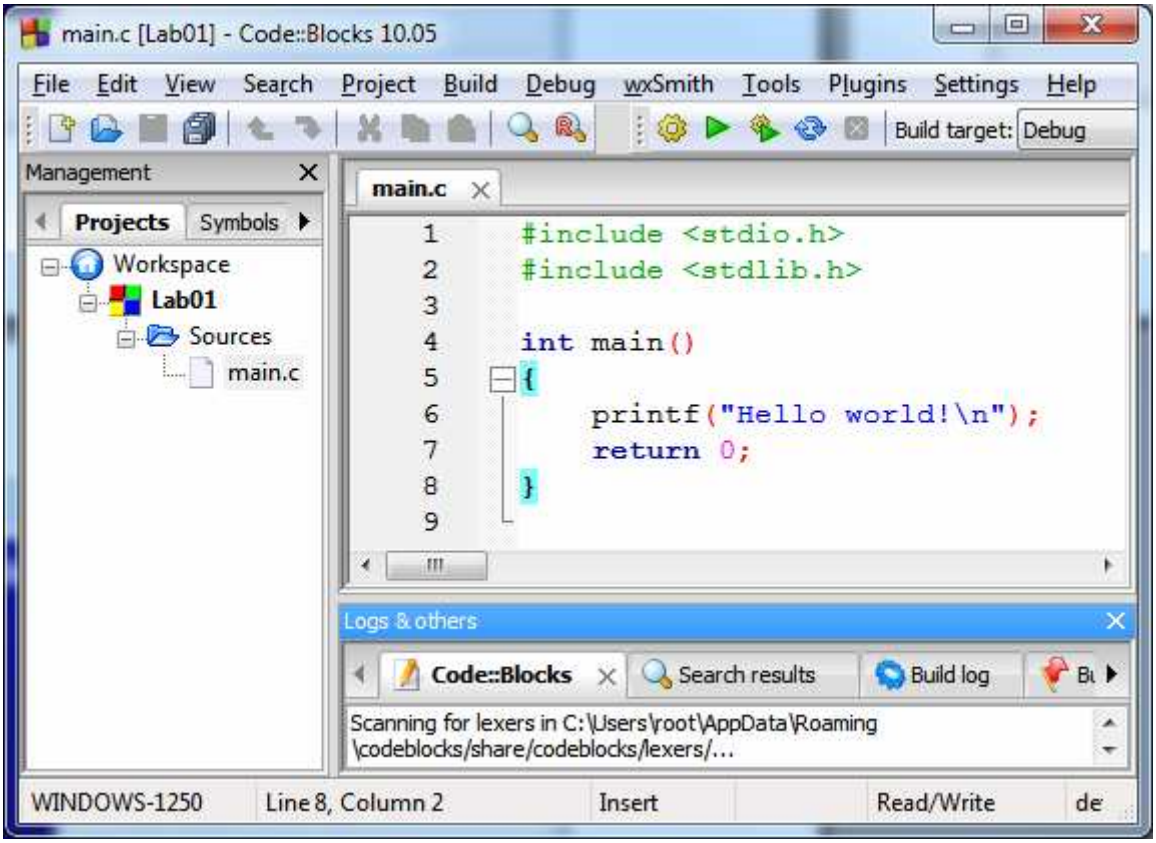
Przygotuj podstawowy projekt konsolowy „Hello, world!" w środowisku Code::Blocks.



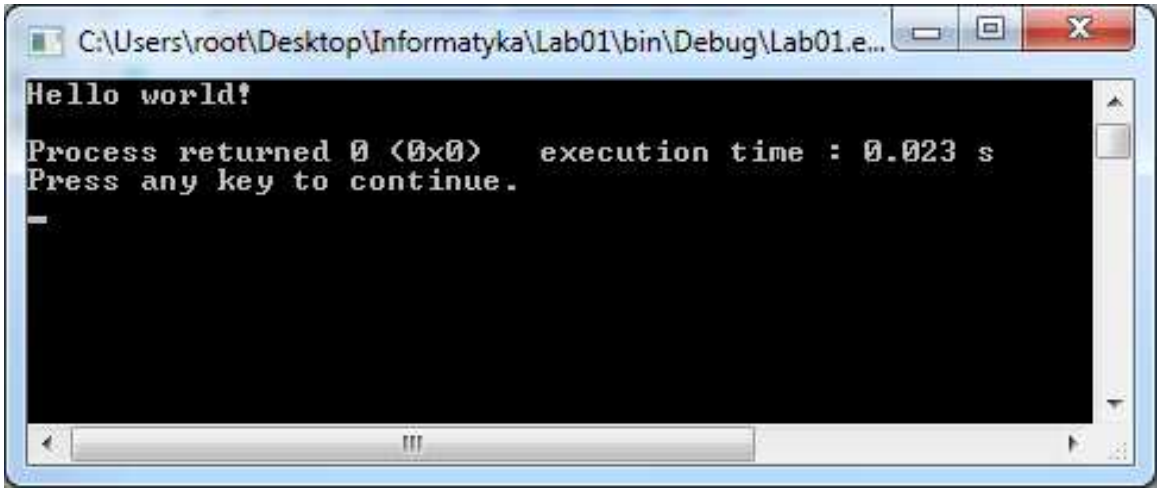



Console application
Console
Please select the folder where you want the new project to be created as well as its title.
Project title:
Lab01
Folder to create project in:
C:\Users\root\Desktop\Informatyka
Project filename:
Lab01.cbp
Resulting filename:
C:\Users\root\Desktop\Informatyka\Lab01\Lab01.cbp
< Back
Next >
Cancel

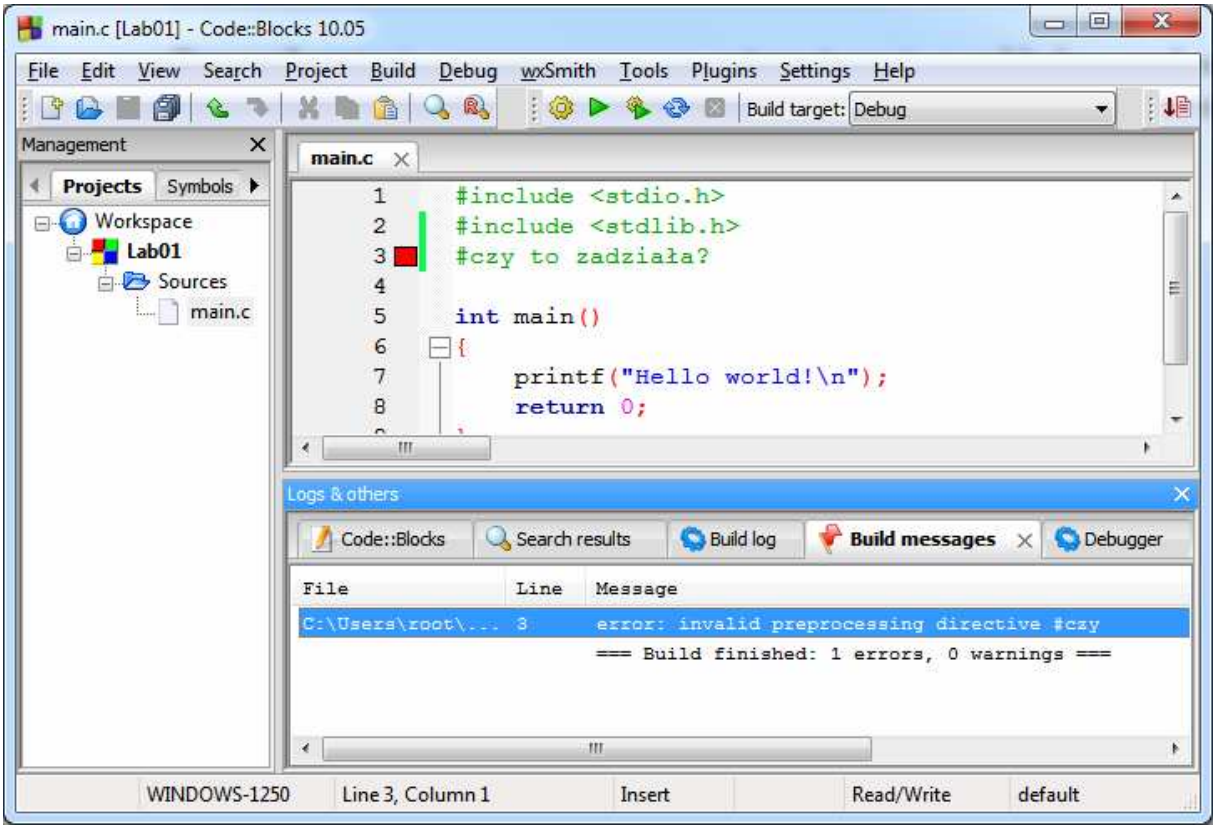

Console application
Console
Please select the compiler to use and which configurations you want enabled in your project.
Compiler:
GNU GCC Compiler
Create "Debug" configuration:
Debug
"Debug" options
Output dir.:
bin\Debug
Objects output dir.:
obj\Debug
Create "Release" configuration:
Release
"Release" options
Output dir.:
bin\Release
Objects output dir.:
obj\Release
< Back
Finish
Cancel



Po uruchomieniu wygenerowanego automatycznie przykładowego kodu (zielony trójkąt w pasku narzędziowym lub Build→Run lub F9) powinno się pojawić okienko konsoli z wynikami:



Błędy preprocesora



Celowo w linii 3 dopisany został następujący tekst:

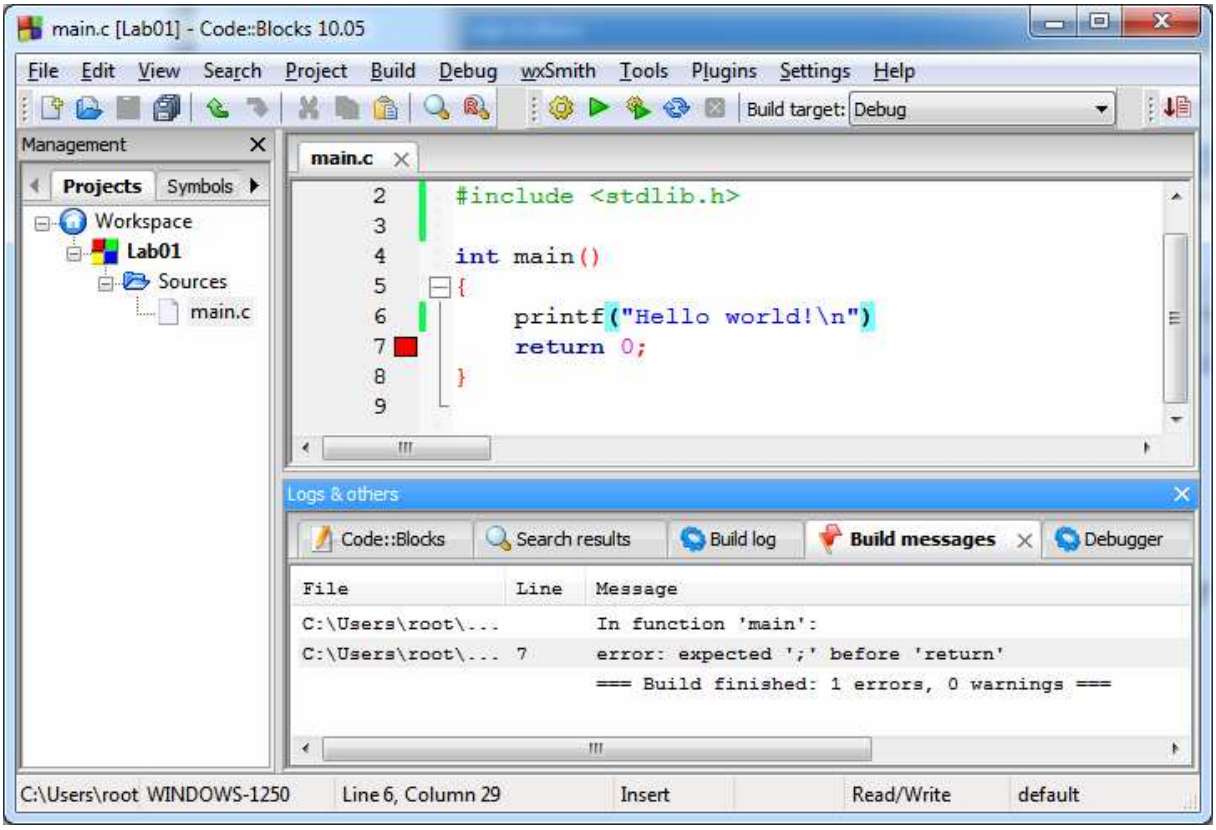
```
#czy to zadziała?
```

Komunikat błędu w tym przypadku będzie miał postać:

```
main.c|3|error: invalid preprocessing directive #czy
```

Wiersze rozpoczynające się od znaku # to dyrektywy preprocesora.

Błędy kompilatora

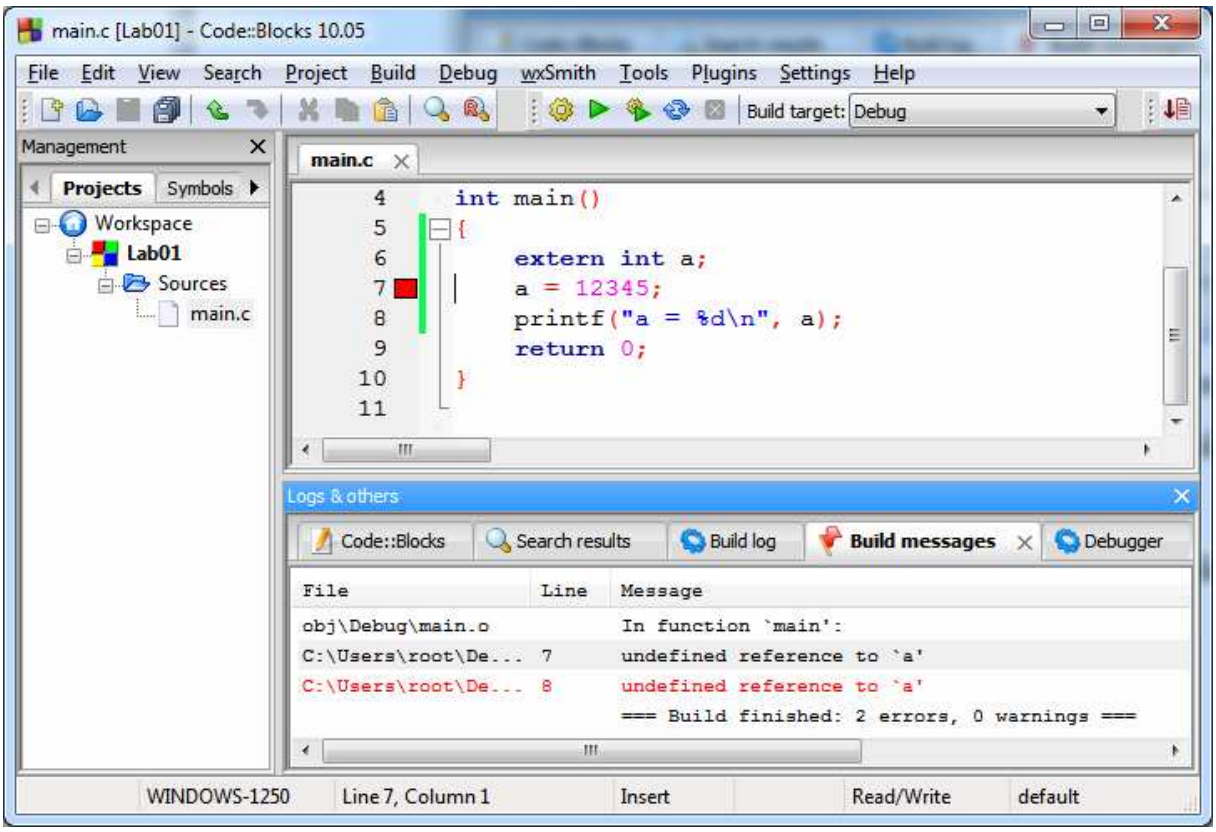


W wierszu 6, za printf(…), usunięty został średnik. Efektem jest kod źródłowy niezgodny z regułami języka C, co spowoduje zgłoszenie błędu kompilatora:

```
main.c|7|error: expected ';' before 'return'|
```

Pomiędzy instrukcjami ‘printf’ (wywołaniem funkcji) a ‘return’ kompilator spodziewa się średnika.

Błędy linkera

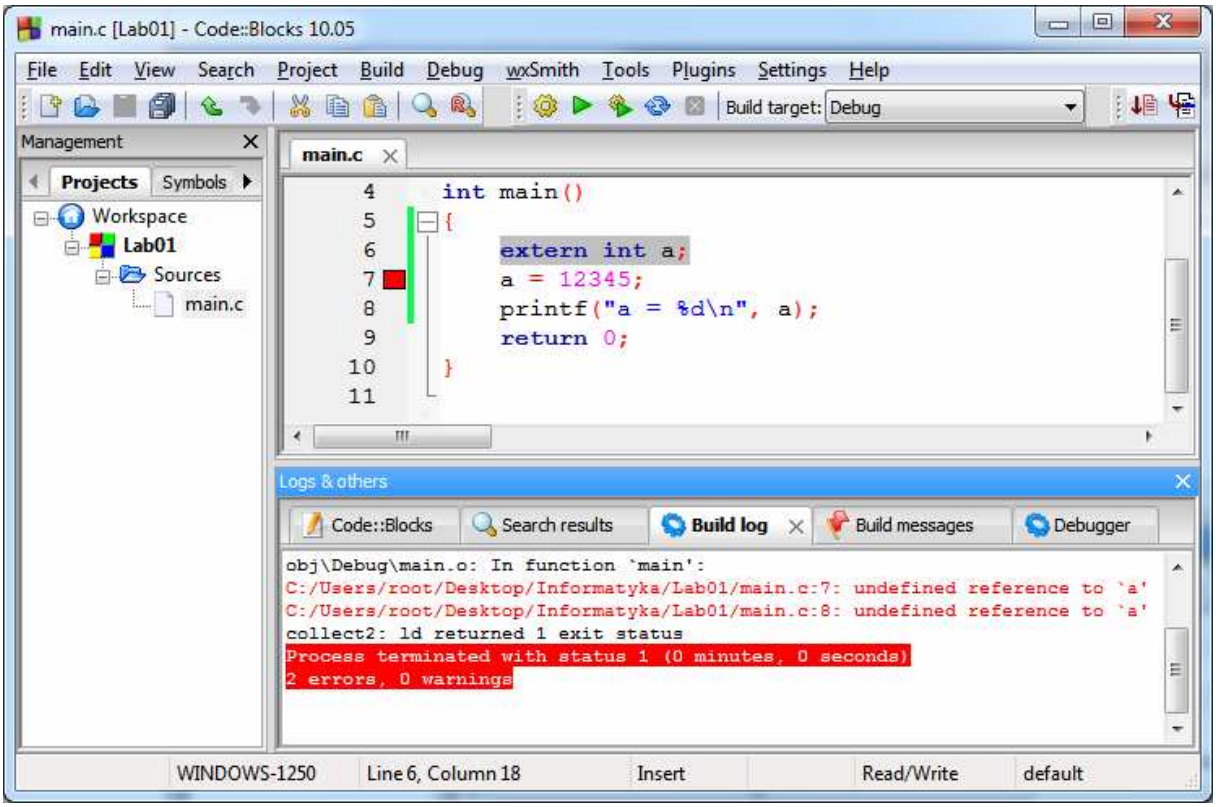


Zapis: **`extern int a;`** oznacza, że informujemy kompilator o istnieniu gdzieś w innym miejscu projektu zmiennej całkowitej (int) o nazwie „a”. Kompilator „wierzy” w nasze zapewnienia i kompiluje kod. Gdy do akcji wkracza linker, nadchodzi chwila prawdy: nigdzie w projekcie nie ma zmiennej o tej nazwie! Jest to sygnalizowane błędem:

```
obj\Debug\main.o||In function `main':
main.c|7|undefined reference to `a'
```

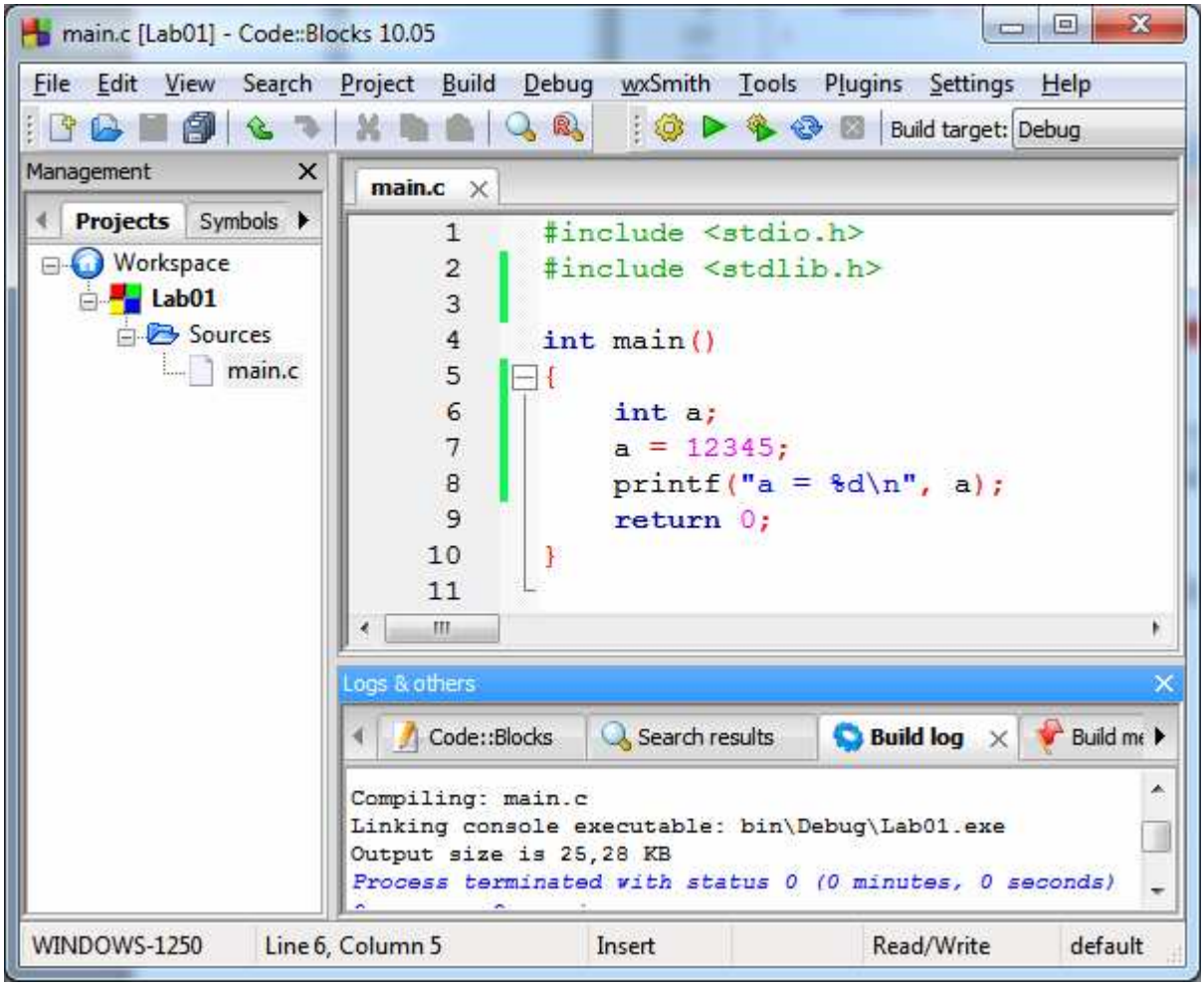
Zwróć uwagę na nazwę pliku main.o – co to za plik?

Kiedy wybierzemy zakładkę „Build messages”, widzimy:

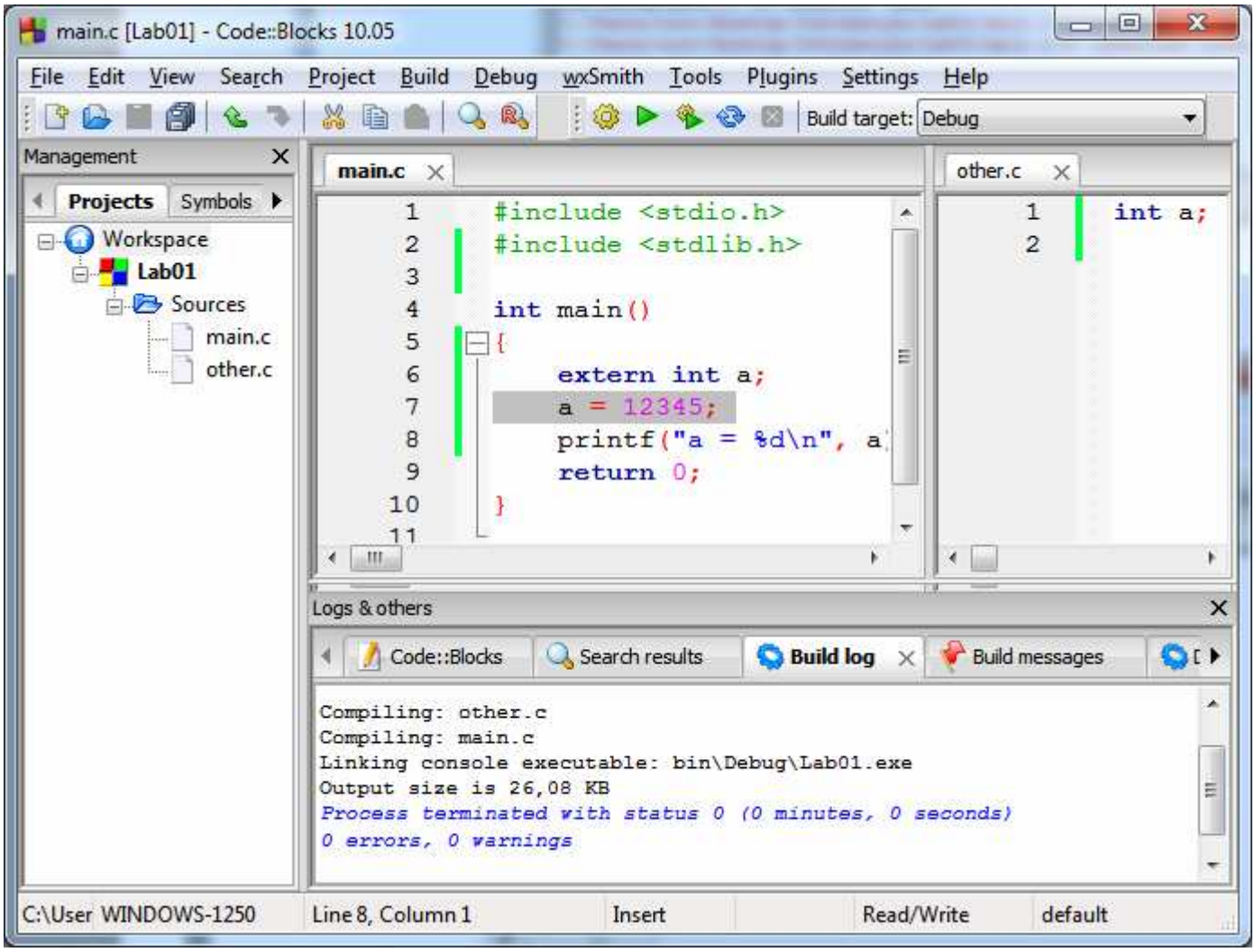


'ld' oznacza tutaj, że błąd został zasygnalizowany przez linker.

Poprawiony kod powinien wyglądać tak (jest to jedno z wielu rozwiązań):



W bardziej ogólnym przypadku można też dodać kolejny plik do projektu (tutaj o nazwie other.c) i w nim umieścić zmienną int a;

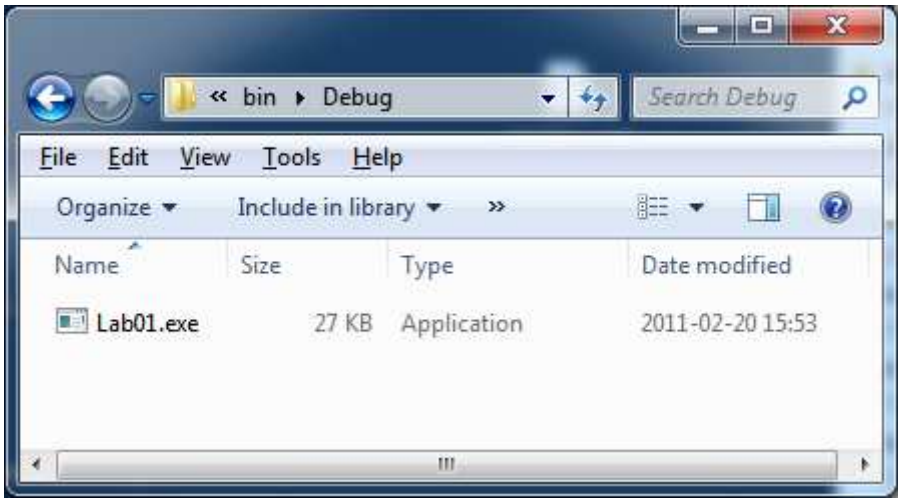


W powyższym przykładzie widać wyraźnie kolejne etapy powstawania pliku EXE:

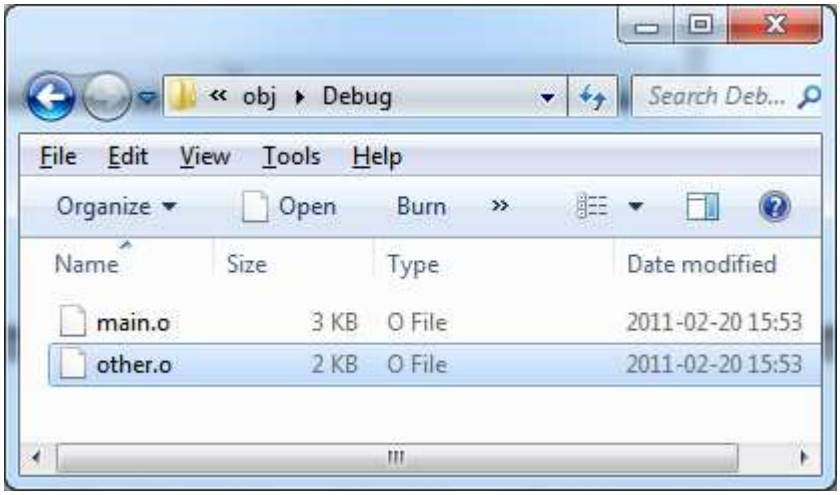
1) kompilacja other.c,

2) kompilacja main.c,

3) konsolidacja (linkowanie) plików pośrednich do postaci pliku końcowego (EXE)

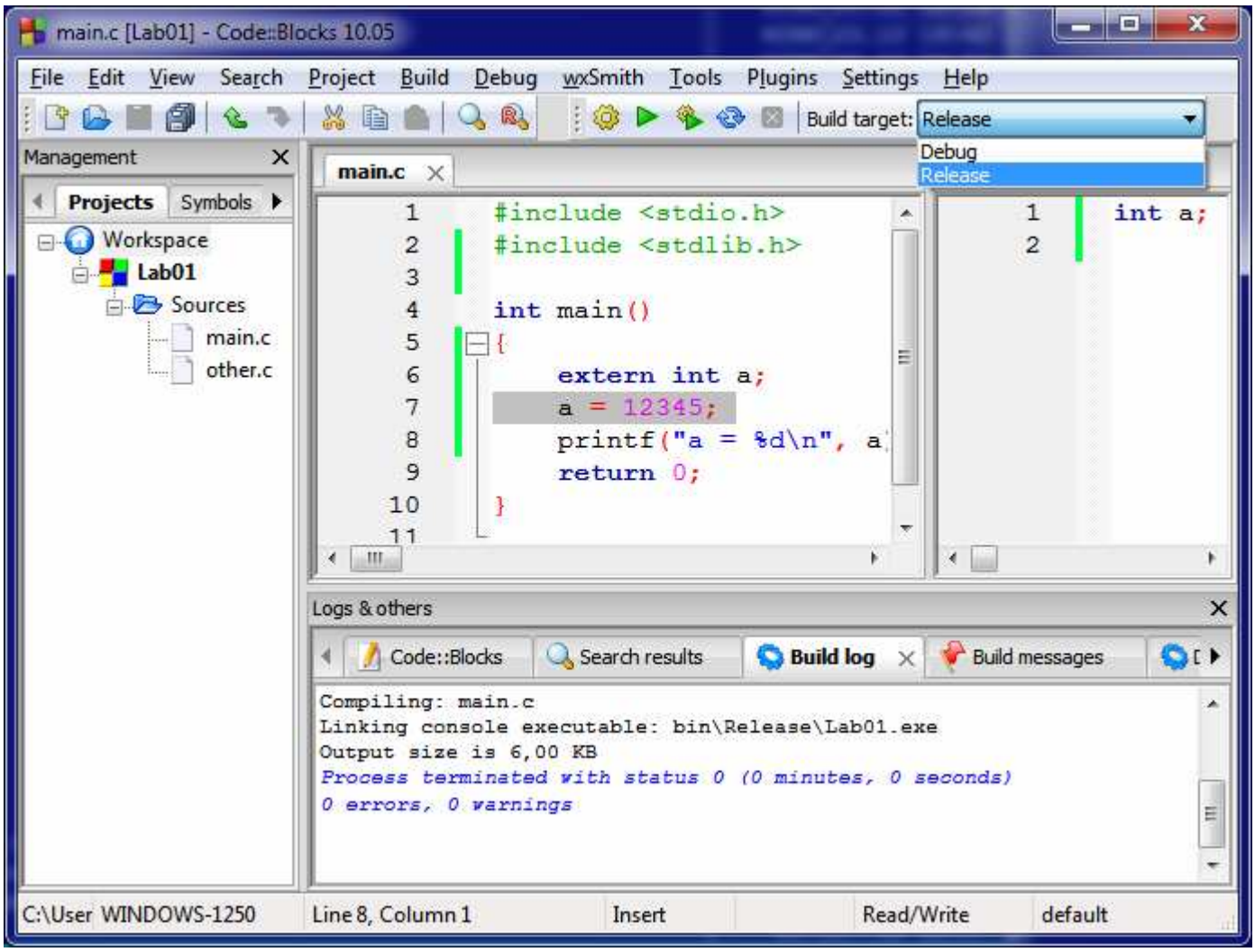


Pliki pośrednie (.o lub .obj) można podejrzeć w podkatalogu obj:



Efekt końcowy (*.exe) nie jest prostą sumą (sklejeniem) plików *.o, co widać chociażby po rozmiarach plików: 3+2 KiB nie jest równe 27 KiB. Dodatkowe kilobajty danych pochodzą ze standardowych bibliotek języka C oraz wynikają częściowo ze specyfiki formatu pliku EXE, wymaganego przez Windows.

Debug a Release



Wybranie „Build Target" Release zamiast domyślnego „Debug" da w efekcie plik EXE prawie identyczny pod względem działania, ale zauważalnie mniejszy (6 KiB zamiast 27 KiB).

1) Jakie dodatkowe informacje zawiera Debug?
2) Kiedy warto używać Debug, a kiedy Release?
3) Obejrzyj (np. w Notatniku) zawartość tych plików. Pamiętaj o włączeniu zawijania wierszy. Wbrew pozorom da się znaleźć wśród „krzaczków" pewne sensowne informacje. Uwolnij swój umysł i spróbuj! To, co oglądasz, to prawie gotowy do załadowania do pamięci komputera kod maszynowy oraz pewne informacje pomocnicze, które są wymagane przez system operacyjny.