

# Programarea calculatoarelor

---

Profesor: Maria Guțu



---

Profesor: Maria Guțu

# Exemple **IDE** pentru dezvoltarea software sub diverse SO

Pentru **Mac OS**:

CodeWarrior  
Xcode, etc.

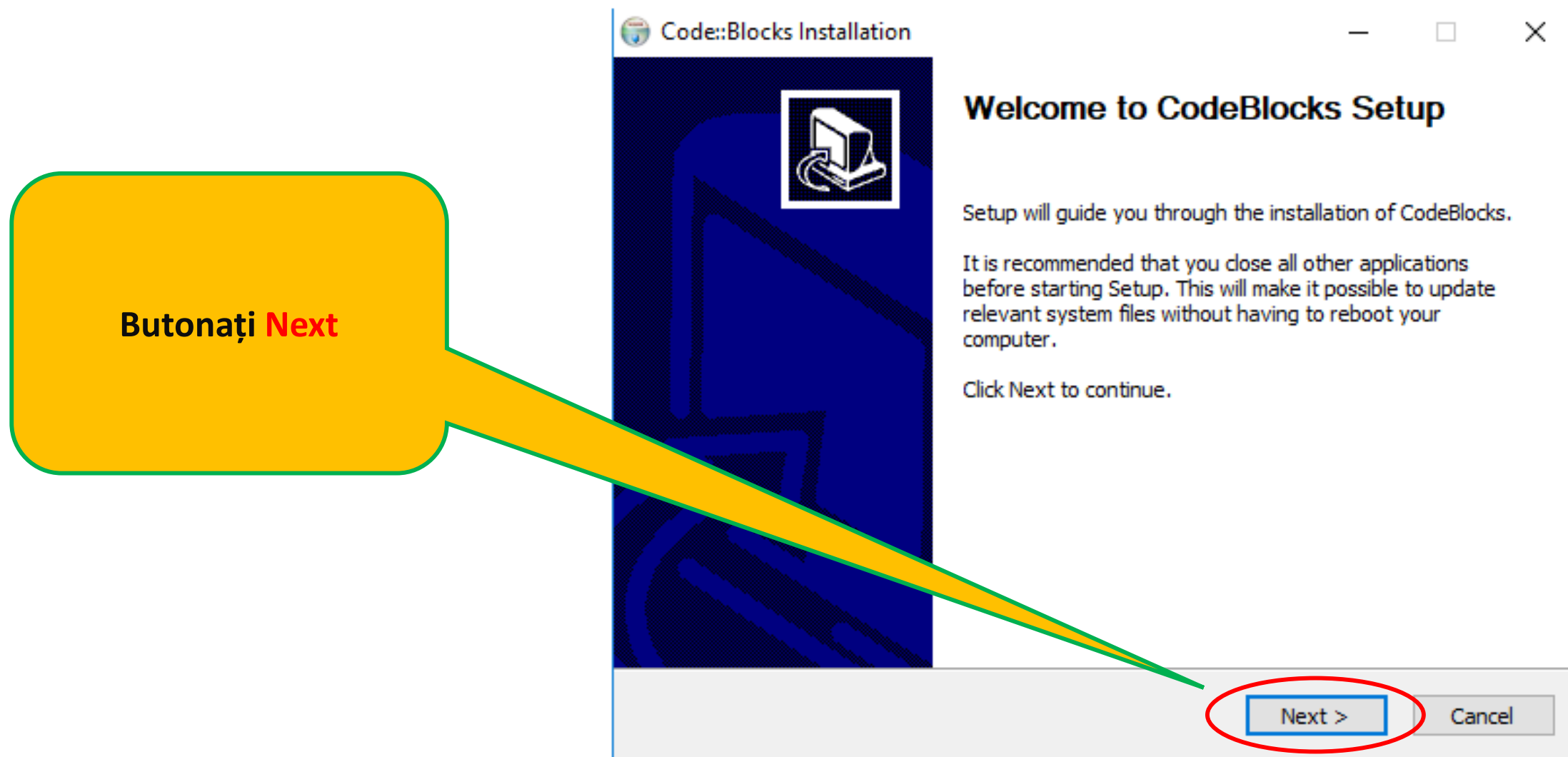
Pentru **Windows OS**:

Microsoft Visual  
Studio;  
Code::Blocks, etc.

Pentru **Linux OS**:

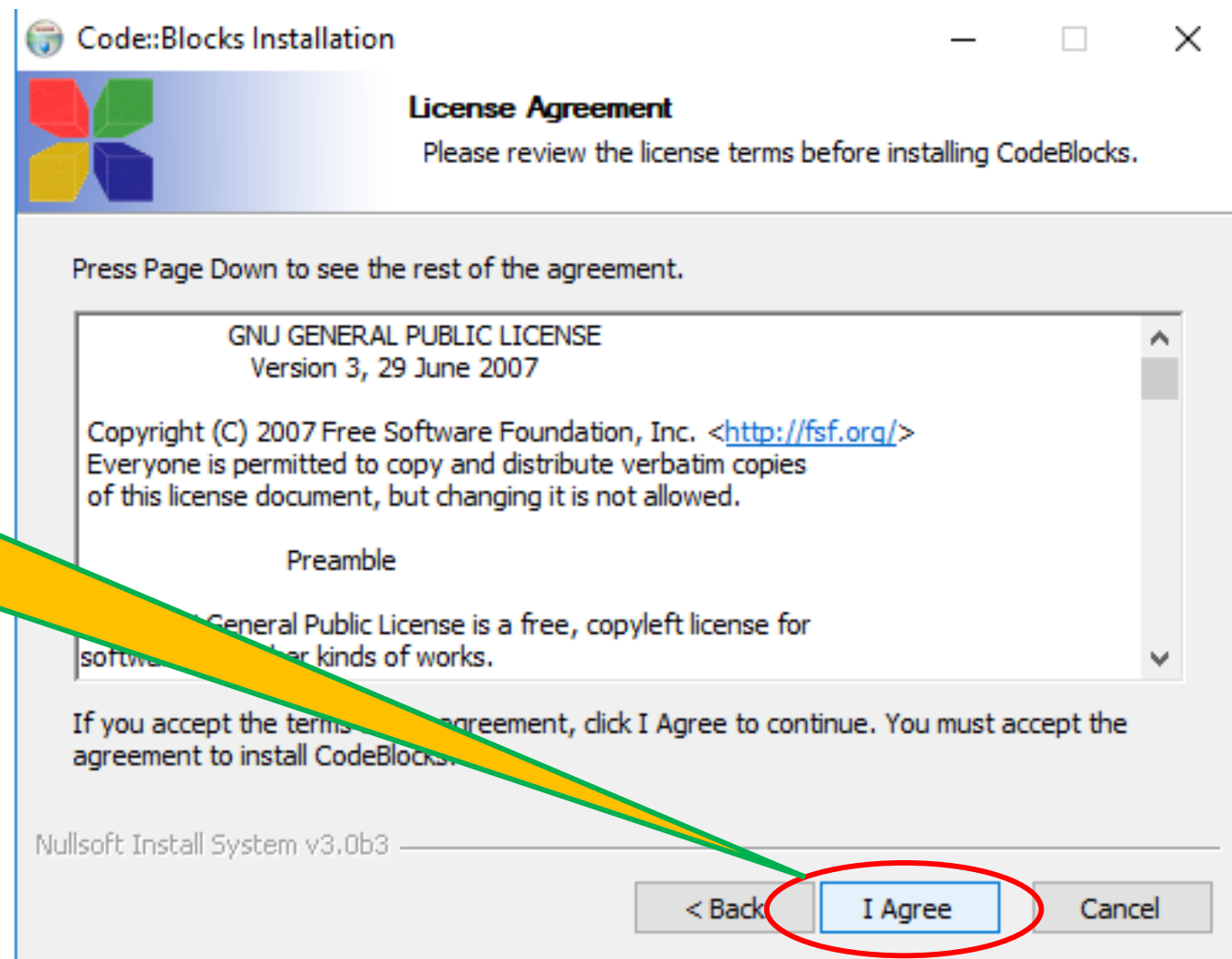
NetBeans  
Atom, etc.

# Instalarea IDE-ului Code::Blocks



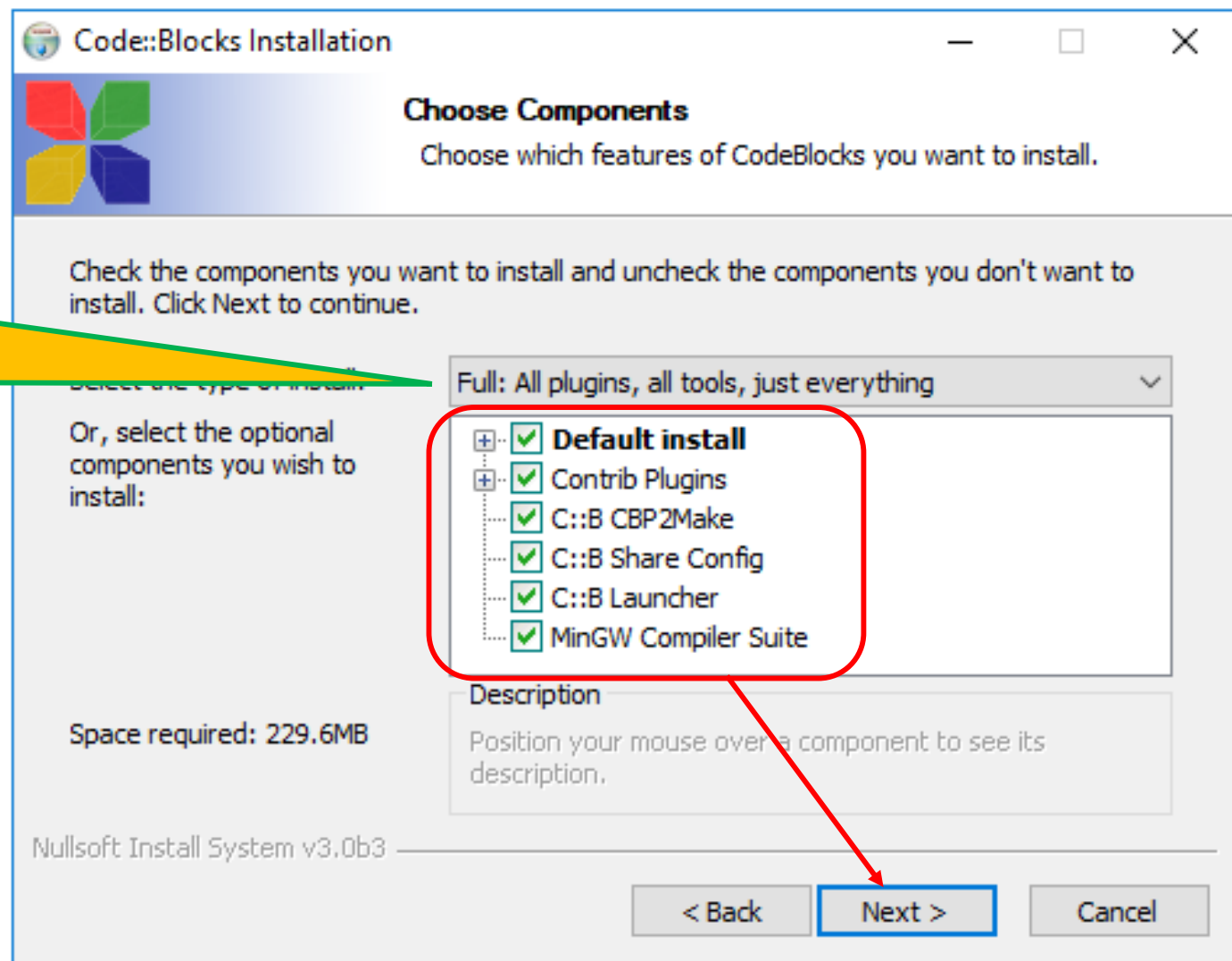
# Instalarea IDE-ului Code::Blocks

Dăm click pe **I Agree**  
după ce citim termenii  
de licență.



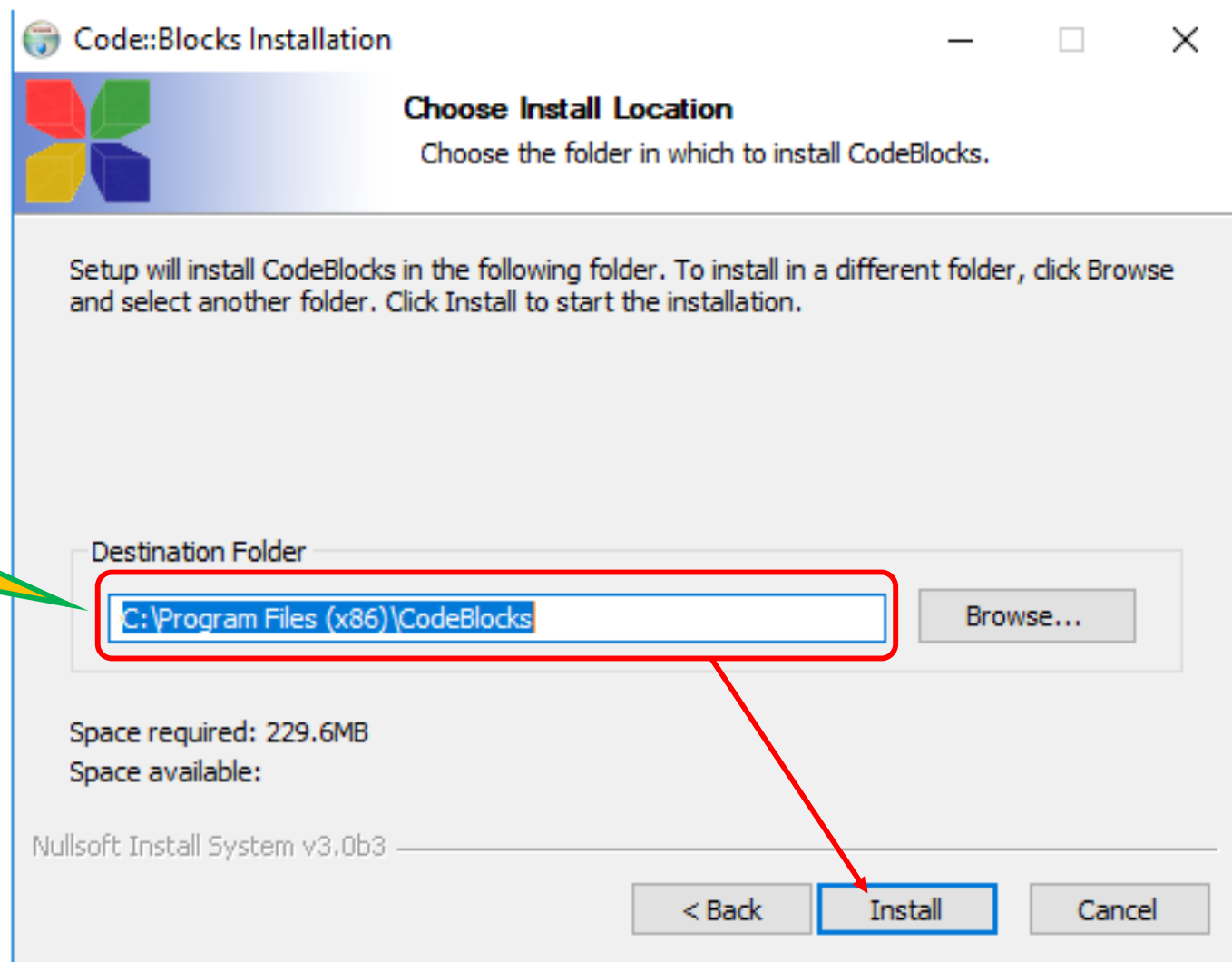
# Instalarea IDE-ului Code::Blocks

Alegem **Full Install**, apoi  
dăm click pe **Next**

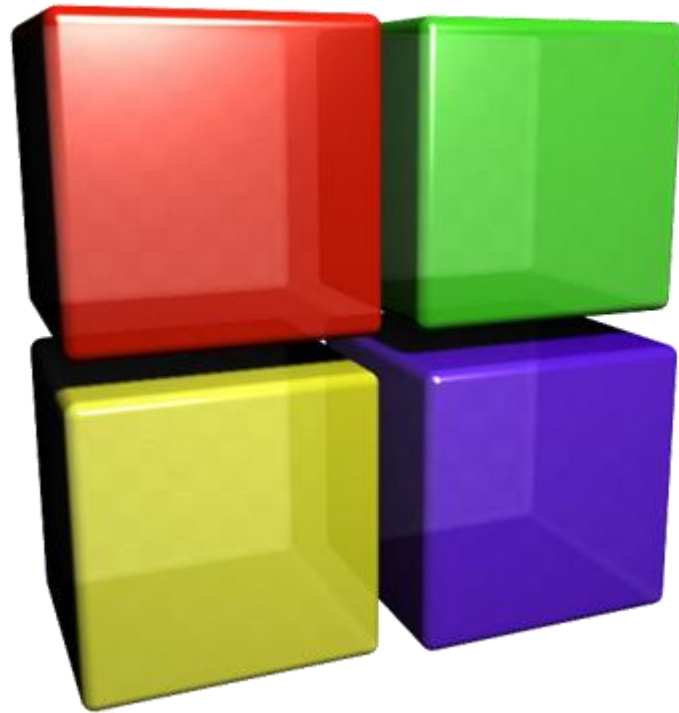


# Instalarea IDE-ului Code::Blocks

Alegem locația în care  
dorim să instalăm  
Code::Blocks



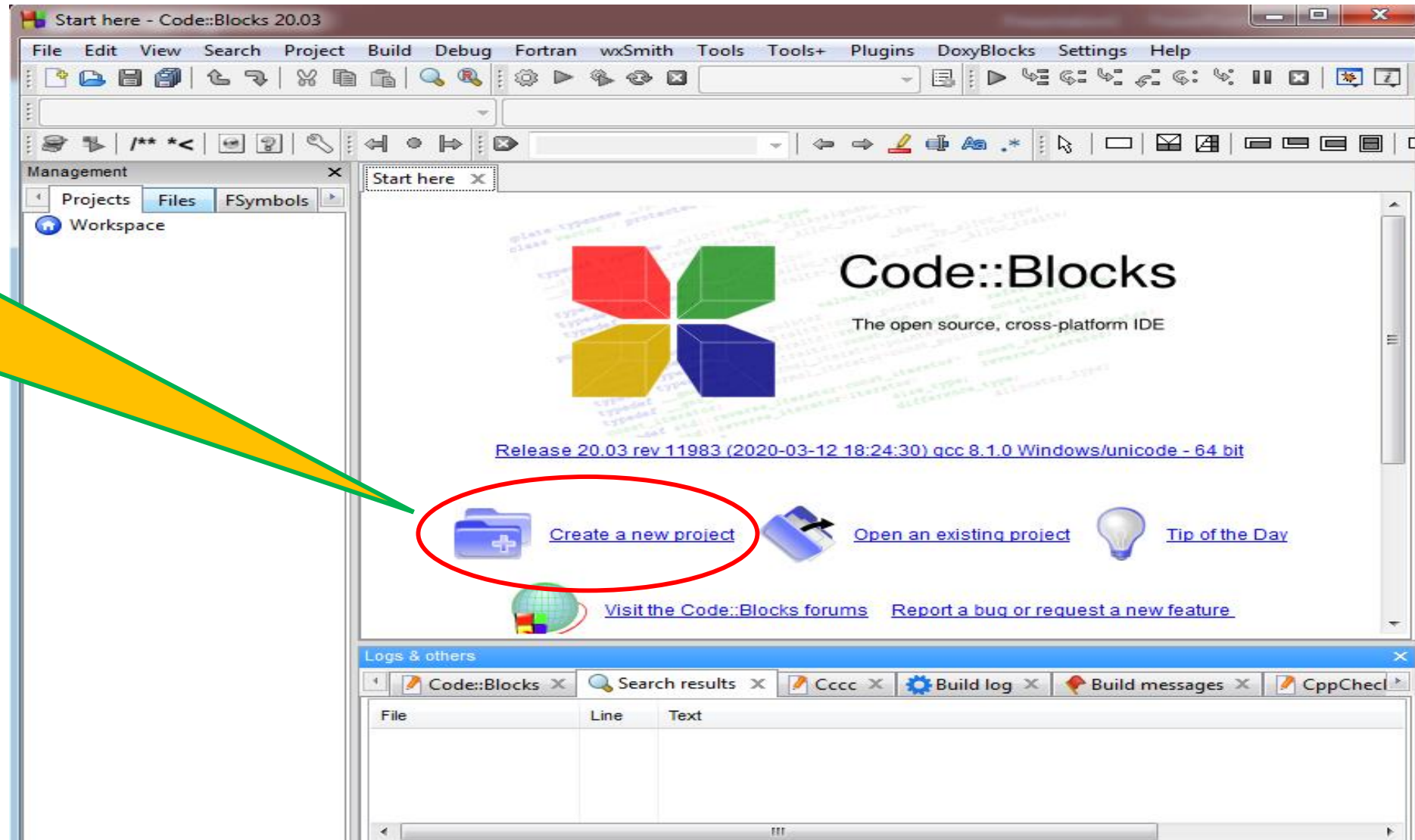
# Lansăm IDE-ului Code::Blocks





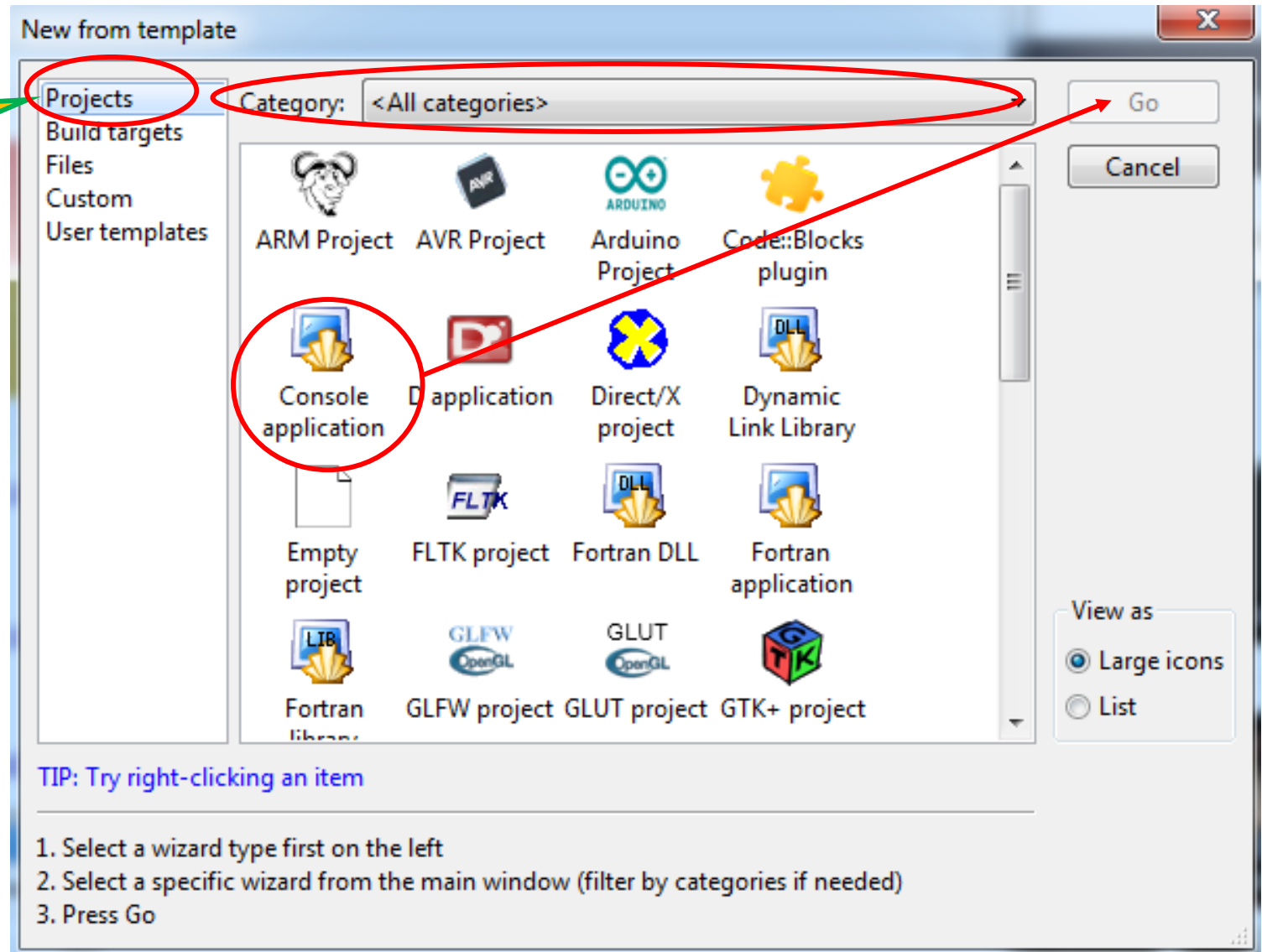
# Crearea unui proiect nou în C

**Selectați  
"Create a new project"**



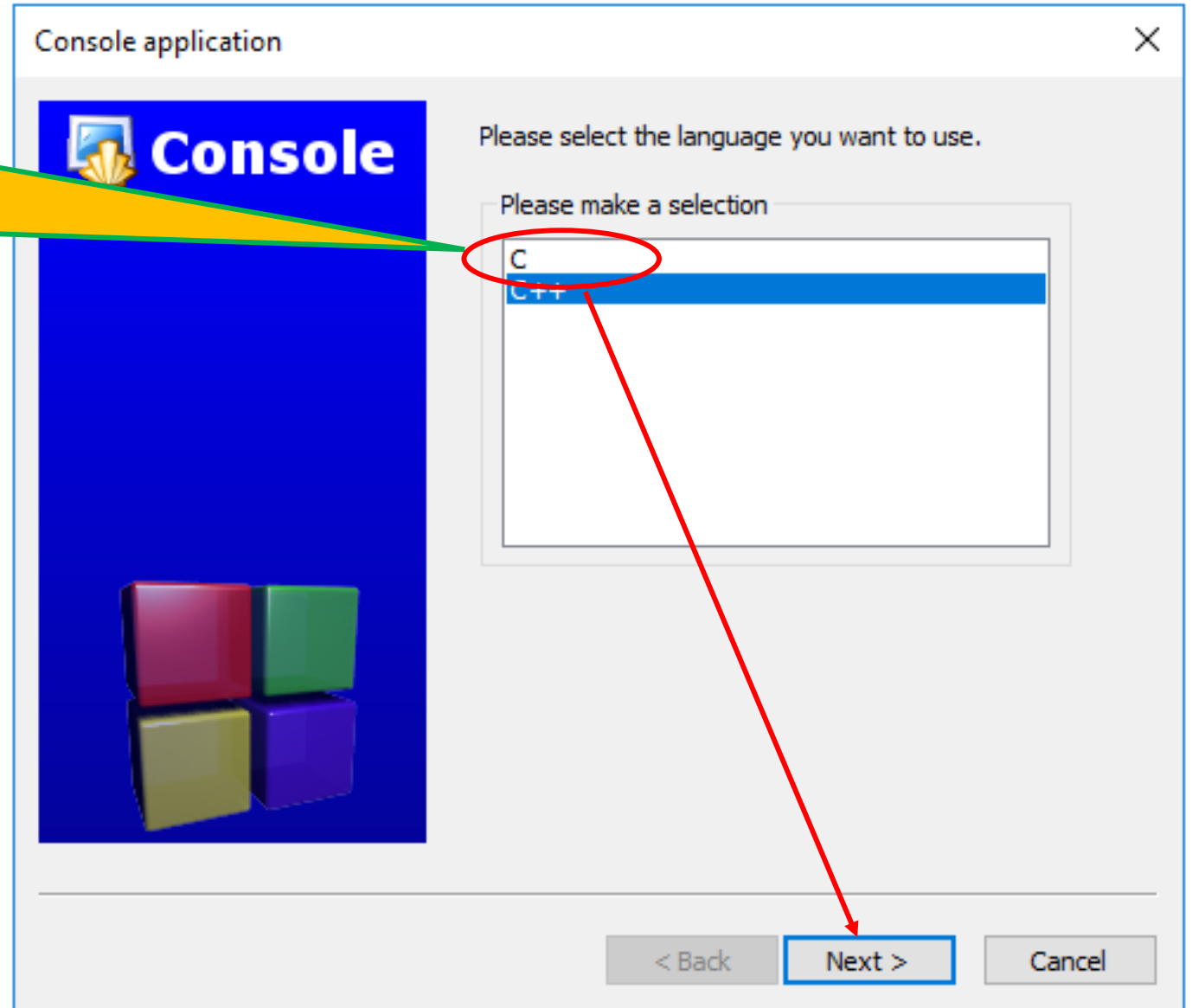
# Crearea unui proiect nou în C

Butonați **Projects** din partea stângă, selectați **All Categories** în secțiunea "Category", apoi dați click pe **Console Application** și apoi pe **GO**



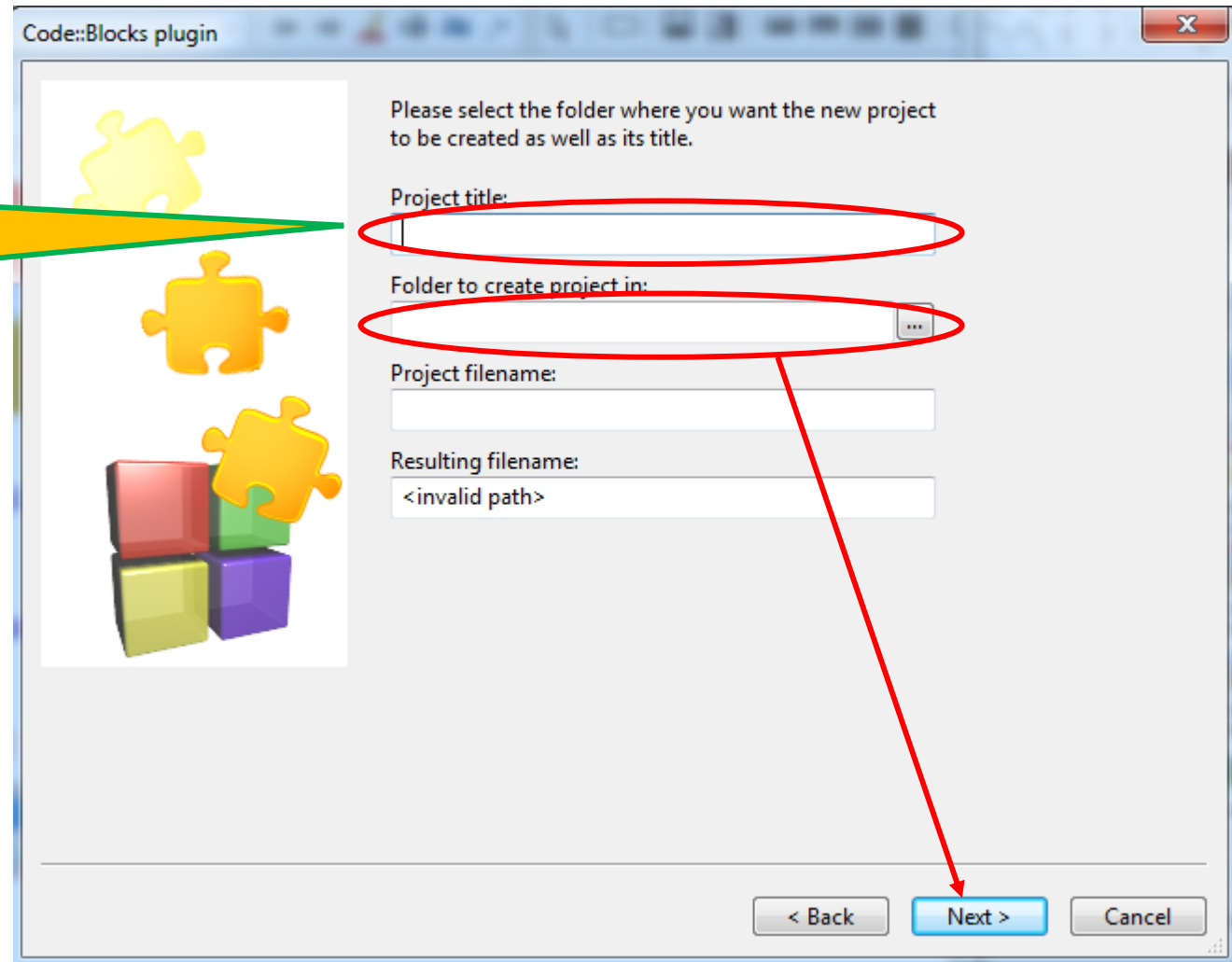
# Crearea unui proiect nou în C

Selectăm **C** și butonăm  
**Next**



# Crearea unui proiect nou în C

Completați câmpurile:  
Titlu: **Primul Program**  
Apoi, alegem unde să  
se creeze folder-ele  
proiectului după care  
butonăm **NEXT**



The screenshot shows the 'Code::Blocks plugin' dialog box. On the left, there are three puzzle pieces (yellow, orange, and a stack of red, green, and purple) and a stack of four colored cubes (red, green, yellow, and purple). The main area contains the following fields and instructions:

- Instruction: "Please select the folder where you want the new project to be created as well as its title."
- Field: "Project title:" with a text input field.
- Field: "Folder to create project in:" with a text input field and a browse button (three dots).
- Field: "Project filename:" with a text input field.
- Field: "Resulting filename:" with a text input field showing "<invalid path>".

At the bottom, there are three buttons: "< Back", "Next >", and "Cancel". A red arrow points from the "Folder to create project in:" field to the "Next >" button.

# Crearea unui proiect nou în C

Ne asigurăm că avem  
selectat ca compilator  
**GNU GCC Compiler**,  
bifați crearea  
configurațiilor **Debug**  
și **Release**, apoi  
butonați **Finish**

Console application

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

# Structura unui program C

<b>Header/antet</b>	<b>#include &lt;stdio.h&gt;</b>
<b>main()</b>	<b>int main() {</b>
<b>Partea declarativă</b>	<b>int a = 15;</b>
<b>Corpul programului</b>	<b>printf(“%d”, a);</b>
<b>Return</b>	<b>return 0; }</b>

# Includerea fișierelor antet

Nume fișier antet	Descrierea
<b>stddef.h</b>	definește mai multe tipuri și macro-uri utile;
<b>stdint.h</b>	definește tipurile de întregi exacte de lățime;
<b>stdio.h</b>	definește funcțiile de intrare și ieșire de bază;
<b>stdlib.h</b>	definește funcțiile de conversie numerică, generatorul de rețea pseudo-aleatoriu, răspunde de alocarea memoriei;
<b>string.h</b>	definește funcțiile de manipulare a șirurilor;
<b>math.h</b>	definește funcțiile matematice comune.

## Sintaxa de includere a fișierelor antet

```
#include <nume_fisier_antet.h>
```



# Declarația metodei principale (main)

Sintaxa funcției main() este:

```
int main()
```

```
{ ...
```

```
}
```

# Declararea variabilelor

```
int main()  
{ int x;  
  float y = 12.3, z;  
  char ch;  
  
  ...  
}
```

## Corpul programului

```
int main()  
{ int x = 15;  
  printf("%d", x);  
  ...  
}
```

## Declarația de returnare

```
int main()  
{ int x = 15;  
  printf("%d", x);  
  return 0;  
}
```

# Specificatori de format

Specificatori format	Descriere
<b>%d</b>	întreg zecimal cu semn
<b>%i</b>	întreg zecimal, octal (0) sau hexazecimal (0x, 0X)
<b>%o</b>	întreg în octal, fără 0 la început
<b>%u</b>	întreg zecimal fără semn
<b>%x, %X</b>	întreg hexazecimal, fără 0x/0X; cu a-f pt. %x, A-F pt. %X
<b>%c</b>	caracter
<b>%s</b>	șir de caractere, până la '\0' sau nr. de caractere dat ca precizie

# Specificatori de format

<b>%f, %F</b>	real fără exp.; precizie implicită 6 poz.; la precizie 0: fără punct real (posibil cu exponent)
<b>%e, %E</b>	numere reale cu mantisă și exponent (al lui 10); precizie implicită 6 poz.; la precizie 0: fără punct
<b>%g, %G</b>	numere reale în format %f sau %e, funcție de valoare real, ca %e, %E dacă exp. < -4 sau precizia; altfel ca %f. Nu tipărește zerouri sau punct zecimal în mod inutil
<b>%p</b>	pointer, în formatul tipărit de printf
<b>%ld, %li</b>	numere întregi lungi
<b>%lf, %le, %lg</b>	numere reale în precizie dublă ( <i>double</i> )
<b>%Lf, %Le, %Lg</b>	numere reale de tip <i>long double</i>
<b>%%</b>	caracterul procent

# Descriptori de format

%d, %f, %lf, %c, %s

## Tipuri de date

**char** – un singur caracter;

**int** – numere întregi;

**float** – numere zecimale;

**double** – numere reale cu dublă precizie;



# Exemple de program C

## Specificatorul de format: %c

```
#include <stdio.h>
int main()
{
    char ch = 'A';
    printf("%c\n", ch);
    return 0;
}
```

# Exemple de program C

## Specificatorul de format de tip întreg: %d

```
#include <stdio.h>
int main()
{
    int x = 25;
    printf("_____\n", ____);
    return 0;
}
```

# Exemple de program C

## Specificatorul de format de tip float: %f

```
#include <stdio.h>
int main()
{
    float nota = 9.89;
    printf("_____\n", ____);
    return 0;
}
```

# Exemple de program C

## Specificatorul de format de tip float: %f

```
#include <stdio.h>
int main()
{
    float z = 12.345673;
    printf("%10f\n", z);
    return 0;
}
```

```
#include <stdio.h>
int main()
{
    float z = 12.345623;
    printf("%-10f\n", z);
    return 0;
}
```

# Exemple de program C

## Specificatorul de format de tip float: %f

```
#include <stdio.h>
int main()
{
    float z = 12.345823;
    printf("%+10f\n", z);
    return 0;
}
```

```
#include <stdio.h>
int main()
{
    float z = 12.347823;
    printf("%10.3f\n", z);
    return 0;
}
```

# Exemple de program C

## Specificatorul de format a unui șir: %s

```
#include <stdio.h>
int main()
{
    char a[] = "Sunt student la UTM";
    printf("%s\n", a);
    return 0;
}
```

# Exemple de program C

## Citirea de la tastatură a unui **char**

```
#include <stdio.h>
int main()
{
    printf("Intrudu un caracter: ");
    char ch;
    scanf("%c", &ch); //input
    printf(" Caracterul introdus este: %c\n", ch);
    return 0;
}
```

# Exemple de program C

## Citirea de la tastatură a unui nr. întreg

```
#include <stdio.h>
int main()
{
    printf("Intrudu un numar: ");
    int x;
    scanf("%d", &x); //input
    printf(" Numarul introdus este: %d\n", x);
    return 0;
}
```



# Exemple de program C

## Citirea de la tastatură a unui string

```
#include <stdio.h>
int main()
{
    printf("Intrudu sirul: \n");
    char str[10];
    scanf("%s", str); //input
    printf(" Sirul introdus este: %s\n", str);
    return 0;
}
```

## Rezolvarea problemelor

1. Se consideră două numere întregi. Să se scrie un program care mărește aceste numere cu 100.

2. Să se scrie un program care citește de la tastatură un număr întreg și afișează pătratul lui.

## Rezolvarea problemelor

3. Se consideră trei numere întregi. Să se scrie un program care calculează suma și media lor aritmetică.

4. Se consideră cunoscute laturile unui dreptunghi. Să se scrie un program care calculează aria și perimetrul dreptunghiului.

## Rezolvarea problemelor

5. Să se scrie un program care calculează valoarea funcției  $y = x^2 + \left(\frac{x^3}{2}\right)^2 + 3$ .

6. Se consideră coordonatele a două puncte  $A(x_1, y_1)$  și  $B(x_2, y_2)$ . Să se scrie un program care calculează distanța dintre aceste puncte.

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

## Rezolvarea problemelor

7. Se consideră că numerele  $a, b, c$  reprezintă lungimile laturilor unui triunghi oarecare, unde  $a, b, c$  – sunt introduse de la tastatură. Să se scrie un program care calculează aria acestui triunghi, folosind formula lui Heron  $s = \sqrt{p(p-a)(p-b)(p-c)}$ , unde  $a, b, c$  sunt laturile triunghiului, iar  $p$  este semiperimetrul lui ( $p=(a+b+c)/2$ ).