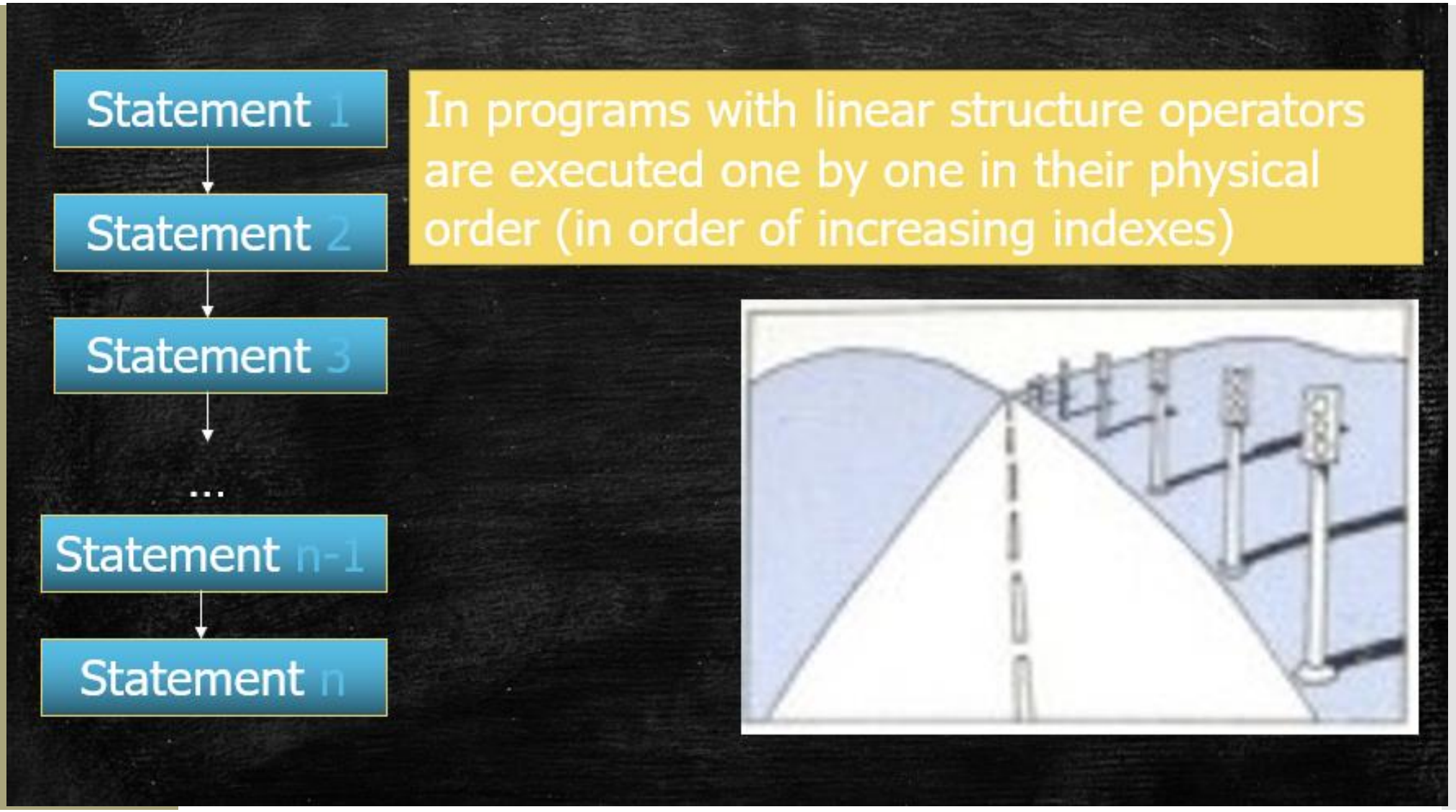
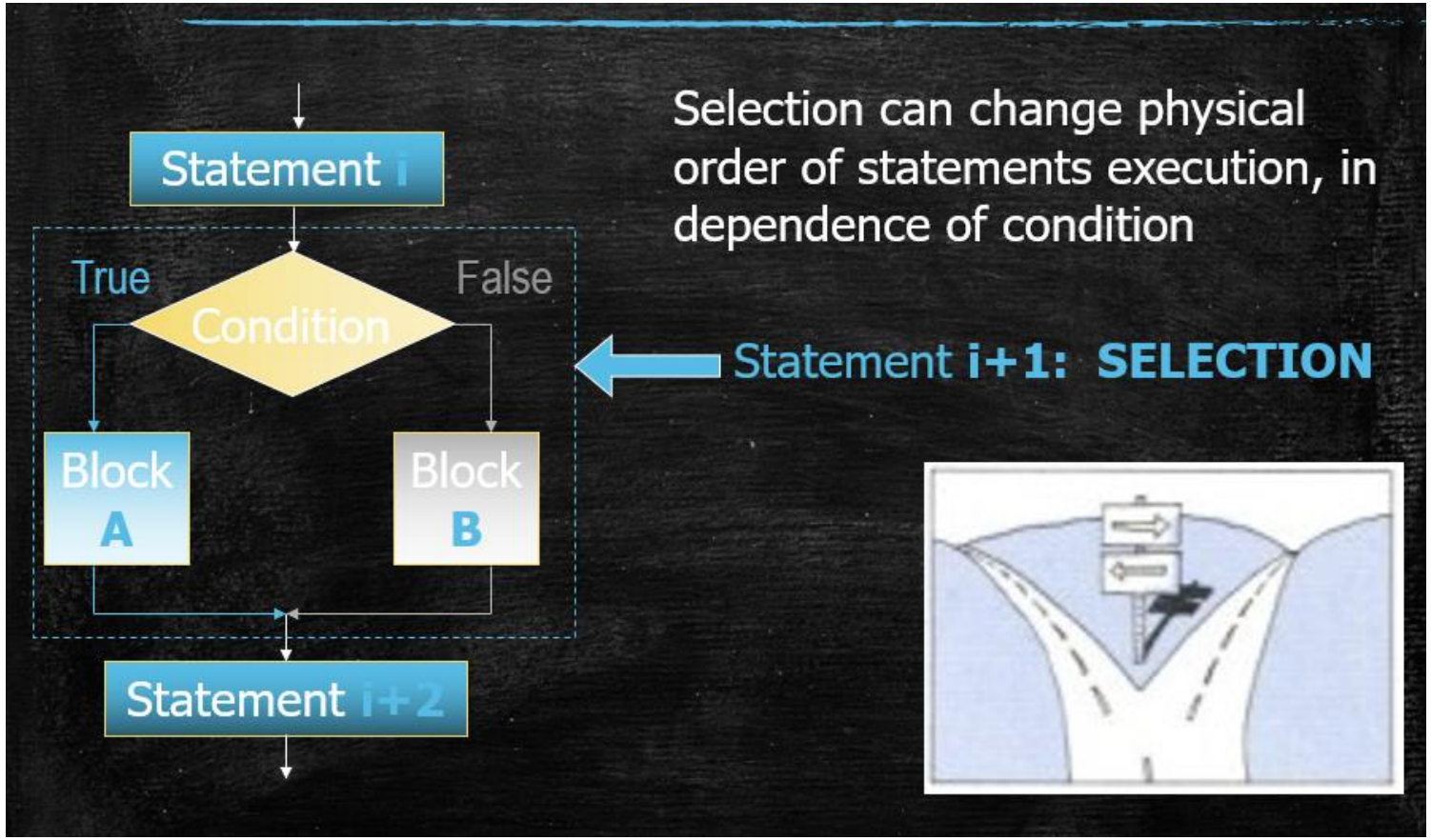
Lesson 3

Selection: if , ?, switch

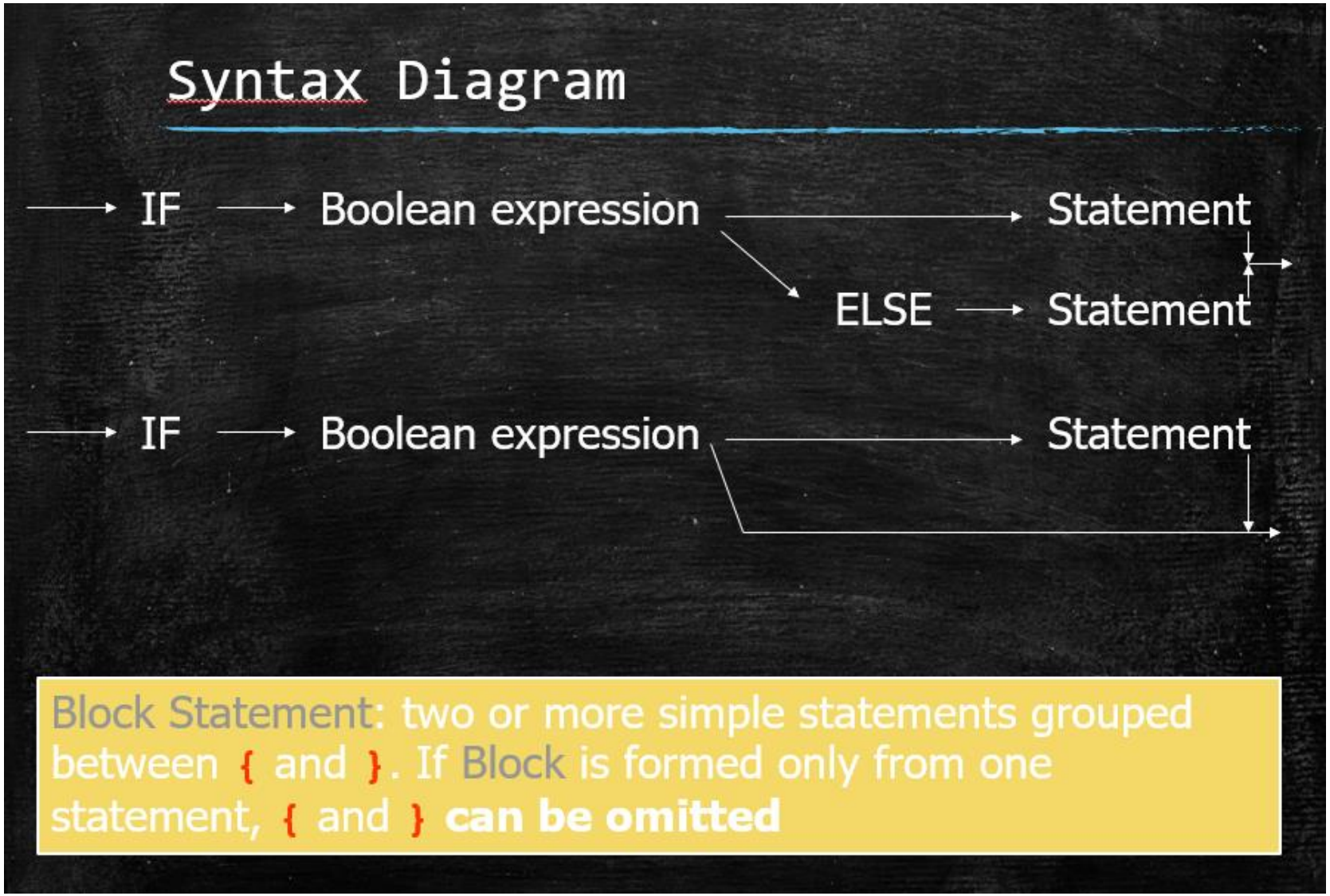
Linear programs



Selection


Statement i
True
Condition
False
Block A
Block B
Statement i+2
Selection can change physical order of statements execution, in dependence of condition
Statement i+1: SELECTION

If. Syntax diagram



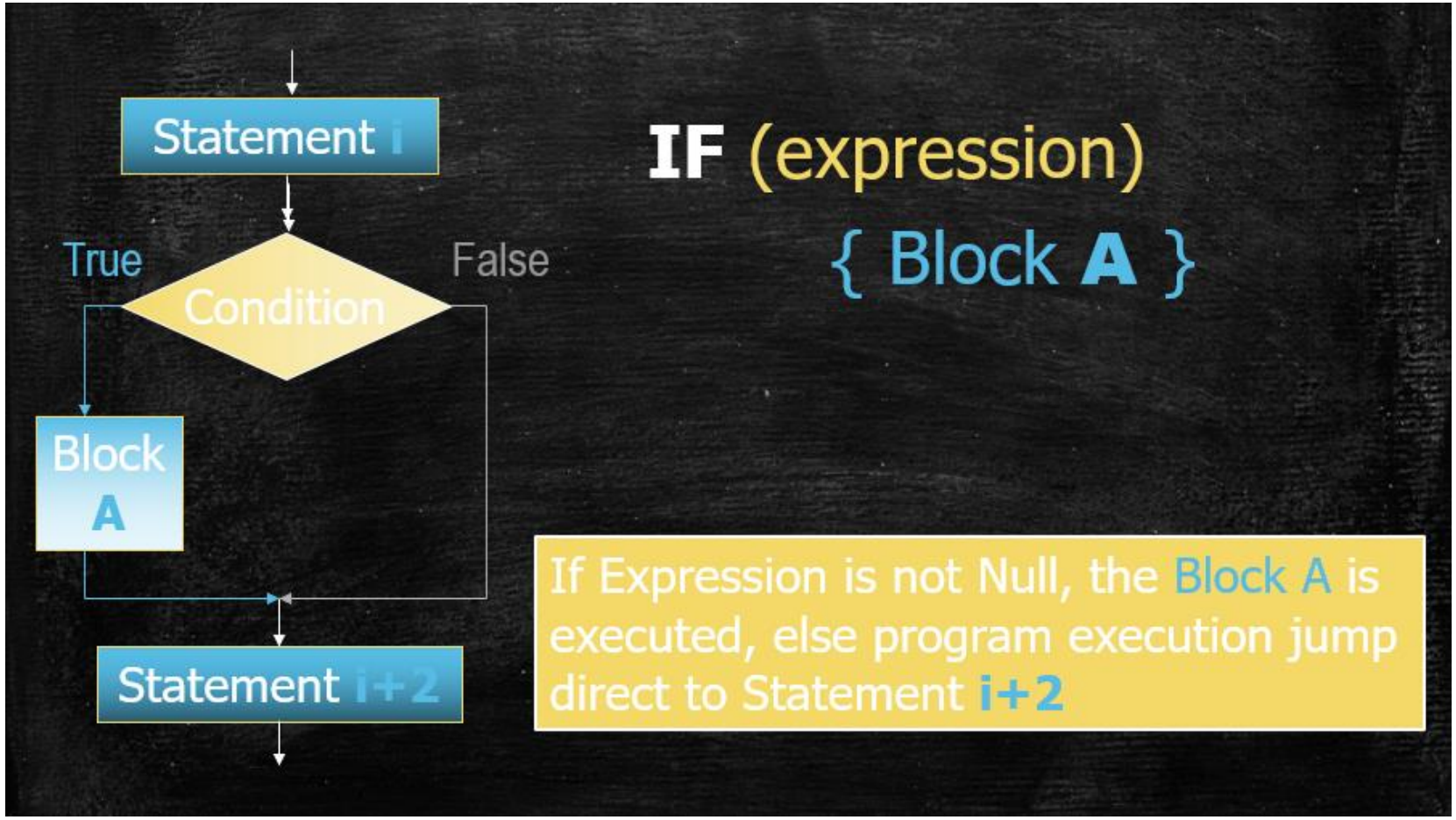
Maximal from two numbers

```
#include <stdio.h>

int a,b,m;

int main()

{ printf("Input a,b:");
  scanf("%d%d", &a,&b);
  if (a > b) m = a;
       else m = b;

printf ("%d ",m);

return 0;

}
```

Dividers of 13

```
#include <stdio.h>

int a;

int main()

 { printf("Input a:");
   scanf("%d", &a);
   if (a % 13)
          printf("ND");
   else   printf("D");
   return 0;

}
```

If – special forms



If operation

Selection operators are '**?**' and '**:**'. Both used together.
Format: **`< expresie_1 > ? <expresie_2> : <expresie_3>`**
expresie_1 is evaluated. If is no-zero, **expresie_2** is executed.
Else (if is equal to 0) **expresie_3** is executed.
Examples :

Expresie	**Valoare**
`x > y ? z=x : z=y`	**z** - maxim dintre x şi y
`x >= 0 ? z=x : z=-x`	**z** - modulul lui x

Logical operators

AND (&&)		
X	Y	X && Y
False	False	False
False	True	False
True	False	False
True	True	True

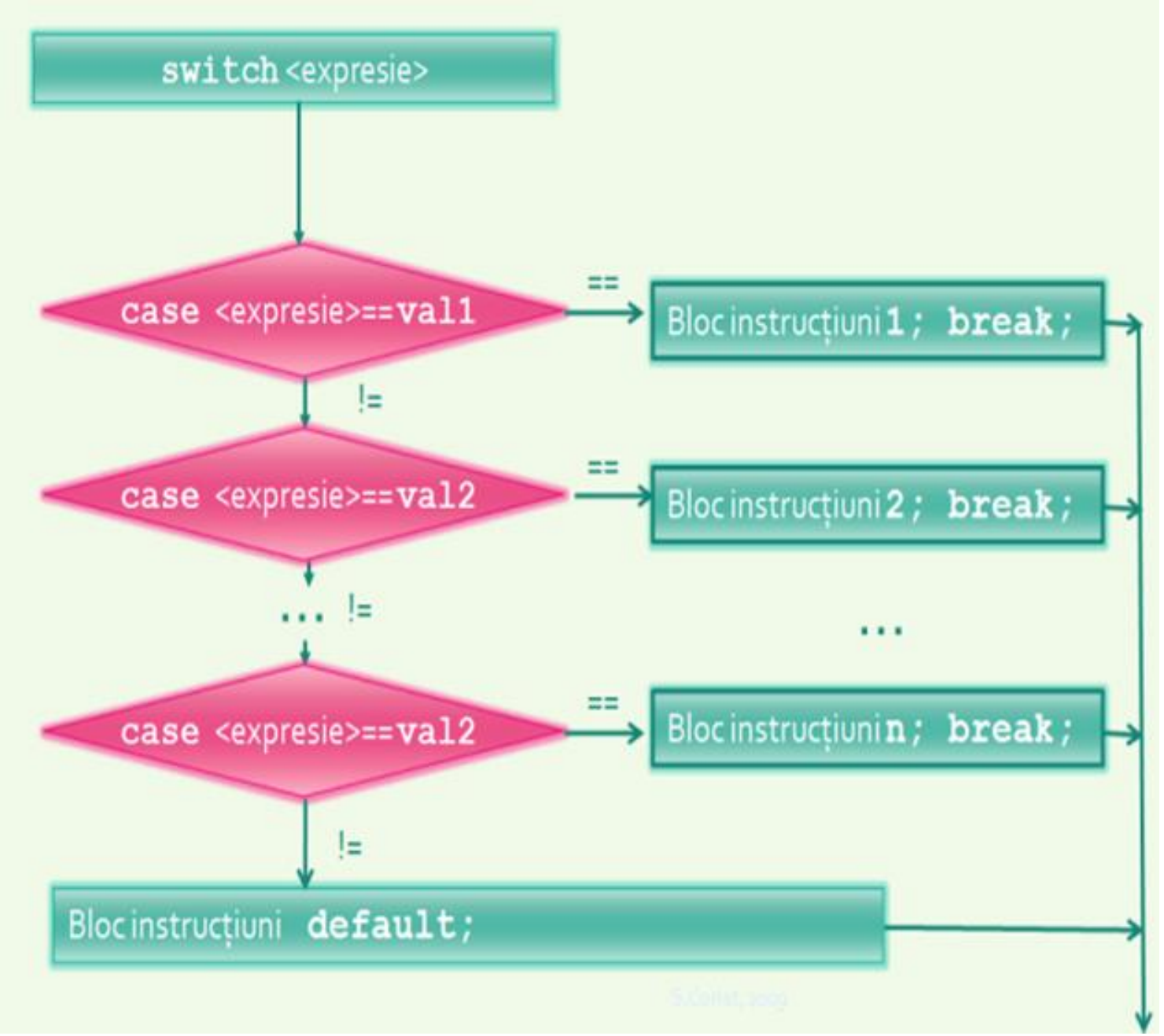
OR (||)		
X	Y	X || Y
False	False	False
False	True	False
True	False	False
True	True	True

NOT (!)	
X	! X
False	True
True	False

Scrieți un program care va determina dacă 3 numere introduse pot forma lungimile laturilor unui triunghi oarecare

```
#include <stdio.h>

int main()
{
   float a,b,c;

   printf("Sides:  ");
   scanf("%f %f %f", &a, &b, &c);

   if (a + b > c && a + c > b && b + c > a)
   printf("Triunghi\n");
   else printf("Nu este Triunghi\n");
   return 0;
}
```

switch


switch <expresie>
case <expresie>==val1
==
Bloc instrucțiuni 1; break;
!=
case <expresie>==val2
==
Bloc instrucțiuni 2; break;
... !=
...
case <expresie>==val2
==
Bloc instrucțiuni n; break;
!=
Bloc instrucțiuni default;

switch

```
switch (expresion)
{
 case <expr. 1> : <commands block 1> break;
 case <expr. 2> : < commands block 2> break;
 …
 case < expr. n> : < commands block n> break;
 default: < commands block n+1 >  break;
}
```

Example: calculator

```
#include <stdio.h>
int a;

int main()
{
  float a,b,c;   char q;

  printf("Operation:");
  q = getchar();
  printf("Operators:  ");
  scanf("%f %f", &a, &b);

  switch(q)
  {     case '+' : c=a+b; break;
    case '-' : c=a-b; break;
    case '*' : c=a*b; break;
    case '/' : c=a/b; break;
  }
  printf("%f %c %f = %f\n", a,q,b,c);
  return 0;
}
```

Problems

Problem 1:

Roman numbers of length two

A correct roman number, formed from two digits is given.

Write a program to find it's arabic equivalent.

Example: VI – 6, IX - 9

Problem 2

Calculate solutions of quadratic equation using switch / case

The solution (problem 2)

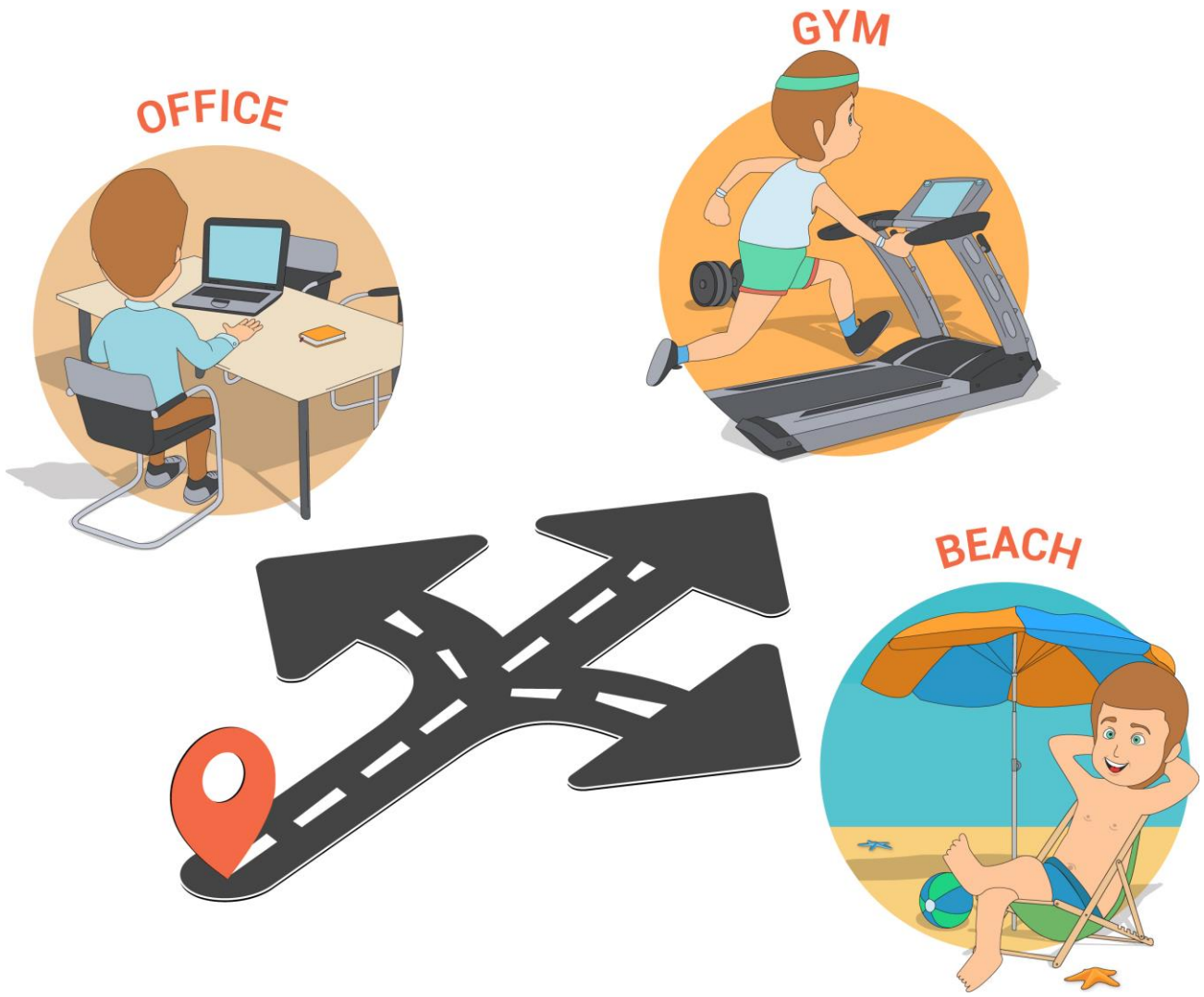
```
#include <stdio.h>
#include <math.h>
int main()
{
        float a, b, c;
        float root1, root2, imaginary;
        float discriminant;
        printf("Enter values of a, b, c : ");
        scanf("%f %f %f", &a, &b, &c);
        /* Calculate discriminant */
        discriminant = (b * b) - (4 * a * c);
```

The solution (problem 2)

```
discriminant = (b * b) - (4 * a * c);
switch(discriminant > 0)
{
        case 1:
                root1 = (-b + sqrt(discriminant))
                        / (2 * a);
                root2 = (-b - sqrt(discriminant))
                        / (2 * a);
                printf("Two real roots: %.2f and %.2f",
                root1, root2);
                break;
```

The solution (problem 2)

```
        case 0:
                switch(discriminant < 0)
                {
                case 1:
                  root1 = root2 = -b / (2 * a);
                  imaginary = sqrt(-discriminant) / (2 * a);
                  printf("Two distinct complex roots:
                  %.2f + i%.2f and %.2f - i%.2f",
                  root1, imaginary, root2, imaginary);
                  break;
                case 0:
                  root1 = root2 = -b / (2 * a);
                  printf("Two equal real roots :
                  %.2f and %.2f", root1, root2);
                  break;
                }
        }
        return 0;
}
```


OFFICE
GYM
BEACH