

CSC10001 – Introduction to Programming

3rd lecture: Flowchart – Control structure: Selection

fit hcmus

Instructor: TRAN Thi Thao Nhi

October 17, 2023

Algorithm

Pseudocode

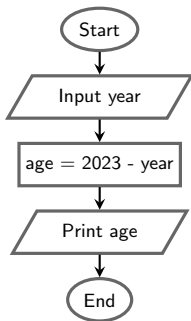
Program

Task: calculate the age of a person

- ▶ display a message on the screen asking "When were you born?"
- ▶ wait for the user to enter the number of year born
- ▶ store user's number in memory
- ▶ subtract user's born year from the current year
- ▶ store the result in memory
- ▶ display the result on the screen

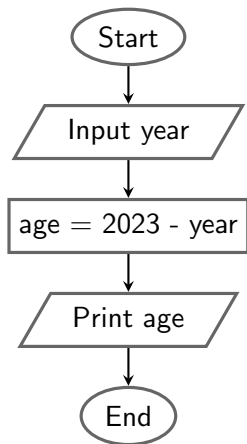
Input born year
Calculate
 $\text{age} = 2023 - \text{year}$
Print the age

Flowchart



```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int year, age;
6     cout << "When were
7         you born? ";
8     cin >> year;
9     age = 2023 - year;
10    cout << "You are "
11        << age << endl;
12    return 0;
13 }
```

Flowchart



A flowchart is a diagram that shows the “flow of control” (the order where instructions are executed) of a program

ellipse

parallelogram

rectangle

diamond

terminal

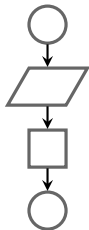
input/output

process/activity

decision

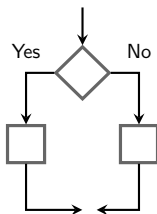
Control structures

A computer can process a program in one of the following ways



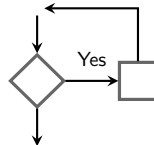
Sequence structure

- ▶ follows the statements in order



Selection structure

- ▶ executes particular statements depending on condition(s)

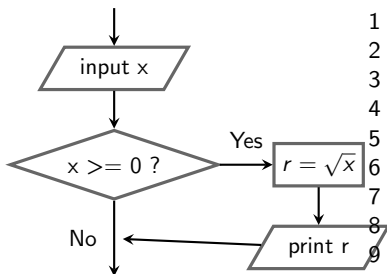


Repetition structure

- ▶ repeats particular statements a certain number of times based on condition(s)

Selection structure

Task: calculate the square root of an input real number



```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 int main(){
6     double x, sqrt_x;
7     cout << "Enter real number x: ";
8     cin >> x;
9     if (x >= 0) {
10         sqrt_x = sqrt(x);
11         cout << "sqrt(x) = "
12              << sqrt_x << "\n";
13     }
14     return 0;
15 }
```

Selection (conditional / branch) statement:

```
if (expression)
    statement;
```

Relational expression

Relational (logical / boolean) expression

- ▶ represents condition
- ▶ has value true or false (logical / boolean values)
- ▶ can be created by relational operators `>` `<` `>=` `<=` `==` `!=`
(left-to-right associativity)

`8 < 15` `true`

`1 == 9.0 >= 9.0` `true`

`6 != 6.0` `false`

`8 > '5'` `false`

`6.0 == 6.000000000000001` `false`

`5` `true`

`0.0` `false`

`-6 * 2 + 7` `true`

Note: Relational expressions have a higher precedence than the assignment operator

Expanding if statement

The `if` statement can conditionally execute a block of statements (*block / compound statement*) enclosed in braces

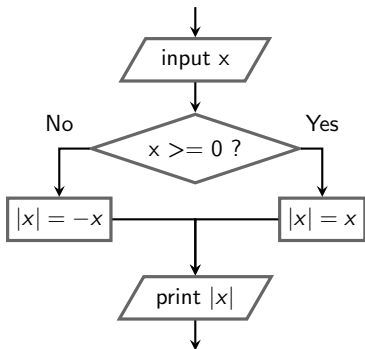
```
if (expression)
{
    statement; // do first
    statement; //      second
    ...
}
```

Statements in the block are executed sequentially

Reminder: Variables defined in a code block can only be seen inside that block (variable's scope)

if else statement

Task: calculate the absolute value of an input integer number



```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 int main(){
6     int x, abs_x;
7     cout << "Enter integer number x:";
8     cin >> x;
9     if (x >= 0)
10         abs_x = x;
11     else
12         abs_x = -x;
13     cout << "|x| = "
14         << abs_x << "\n";
15     return 0;
16 }
```

```
if (expression)
    statement or block
else
    statement or block
```


Conditional operator

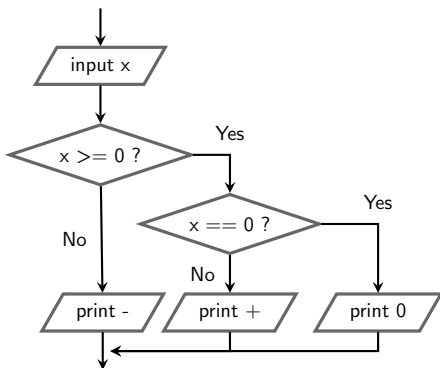
You can use the ternary conditional operator `? :` to create short expressions that work like if/else statements

```
expression ? expression : expression;  
// conditional expression
```

```
int abs_x = x >= 0 ? x : -x;  
// abs_x = (x >= 0) ? (x) : (-x)  
  
cout << "You are "  
      << (age < 18 ? "a child" : "an adult");
```

More than one condition

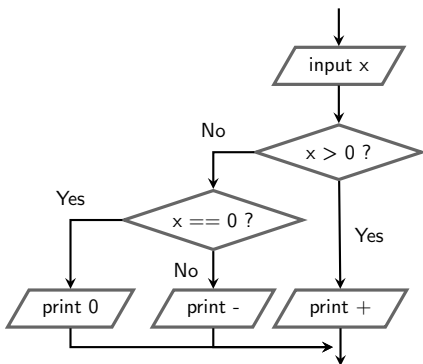
Nested if



```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 int main(){
6     int x;
7     cout << "Enter integer x:";
8     cin >> x;
9     if (x >= 0) {
10         if (x == 0)
11             cout << "x is zero";
12         else
13             cout << "x is
14                 positive";
15     }
16     else {
17         cout << "x is negative";
18     }
19     return 0;
20 }
```

More than one condition

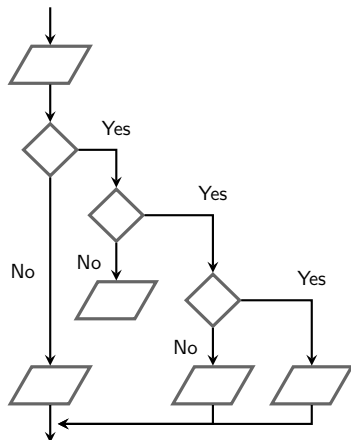
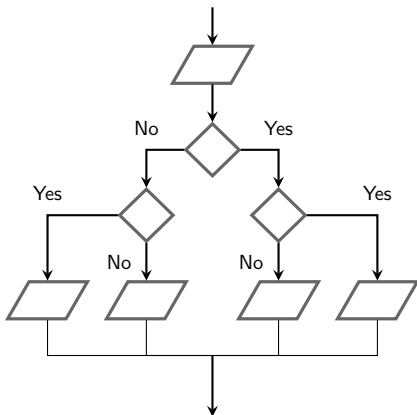
Nested if (if else if)



```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 int main(){
6     int x;
7     cout << "Enter integer x:";
8     cin >> x;
9     if (x > 0) {
10         cout << "x is positive";
11     }
12     else {
13         if (x == 0)
14             cout << "x is zero";
15         else
16             cout << "x is
17                     negative";
18     }
19     return 0;
20 }
```

More than one condition

Nested if



TODO: find the number with a value in middle of a integer triple?

Relational expression

Logical operators

Logical operators connect two or more relational expressions into one or reverse the logic of an expression

NOT		AND			OR		
x	(!x)	x	y	(x && y)	x	y	(x y)
true	false	true	true	true	true	true	true
false	true	true	false	false	true	false	true
		false	true	false	false	true	true
		false	false	false	false	false	false

`!(6 <= 7)` false

`(14 >= 5) && ('A' < 'B')` true

`!age + 2 > 20 || name == 'A' && score > 2` ?

`age > 18 & age < 55` ?

Note: Logical operators rank lower precedence (except !) than the relational operators (reread the precedence table)

Relational expression

Short-circuit evaluation

The computer evaluates the logical expression from left to right.
As soon as the value of the entire logical expression is known, the evaluation stops

<code>(x > y) (x == 5)</code>	<code>if (x > y) == true: then stop evaluation and return true</code>
<code>(x > y) && (x == 5)</code>	<code>if (x > y) == false: then stop evaluation and return false</code>

TODO: [D.S.Malik book] chap 4, programming exercises 9

TODO

- ▶ Finish chapter 4, part 2.4 (VQHoang book)
Read by yourself about `switch case` statement
- ▶ Read chapter 5