

# Limbaajul C++

## 1. Tipuri de date standard. Variabile.

Documentatie : <http://www.cplusplus.com/doc/tutorial/variables/>

Tipurile standard simple (predefinite) din MinGW sunt organizate astfel:

### - tipul caracter

- char

### - tipuri intregi

- int - Domeniu de valori este [-2147483647, 2147483647]
- long long - Domeniu de valori este [  $-2^{62}+1$ ,  $2^{62}-1$ ]

### - tipuri reale

- float
- double
- long double

## 2. Operatori C++

Documentatie : <http://www.cplusplus.com/doc/tutorial/operators/>

## 3. Operatii de citire/ scriere

### a) Citirea variabilelor de la tastatura

```
cin>>variabila1>>variabila2
```

### b) Afisarea expresiilor pe ecran

```
cout<<expresie1<<expresie2
```

## 4. Implementarea structurilor de control in C++

### 1. Structura alternativa

In limbaj pseudocod	C++
<b>Daca</b> conditie	<b>if</b> (expresie)
<b>Atunci</b>	instructiune 1;
Instructiune1	<b>else</b>
<b>Altfel</b>	instructiune 2;
Instructiune2	
Sfarsit-daca	

### 2. Structura repetitiva cu test initial

<p><b>In limbaj pseudocod</b>  <b>a) cu numar nedeterminat de iteratii</b></p> <p><b>Cat timp</b> conditie executa  Instructiune  Sfarsit-cat timp</p> <p><b>b) cu numar determinat de iteratii</b></p> <p><b>pentru</b> variabila=<math>v_{init}</math>, <math>v_{final}</math>, pas <b>executa</b>  instructiune  Sfarsit-pentru</p>	<p>C++</p> <p><b>while (expresie)</b> instructiune</p> <p><b>for</b> (expresie1; expresie2; expresie3)  instructiune;</p>
--	---

**Modul de executie al instructiunii while**

- daca expresia este adevarata (expresie=1) se executa prelucrarile din ciclu;
  - altfel, se trece la executia instructiunii ce urmeaza dupa while in program
- Cu alte cuvinte, prelucrarile din ciclu se executa atat timp cat expresia este adevarata (expresie=1). Daca expresia este falsa de la inceput corpul ciclului nu se va executa deloc.

**Modul de executie al instructiunii for**

- expresiile au urmatoarea semnificatie generala:
  - expresie1 - expresie de initializare;
  - expresie2 - conditia de continuare a executiei ciclului;
  - expresie3 - expresie de actualizare;
- atat timp cat expresie2=1(conditie adevarata) se executa corpul ciclului;
- de fiecare data se evalueaza expresia de actualizare, care are rolul esential de a determina ca, dupa un numar de iteratii, expresie2 sa devina 0 (conditie falsa) ;
- expresie1 se evalueaza o singura data.

**3. Structura repetitiva cu test final**

<p><b>In limbaj pseudocod</b></p> <p><b>Executa</b>  instructiune  <b>Cat timp</b> conditie</p>	<p>C++</p> <p><b>do</b>  instructiune  <b>while (expresie)</b></p>
---	--

**Mod de de executie al instructiunii do while**

- se executa corpul instructiunii;
  - se evalueaza expresia: daca expresia=1 se reia executia, altfel se trece la urmatoarea instructiune din program;
- Deosebirea esentiala fata de instructiunea *while* este aceea ca expresia se evalueaza dupa iteratie.

In cazul in care expresie=0 de la inceput, corpul instructiunii do while se executa o singura data.

## Probleme ce utilizează structuri repetitive

### 1. Multimi

Se consideră  $n$  mulțimi. Fiecare mulțime conține numai numere naturale consecutive. Pentru a indica o astfel de mulțime este suficient să dăm primul și ultimul element din ea.

#### Cerință

Scrieți un program care să determine elementele intersecției celor  $n$  mulțimi.

#### Date de intrare

De la tastatură se citește numărul  $n$ . Apoi, de pe fiecare din cele  $n$  linii următoare, se citesc perechi de numere, câte o pereche pe linie, separate prin câte un spațiu, care reprezintă cel mai mic, respectiv cel mai mare element din fiecare mulțime.

#### Date de ieșire

Pe ecran se vor afișa elementele intersecției cu câte un spațiu între ele. În cazul în care intersecția nu are nici un element se va afișa mesajul "**multimea vida**"

#### Restricții

$0 < n < 31$

Elementele mulțimilor sunt numere naturale  $< 30001$

#### Exemple

Date de intrare	Date de iesire	Explicații
3 5 10 4 11 2 9	5 6 7 8 9	Avem mulțimile {5,6,7,8,9,10} {4,5,6,7,8,9,10,11} {2,3,4,5,6,7,8,9} și intersecția: {5,6,7,8,9}

2. Se considera un șir format din  $n$  numere întregi. O secvență din șir este formată din numere situate pe poziții consecutive. Scrieți un program care obține numărul de numere al celei mai lungi secvențe ordonate crescător din șirul dat.

### Date de intrare

De la tastatură se citește un număr natural  $n$  și  $n$  numere întregi, separate între ele printr-un spațiu.

### Date de ieșire

Pe ecran se va scrie un număr natural nenul ce reprezintă numărul de numere al celei mai lungi secvențe ordonate crescător din șirul dat.

### Restricții

- ( $n \leq 10000$ )
- fiecare număr natural este mai mic decât 2000000000

### Exemple

Date de intrare	Date de ieșire	EXPLICATII
9 123 345 567 502 2390 2567 6789 345600 230	5	Se obține valoarea 5 deoarece cea mai lungă secvență de numere ordonată crescător din șirul dat are 5 numere : 502, 2390, 2567, 6789,345600

### 3. Case

Sorin a luat hotărârea de a scurta numerele caselor, deoarece i se pareau prea lungi. Astfel, a propus ca fiecare cifră a unui număr să se înlocuiască cu complementul cifrei respective față de cea mai mare cifră a numărului. Aplicând aceeași operație pe numărul obținut a observat că anumite numere se reduc la o cifră.

### Cerință

Afișați cifra la care se reduce un număr  $n$  ( $0 < n \leq 999999999$ ) sau cifra 0 dacă numărul nu se poate reduce la o cifră, aplicând regulile date.

### Date de intrare

De la tastatură se citește un număr natural  $n$ .

### Date de ieșire

Pe ecran se afișează cifra la care se reduce numărul  $n$  sau 0 dacă numărul nu se poate reduce la o cifră.

### Exemple

Date de intrare	Date de ieșire	Explicatii
1234	1	1234 -> 3210 -> 0123 -> 210 -> 012 -> 10 -> 1
29340	0	29340 -> 70659 -> 29340 (ciclu infinit)
195	4	195 -> 804 -> 084 -> 4

4. Se citesc  $n$  numere naturale . Scrieti un program care determina numarul de zerouri in care se termina produsul numerelor citite (fara a calcula acest produs).

**Date de intrare**

De la tastatura se citeste un numar natural  $n$  si cele  $n$  numere naturale date, separate intre ele printr-un spatiu.

**Date de iesire**

Pe ecran se afla un numar natural , egal cu numarul de zerouri in care se termina produsul numerelor citite.

**Restrictii**

- ( $n \leq 1000000$ ) .
- fiecare numar natural este mai mic decat 30000

**Exemplu**

Date de intrare	Date de iesire	Explicatie
7 105 200 45 15000 10010 17 1024	9	Produsul celor 7 numere naturale se termina cu 9 zerouri.

5.Cabina

<http://campion.edu.ro/arhiva/index.php?page=problem&action=view&id=835>

6. Case1

<http://campion.edu.ro/arhiva/index.php?page=problem&action=view&id=1084>

7. Cifra1

<http://campion.edu.ro/arhiva/index.php?page=problem&action=view&id=1077>

8. Cifre2

<http://campion.edu.ro/arhiva/index.php?page=problem&action=view&id=825>

**9. Schi**

În tabără la munte s-a organizat un concurs de schi fond. Toți concurenții au parcurs aceeași distanță și au luat startul în același moment. La start s-au aliniat  $n$  concurenți iar pentru fiecare concurent se cunoaște timpul în care a parcurs traseul, exprimat în minute și secunde.

Se cunoaște de asemenea ora, minutul și secunda la care s-a dat startul.

**Cerință**

Scrieți un program care să determine ora, minutul și secunda în care ajunge la linia de sosire primul concurent și ora, minutul și secunda în care ajunge la linia de sosire ultimul concurent.

**Date de intrare**

Fișierul de intrare **schii.in** conține:

- pe prima linie **3** numere naturale, separate prin câte un spațiu, reprezentând ora, minutul și secunda la care s-a dat startul
- pe a doua linie numărul natural  $n$  reprezentând numărul concurenților
- Următoarele  $n$  linii vor conține câte două numere naturale  $m$  și  $s$ , separate prin câte un spațiu, reprezentând timpul realizat de fiecare concurent, exprimat în minute și secunde

## Date de ieşire

În fişierul **sch\_i.out** se vor scrie două linii:

- pe prima linie se vor scrie trei numere naturale, separate prin câte un spaţiu, reprezentând ora, minutul şi secunda sosirii concurentului care a ajuns primul la linia de sosire
- pe a doua linie se vor scrie trei numere naturale, separate prin câte un spaţiu, reprezentând ora, minutul şi secunda sosirii concurentului care a ajuns ultimul la linia de sosire

## Restricţii şi precizări

- $1 < n < 100$
- Startul se dă între orele 8:00 si 20:00
- $0 \leq m < 60$  ,  $0 \leq s < 60$

## Exemplu

Date de intrare	Date de ieşire	Explicaţie
10 50 3 5 10 20 12 15 8 58 15 3 10 12	10 59 1 11 5 6	Momentul startului: ora <b>10, 50</b> de minute şi <b>3</b> secunde.  Au participat <b>5</b> schiori.  Concurentul care ajunge primul la linia de sosire este schiorul care a parcurs traseul în <b>8</b> minute şi <b>58</b> de secunde şi a trecut linia de sosire la momentul: ora <b>10, 59</b> de minute şi <b>1</b> secundă.  Concurentul care ajunge ultimul la linia de sosire este schiorul care a parcurs traseul în <b>15</b> minute şi <b>3</b> secunde şi a trecut linia de sosire la momentul: ora <b>11, 5</b> minute şi <b>6</b> secunde.

**Timp maxim de execuţie/test: 1 secundă.**

## 10. Numere

Se generează un şir de numere naturale ai cărui primi termeni sunt, în această ordine:

**1, 2, 3, 5, 8, 3, 1, 4, 5, 9, 4, 3, 7, 0, 7, 7, 4, ...**

## Cerinţe

Deduceţi regula după care sunt generaţi termenii şirului şi scrieţi un program care să citească numerele naturale **n**, **k** şi **p** şi care să determine:

- suma tuturor numerelor prime aflate printre primii **n** termeni ai şirului din enunţ;
- numărul de apariţii ale cifrei **k** printre primii **n** termeni ai şirului din enunţ;
- cel de-al **p**-lea termen al şirului din enunţ.

### Date de intrare

Fișierul `nr.in` conține o singură linie pe care sunt scrise trei numere naturale  $n$ ,  $k$  și  $p$ , separate prin câte un spațiu.

### Date de ieșire

Fișierul de ieșire `nr.out` va conține 3 linii:

- pe prima linie, un număr natural reprezentând suma tuturor numerelor prime aflate printre primii  $n$  termeni ai șirului din enunț
- pe a doua linie, numărul de apariții ale cifrei  $k$  printre primii  $n$  termeni ai șirului din enunț
- pe a treia linie se va afișa cel de-al  $p$ -lea termen al șirului din enunț

### Restricții și precizări:

- Numerele  $n$ ,  $k$  și  $p$  sunt naturale
- $3 \leq n \leq 100$
- $0 \leq k \leq 9$
- $1 \leq p \leq 2000000000$

### Exemplu:

<code>nr.in</code>	<code>nr.out</code>	<i>Explicații</i>
19 5 26	47 3 8	Primii 19 termeni ai șirului sunt: 1, <u>2</u> , <u>3</u> , <u>5</u> , 8, <u>3</u> , 1, 4, <u>5</u> , 9, 4, <u>3</u> , <u>7</u> , 0, <u>7</u> , <u>7</u> , 4, 1, <u>5</u> . Numere prime care apar printre acești termeni sunt numerele subliniate. Calculăm suma acestora: $2+3+5+3+5+3+7+7+7+5=47$ . Valoarea 47 se scrie pe prima linie a fișierului <code>nr.out</code> . Pe linia a doua a fișierului <code>nr.out</code> se scrie numărul 3, deoarece sunt 3 termeni egali cu $k$ ( $k=5$ ) printre primii $n=19$ termeni ai șirului. Pe ultima linie a fișierului se scrie numărul 8, deoarece cel de-al $p$ -lea ( $p=26$ ) termen al șirului este 8.