

Exercicios con CodeBlocks

Programación I

1º curso (1C)

Exercicios con CodeBlocks

Mentres en teoría non avancemos un pouco máis, serán bastante sinxelos os programas que poidamos escribir en C++. Por iso, o obxectivo desta Práctica, máis que aprender moito C++, será unicamente entender o funcionamento de CodeBlocks como contorna de traballo: como crear un proxecto, escribir o código do programa, compilalo, corrixir os posibles erros e, por último, executalo.

I. Exemplos resoltos

Exemplo 1

Obter a media e a desviación típica dun conxunto de 5 números.

Lembremos que xa temos o pseudocódigo deste algoritmo escrito e probado con PSeInt.

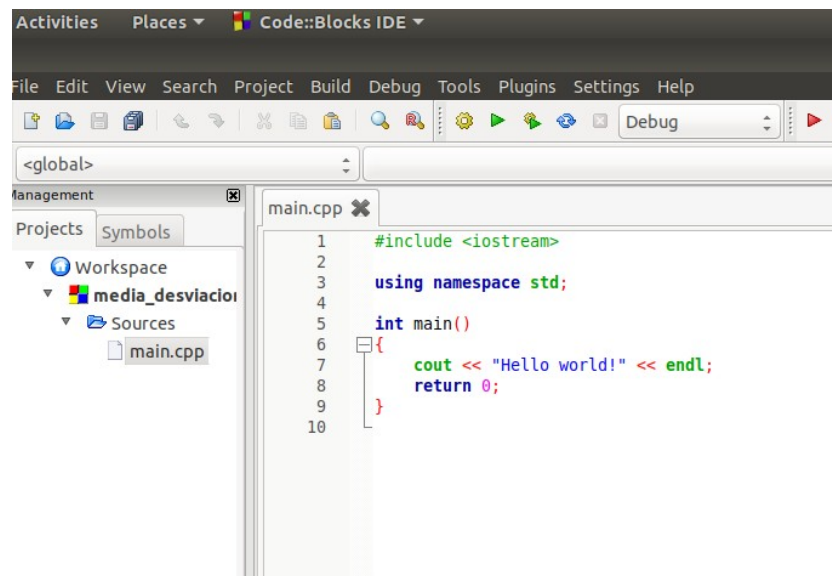
```
estadistica_5numeros.psc* [X]
1  Algoritmo estatistica 5numeros
2  // Algoritmo de cálculo da media e desviación dunha lista de 5 números
3  // v1,v2,v3,v4,v5 : Real
4  // med, desv : Real
5  Escribir "Algoritmo de calculo da media e desviacion dunha lista de 5 números"
6  Escribir "Teclee os 5 números introducindo un retorno de carro despois de cada un"
7  Leer v1,v2,v3,v4,v5
8  med <- (v1+v2+v3+v4+v5)/5
9  desv <- (((v1-med)^2+(v2-med)^2+(v3-med)^2+(v4-med)^2+(v5-med)^2)/4)^0.5
10 Escribir "A media dos 5 números é ", med
11 Escribir "A desviación é ", desv
12 FinAlgoritmo
```

O primeiro que teremos que facer é crear un proxecto en CodeBlocks. Como xa coñecemos os pasos, facémolo. Chamáremoslle “media_desviacion” (lembra indicar a carpeta na que queres gardar o proxecto). Se despois de facer todos os pasos non ves o código base na pantalla, fíxate na fiestra da esquerda (*management*) e verás unha carpeta chamada “*source*”. Preme no “▶”, e verás que se desprega unha lista cos arquivos desa carpeta. Debería haber un arquivo denominado “*main.cpp*”. Fai dobre clic nel e listo!

Garda o proxecto e despois comproba, empregando o xestor de arquivos, os arquivos que se xeraron.

Antes de poñernos a escribir o código do noso algoritmo, deteñámonos no código do programa por defecto. De momento non comentaremos o significado das dúas primeiras liñas (farémolo máis adiante), pero verás que estarán presentes en todos os programas que escribas en C++.

Todos os programas terán tamén unha “función principal”, denominada *main*, cunha sintaxe como a amosada na figura seguinte.



Este programa consta dunha unha única liña, ademais do “esqueleto”. Trátase da liña:

cout << “Hello world!” << endl;

Esta liña equivale ao pseudocódigo seguinte:

ESCRIBIR “Hello world!”

A continuación, debes substituír esa liña pola código do programa que queremos implementar, neste caso o que calcula a media e a desviación típica dun conxunto de 5 números. Seguindo paso a paso o pseudocódigo probado en PseInt, o código en C++ sería o seguinte:

```

#include <iostream>
#include <cmath>
using namespace std;

int main()
{
    /* Algoritmo de cálculo da media e desviación dunha lista de 5
    números */
    float v1,v2,v3,v4,v5;
    float med,desv;
    cout << "Algoritmo de calculo da media e desviación dunha lista
    de 5 números" << endl;
    cout << "Teclee os 5 numeros introducindo un retorno de carro
    despois de cada un" << endl;
    cin >> v1 >> v2 >> v3 >> v4 >> v5;
    med = (v1+v2+v3+v4+v5)/5;
    desv = sqrt(((v1-med)*(v1-med)+(v2-med)*(v2-med)+(v3-med)*(v3-
    med)+(v4-med)*(v4-med)+(v5-med)*(v5-med))/4);
    cout << "A media dos 5 números é " << med << endl;
    cout << "A desviacion e " << desv << endl;
    return 0;
}

```

Unha vez tecleado o código, hai que compilalo. Para iso xa coñeces o botón correspondente. Poden pasar dúas cousas, que non haxa erros ou que os haxa. Se non os hai, na fiestra inferior, debería aparecer algo como o seguinte:

```

----- Build: Debug in media_desviacion (compiler: GNU GCC Compiler)-----
g++ -Wall -fexceptions -g -c /home/anton/ESEI/docencia/PROI/curso2020_21/material/practicas/clase3/exercicios/media_desviacion/main.cpp -o obj/Debug/main.o
g++ -o bin/Debug/media_desviacion obj/Debug/main.o
Output file is bin/Debug/media_desviacion with size 32.78 KB
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))

```

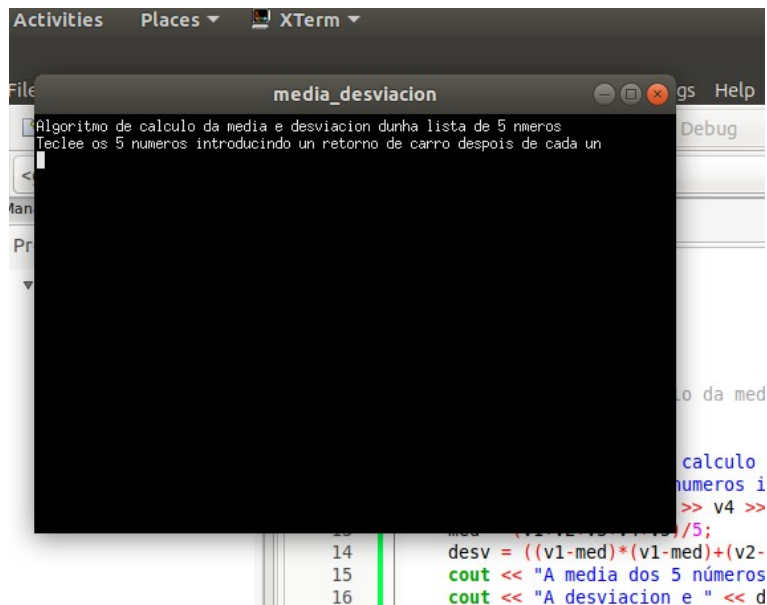
As dúas primeiras liñas indican o comando concreto empregado para a compilación e enlazado. A terceira indica o nome e tamaño do arquivo executable final. A máis importante, a cuarta, indica que o proceso finalizou sen erros.

Se cometemos algún erro aparecerá algo como o seguinte:

File	Line	Message
=== Build: Debug in media_desviacion (compiler: GNU GCC Compiler) ===		
/home/anton/ESEI/...		In function 'int main()':
/home/anton/ESEI/...	12	error: 'v55' was not declared in this scope
/home/anton/ESEI/...	12	note: suggested alternative: 'v5'

Esta mensaxe infórmanos de que na liña 12 hai un erro, pois estamos a empregar a variable “v55”, que non foi declarada previamente. É moi importante ler e entender as mensaxes de erro para corrixir os correspondentes fallos.

Unha vez compilado o programa sin erros procedemos a executalo, premendo no botón correspondente. Facémolo e abríranos unha fiestra como a seguinte:



Non temos máis que seguir as instrucións e teclear os 5 valores pedidos para ver impreso na pantalla o resultado. Finalmente, seguindo as instrucións escritas na pantalla, prememos de novo o retorno de carro para que se peche esa fiestra.

Neste momento podemos pechar o proxecto, ou saír de Codeblocks.

Exemplo 2

Escribe un algoritmo para demostrar que a propiedade asociativa se cumpre, ou sexa: $A*(B+C)=A*B+A*C$

Para este exercicio, cómpre que crear un novo proxecto. Lembra todos os pasos precisos.

A continuación, colle o pseudocódigo creado na clase anterior e vai “traducindo” a código C++. Fixándote no exemplo anterior, non deberías de ter problema.

Unha vez compilado sen erros, proba a executalo.

Para practicar algúns erros típicos e ver que mensaxes xera na fiestra de *logs*, proba as seguintes modificacións, unha por unha. Para cada unha trata de compilar e fíxate na fiestra de *logs*.

- Borrar un “;” no final dalgunha liña.
- Escribir mal o nome dunha variable.
- Esquecerse de pechar unhas comiñas.
- Escribir “end” no canto de “endl” nalgunha liña.

II. Exercicios propostos

Algúns dos exercicios desta Práctica son repetidos respecto da de pseudocódigo, pero algúns son novos. Aínda que máis adiante xa codificarás os algoritmos directamente en C++, nestas primeiras prácticas trata de facelos primeiro en pseudocódigo, para comprobar que alomenos a estrutura está ben, antes de pasalos a C++, onde seguramente cometerás erros de sintaxe, polo menos ao principio.

Exercicio 1

Hai que calcular a superficie e o perímetro dun terreo que ten a forma amosada na figura. Indica que medidas é preciso tomar para iso, e escribe un algoritmo para calcular ambos os dous valores.

Documenta o programa (significado das variables, obxecto do mesmo, ...).



Exercicio 2

Escribe un algoritmo para calcular a altura dun depósito de líquidos de forma cilíndrica, coñecido o diámetro da base e a capacidade do mesmo. Saberías calcular tamén canto aceiro fai falta para fabricalo se a chapa é de 3 mm de espesor? (o peso do aceiro é de 7850 Kg/m³).

Exercicio 3

O salario bruto dun traballador consta dos seguintes compoñentes:

- Salario base
- Complementos
- Dietas
- Plus de produtividade

Para calcular o salario neto hai que restarlle o seguinte:

- 5% de salario base máis complementos para cotización por desemprego.
- 8% de retencións IRPF do salario bruto excepto dietas.

Escribe un programa para calcular dito salario neto, informando dos distintos conceptos (salario bruto, cotización para desemprego, retención IRPF e salario neto).

Exercicio 4

A empresa ALFA emprega un sinxelo programa de cifrado para transmitir datos numéricos entre os seus empregados. Cada empregado ten unha clave propia, un número enteiro de 3 díxitos que só é coñecido polos traballadores da empresa. Cada vez que ten que transmitir un dato, fai a seguinte operación:

$$M = N * c_orixe - c_destino$$

sendo c_orixe a súa propia clave, $c_destino$ a do destinatario, e N o número que quere transmitir.

Cando o destinatario recibe a mensaxe, para descifrala realiza o proceso inverso:

$$N = (M + c_destino) / c_orixe$$

Escribe un programa de cifrado e outro de descifrado, e comproba o seu funcionamento.

Nota: para este programa deberás empregar variables enteiras (tipo int).

Exercicio 5

Facer un programa para calcular o tempo que tarda un raio láser (ou calquera onda electromagnética) entre un punto na terra e un satélite de comunicacións, coñecida a súa distancia á superficie terrestre. Podes probar cos satélites de comunicacións xeoestacionarios, que están a 35.786 km. Tamén podes probar coa lúa, que está a 384.400 km. Ou tamén co sol, que está aproximadamente a 150 millóns de km. Se non a sabes, busca na Internet a velocidade das ondas electromagnéticas e da luz.

Exercicio 6

Facer un programa para calcular o Índice de Masa Corporal (IMC) dunha persoa. A fórmula para calculalo é: $IMC = \text{Peso (Kg)} / \text{Altura (m)}^2$. Pídelle ao usuario que introduza o seu peso (en kg) e a súa altura (en cm), e finalmente infórmalle do seu IMC.

Exercicio 7

Fai un programa para calcular a frecuencia cardíaca máxima que unha persoa non debería de exceder cando fai exercicio físico. Unha fórmula que se emprega habitualmente para calculala é:

$$FCM = 208,75 - [0,73 * idade]$$