

Variables e instrucciones

Programación I

1º curso (1C)

Variables e instrucións

Nesta Práctica imos facer pequenos programas nos cales poidamos ir probando os distintos tipos de variables, operadores, e instrucións de Entrada/Saída.

Exercicio 1

Tipos de datos

Crea un novo proxecto e escribe o seguinte programa:

Antes de executalo, pensa cal sería a saída. Que pasa coa variable a?

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

int main()
{
    int a;
    const float b = 3.5;
    char letra = 'x';
    char frase [20] = "habelas hainas";
    //b=2.5;
    cout << "Valor da variable enteira a = " << a << endl;
    cout << "Valor da variable real b = " << b << endl;
    cout << "valor da variable character letra = " << letra << endl;
    cout << "Valor da cadea de caracteres frase = " << frase << endl;
    return 0;
}
```

Antes das liñas de saída, introduce a seguinte liña:

b=2.5;

Proba a compilar, que pasa?

Exercicio 2

Operadores

Crea un novo proxecto e escribe o programa que se amosa a continuación.

Antes de executalo, tenta adiviñar cal sería a saída. Acertaches? Se fallaches algunha, entendes a razón do fallo?

```

#include <iostream>

using namespace std;

int main() {
    int a,b;
    cout << "Valor da variable a? ";
    cin >> a;
    cout << "Valor da variable b? ";
    cin >> b;

    cout << "a%b= " << a%b << endl;
    cout << "a>b " << (a>b) << endl;
    cout << "++a " << ++a << endl;
    cout << "b++" << b++ << endl;
    cout << "(a>b) && (a!=0) = " << ((a>b) && (a!=0)) << endl;
    return 0;
}

```

Exercicio 3

Operadores e constantes

Crea un novo proxecto cun programa que calcule o perímetro e a área dunha circunferencia a partir do seu radio, que debe ser un dato que se pida ao usuario por teclado. Define unha constante para o valor de PI, e variables de tipo real para recoller o valor do radio, calcular o perímetro e a área. Funciona?. Que ocorre se cambias o tipo das variables para o perímetro e a área por *int*?

Exercicio 4

Operadores de asignación adicionais

Crea un novo proxecto e proba o seguinte código. Lembra analizar antes de executalo cal sería a saída.

```

#include <iostream>

using namespace std;

int main()
{
    int i = 8, j = 5;
    float x = 0.005, y = -0.01;

    cout << "i = " << i << ", j = " << j << ", x = " << x << ", y = " << y << endl;

    i /= j;
    cout << "i = i/j: " << i << endl;
    x *= 2;
    cout << "x = x*2: " << x << endl;
    y -= i;
    cout << "y = y-i: " << y << endl;
    i -= (i-j);
    cout << "i = i - (i-j): " << i << endl;
    x /= i;
    cout << "x = x/i: " << x << endl;
    i %= j;
}

```

Exercicio 5

Expresións aritméticas

Realiza un programa que calcule a coordenada y dun punto dunha función lineal, dadas a pendente, a intersección co eixo de abscisas, e a coordenada x do punto. Pódese empregar a seguinte ecuación:

$$y = \text{pendente} * x + \text{intersección}$$

Exercicio 6

Expresións aritméticas

Realiza un programa que calcule a distancia en quilómetros percorrida por un coche e a súa velocidade media. O usuario deberá de introducir o punto quilométrico inicial e final, xunto co tempo (hora, minuto e segundo) empregado.

Exercicio 7

Expresións aritméticas

Fai un programa que solicite ao usuario 3 valores enteiros a, b y c, e calcule o resultado X da seguinte expresión, que debe aparecer en pantalla:

$$x = \frac{a+b}{2c} + 2(a/bc)$$

Como debes indicar a expresión en C++ para que as operacións se realicen na orde adecuada?

Exercicio 8

Expresións aritméticas

Deseña un programa que lea por teclado unha temperatura en grados Celsius e a transforme en Fahrenheit e Kelvin, amosando eses valores por pantalla.

Exercicio 9

Variables temporais

Escribe un programa que solicite os valores de dúas variables A e B por teclado, logo intercambie os valores destas variables e, por último, visualice os novos valores delas.

Exercicio 10

Enumerados

Crea un novo proxecto e analiza o funcionamento de seguinte programa:

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

// Tipos de contrato de traballo
enum TipoContrato {INDEFINIDO, TEMPORAL, DISCONTINUO, FORMACION};
const char *nomeContrato [] = {"Indefinido", "Temporal", "Discontinuo", "Formación"};
const int NUM_CONTRATOS = 4;

int main()
{
    char dni[10];
    TipoContrato contrato;
    char entrada[2];
    int c;

    // Entrada de datos
    cout << "Introducir dni: ";
    cin.getline(dni, 10);
    do {
        cout << "\nIntroducir tipo de contrato:";
        for (int i = 0; i < NUM_CONTRATOS; i++) {
            cout << "\n" << i+1 << " " << nomeContrato[i];
        }
        cout << endl;
        cin.getline(entrada, 2);
        c = atoi(entrada)-1;
    } while (c<0 || c>3);

    switch (c) {
        case 0: contrato = INDEFINIDO;
                break;
        case 1: contrato = TEMPORAL;
                break;
        case 2: contrato = DISCONTINUO;
                break;
        case 3: contrato = FORMACION;
                break;
    }

    // Saída de datos
    cout << "DNI: " << dni << endl;
    cout << "Tipo de contrato: " << nomeContrato[contrato];

    return 0;
}
```

Modifica o código anterior para engadir un novo tipo de contrato: PRACTICAS.