

## Practica 2

Esta práctica consta de 9 ejercicios. Aquí escribiré los códigos fuente (en azul) y comentare los resultados obtenidos al ejecutarlos (en negro). Los archivos \*.cpp y los archivos \*.exe están en el mini cd adjunto.

Ejercicio 1(e\_1.cpp/e\_1.exe);

```
/*(Practica 2)e_1.cpp.
*Programa realizado por Marc Belda.
*/
#include <iostream>
using namespace std;
int main()
{
    bool unidad = true;
    while (unidad == true)
    {
        char c;
        cout <<"Este programa imprime en pantalla el valor numerico asociado a un\n";
        cout <<"caracter arbitrario.Introduce una letra cualquiera(presiona la letra\n";
        cout <<"(f) o (F) para finalizar la ejecucion).\n";
        cin >> c;
        if (c == 'f' || c == 'F')
        {
            unidad = false;
        }
        cout <<"El valor numerico asociado a "<< c <<" es "<<int(c)<<endl;
    }
    return 0;
}
```

Comprobaremos el funcionamiento del programa imprimiendo en pantalla el valor numérico de algunos caracteres;

*a = 97*  
*b = 98*  
*c = 99*  
*x = 120*  
*y = 121*  
*z = 122*

Es como si nos números estuviesen ordenados “alfabéticamente”, de la misma forma que la letra *b* sigue a la letra *a* en el abecedario latino, el numero 98 sigue al 97.

También llama la atención que el valor numérico de la letra *A* sea distinto al de la letra *a*.

Unix distingue entre mayúsculas y minúsculas.

Ejercicio 2(e\_2.cpp/e\_2.exe);

```
/*(Practica 2)e_2.cpp.
*Programa realizado por Marc Belda.
```

```

*/
#include <iostream>
using namespace std;
int main()
{
    int i;
    cout <<"Este programa imprime el valor entero de las letras minusculas del\n";
    cout <<"del alfabeto."<<endl;
    for ( i = 'a';i<='z';i++)
    {
        cout <<int (i)<<"\n";
    }
    return 0;
}

```

El programa funciona correctamente.

Ejercicio 3(e\_3.cpp/e\_3.exe);

```

/*(Practica 2)e_3.cpp.
*Programa realizado por Marc Belda.
*/
#include <iostream>
using namespace std;
int main()
{
    long double a;
    long int b;
    short int c;
    cout <<"El tamaño de long double es "<<sizeof(a)<<" bits.\n";
    cout <<"El tamaño de long int es "<<sizeof(b)<<" bits.\n";
    cout <<"El tamaño de short int es "<<sizeof(c)<<" bits.\n";
    return 0;
}

```

El tamaño del tipo *long double* en mi ordenador es de *12 bits*, el de *long int* de *4 bits* y el de *short int* de *2 bits*.

Ejercicio 4(e\_4.cpp/e\_4.exe);

```

/*(Practica 2)e_4.cpp.
*Programa realizado por Marc Belda.
*/
#include <iostream>
using namespace std;
int main()
{
    double darray[5]={ 11,22,33,44,55};
    cout <<"Primero imprimiremos el contenido del arreglo usando indices.\n";
}

```

```

cout <<"El valor de darray[0] es " <<darray[0]<<endl;
cout <<"El valor de darray[1] es " <<darray[1]<<endl;
cout <<"El valor de darray[2] es " <<darray[2]<<endl;
cout <<"El valor de darray[3] es " <<darray[3]<<endl;
cout <<"El valor de darray[4] es " <<darray[4]<<endl;
cout <<"Ahora imprimiremos el contenido del arreglo usando punteros.\n";
double *p = &darray[0];
cout <<"El valor de darray[0] es " <<*p<<endl;
cout <<"El valor de darray[1] es " <<*(p+1)<<endl;
cout <<"El valor de darray[2] es " <<*(p+2)<<endl;
cout <<"El valor de darray[3] es " <<*(p+3)<<endl;
cout <<"El valor de darray[4] es " <<*(p+4)<<endl;

return 0;
}

```

El programa funciona correctamente.

Ejercicio 5(e\_5.cpp/e\_5.exe);

```

/*(Practica 2)e_5.cpp.
*Programa realizado por Marc Belda.
*/
#include <iostream>
using namespace std;
int main()
{
double darray[10]={ 1,2,3,4,5,6,7,8,9,10};
cout <<"Primero imprimiremos el contenido del arreglo usando indices.\n";
for (int i=0;i<10;i++)
{
cout <<"El valor de darray["<<i<<"]es,darray["<<i<<"]= "<< darray[i]<< endl;
}
cout <<"Ahora invertimos el contenido del arreglo."<< endl;
float temp;
temp=0;
int i=0;
int j=9;
while (i<j)
{
temp=darray[i];
darray[i]=darray[j];
darray[j]=temp;
cout <<"El valor de darray["<<i<<"]es,darray["<<i<<"]= "<< darray[i]<< endl;
i++;
j--;
}
int l=0;
int m=9;

```

```

while (l<m)
{
temp=darray[l];
darray[l]=darray[m];
darray[m]=temp;
cout <<"El valor de darray["<<m<<"]es,darray["<<m<<"]= "<< darray[l]<< endl;
l++;
m--;
}
return 0;
}

```

El programa funciona correctamente.

Ejercicio 6(e\_6.cpp/e\_6.exe);

```

/*(Practica 2)e_6.cpp.
*Programa realizado por Marc Belda.
*/
#include <iostream>
using namespace std;
int main()
{
double darray[10]={ 1,2,3,4,5,6,7,8,9,10};
double * p=&darray[0];
cout <<"Primero imprimiremos el contenido del arreglo usando indices.\n";
for (int i=0;i<10;i++)
{
cout <<"El valor de darray["<<i<<"]es,darray["<<i<<"]= "<<*(p+i)<< endl;
}
cout <<"Ahora invertiremos el arreglo utilizando punteros.\n";
float temp;
temp=0;
int i=0;
int j=9;
while (i<j)
{
double *p=&darray[0];
temp=*(p+i);
*(p+i)=*(p+j);
*(p+j)=temp;
cout <<"El valor de darray["<<i<<"]es,darray["<<i<<"]= "<<*(p+i)<< endl;
i++;
j--;
}
int l=0;
int m=9;
while (l<m)
{

```

```

double* p=&darray[0];
temp=*(p+l);
*(p+l)=*(p+m);
*(p+m)=temp;
cout <<"El valor de darray["<<m<<"]es,darray["<<m<<"]= "<<*(p+l)<< endl;
l++;
m--;
}
return 0;
}

```

El programa funciona correctamente.

Ejercicio 7(e\_7.cpp/e\_7.exe);

```

/*(Practica 2)e_7.cpp.
*Programa realizado por Marc Belda.
*/
#include <iostream>
using namespace std;
//Declaro la funcion
int intercambia (int&i,int&j)
{
i=7;
j=2;
}
int main()
{
int i=2;
int j=7;
cout <<"Este programa intercambia mediante el uso de una funcion el valor de\n";
cout <<"dos variables enteras,i y j.\n";
cout <<"El valor de i es,i= "<<i<<endl;
cout <<"El valor de j es,j= "<<j<<endl;
//LLamo a la funcion
intercambia(i,j);
cout <<"Despues de intercambiar las dos variables enteras tenemos que;\n";
cout <<"i= "<<i<<endl;
cout <<"j= "<<j<<endl;
return 0;
}

```

El programa funciona correctamente.

Ejercicio 8(e\_8.cpp/e\_8.exe);

```

/*(Practica 2)e_8.cpp.
*Programa realizado por Marc Belda.
*/
#include <iostream>

```

```

using namespace std;
//Declaro la funcion
int intercambia (int* i,int* j)
{
*i=7;
*j=2;
}
int main()
{
int i=2;
int j=7;
cout <<"Este programa intercambia mediante el uso de una funcion el valor de\n";
cout <<"dos variables enteras,i y j.\n";
cout <<"El valor de i es,i= "<<i<<endl;
cout <<"El valor de j es,j= "<<j<<endl;
//LLamo a la funcion
intercambia(&i,&j);
cout <<"Despues de intercambiar las dos variables enteras tenemos que;\n";
cout <<"i= "<<i<<endl;
cout <<"j= "<<j<<endl;
return 0;
}

```

El programa funciona correctamente. Creo que la utilización de punteros es más intuitiva.

Ejercicio 9(e\_9.cpp/e\_9.exe);

```

/*(Practica 2)e_9.cpp.
*Programa realizado por Marc Belda.
*/
#include <iostream>
using namespace std;
double productoEscalar (double v1[], double v2[], unsigned int n);
int main()
{
cout <<"Este programa calcula el producto escalar de dos vectores.\n";
cout <<"Introduzca la dimension de los vectores,\n n=";
unsigned int n;
cin >> n;
cout <<"Creamos el vector v1."<<endl;
double* v1= new double[n];
for (unsigned int i=0;i<n;i++)
{
cout <<"Introduzca el valor de v1["<<i<<"]="";
cin >> v1[i];
}
cout <<"Ahora, creamos v2."<< endl;
double* v2=new double[n];
for (unsigned int i=0;i < n;i++)

```

```

{
cout <<"Introduzca el valor de v2["<<i<<"]=";
cin >>v2[i];
}
double v1v2=productoEscalar ( v1, v2, n);
cout <<"El producto escalar vale "<< v1v2 <<endl;
delete [] v1;
delete [] v2;
return 0;
}
double productoEscalar (double v1[], double v2[], unsigned int n)
{
double v1v2=0;
for (unsigned int i=0;i<n;i++)
{
v1v2 = v1v2+((v1[i]) * (v2[i]));
}
return (v1v2);
}

```

Comprobaremos el funcionamiento del programa calculando el producto escalar de dos vectores de, por ejemplo, dimensión  $n = 4$ .

$$v1 = (1,3,5,7)$$

$$v2 = (2,1,2,3)$$

El producto escalar de estos dos vectores es 36.El programa funciona correctamente.