Extensiones para el lenguaje de programación C++
__ ec++ __

Propuestas realizadas por Octulio Biletán.
Las palabras reservadas nuevas son: entry, exit, access, forward, rewind, loop, until y repeat.

[*]Ciclos

Ciclo do...loop
(1) Usar:

```
do
{
        cuerpo;
}loop;
```

(2) Usar:

```
loop
{
        cuerpo;
}
```

```
#define loop        while(1)
```

Ciclo do...until
(1) Usar:

```
do
{
        cuerpo;
} until(cond);
```

(2) Usar:

```
until(cond)
{
        cuerpo;
}
```

```
#define until(cond) while(!(cond))
```

Ciclo repeat(n)
Usar:

```
repeat(n)
{
        cuerpo;
}
```

```
#define repeat(n)  for(unsigned int a = 0; a < n; a++)
```

Ciclo access
Recorre secuencialmente un arreglo unidimensional ya sea desde su primera posición o desde su última posición
hasta la última posición o hasta la primera posición.
Sintaxis:

```
        access(<arreglo>; forward | rewind) { cuerpo; }
```

Ciclo access y forward
Recorre secuencialmente el arreglo unidimensional desde la posición 0 hasta su última posición.
Por ejemplo:

```
        int vector[] = { 4, 33, 23, 1, 0, 7, 99, -6, -62, 19 };

        access(vector; forward)
        {
            // Muestra el contenido del vector, posición vigente
            cout << *vector << endl;

            // Si en la posición vigente del vector contiene un 7
            // entonces se sale del ciclo 'access'
            if(*vector == 7)
                break;
        }
```

Ciclo access y rewind
Recorre secuencialmente el arreglo unidimensional desde la última posición hasta la posición 0.
Por ejemplo:

```
        int vector[] = { 4, 33, 23, 1, 0, 7, 99, -6, -62, 19 };

        access(vector; rewind)
        {
            // Muestra el contenido del vector, posición vigente
            cout << *vector << endl;

            // Si en la posición vigente del vector contiene un 7
            // entonces se sale del ciclo 'access'
            if(*vector == 7)
                break;
        }
```

Para todas las instrucciones de ciclos se puede utilizar la palabra reservada

'break' para salir del ciclo. Por ejemplo:

```
        int vector[] = { 4, 33, 23, 1, 0, 7, 99, -6, -62, 19 };
        int a = (sizeof(vector) / sizeof(int)) - 1;
        until(a == -1)
        {
          cout << vector[a] << endl;
          if(vector[a] == 7)
            break;
          a--;
        }
```

También es posible utilizar la palabra reservada 'continue' para continuar ciclando.
Por ejemplo:

```
        int vector[] = { 4, 0, 23, 7, 0, 0, 0, -6, -62, 19 };

        access(vector; rewind)
        {
            // Si en la posición vigente del vector contiene un 0
            // entonces continúa ciclando.
            if(*vector == 0)
              continue;

            // Muestra el contenido del vector, posición vigente.
            cout << *vector << endl;

            // Si en la posición vigente del vector contiene un 7
            // entonces se sale del ciclo 'access'.
            if(*vector == 7)
                break;
        }
```

---

[*] Entrada al programa principal: entry.

```
        Usos:
        (1)     entry int main(int argc, char **argv, char **env)
                {
                        cuerpo;
                }
        no se requieren cambios.

        (2)     entry void principal(void)
                {
                        cuerpo;
                }
        se cambia por:
```

```
                void main(void)
                {
                        cuerpo;
                }

        (3)     entry void main_program(void)
                {
                        cuerpo;
                }
        se cambia por:
                void main(void)
                {
                        cuerpo;
                }
        Para todos los casos se quita la palabra 'entry' y se reemplaza el
nombre
        de la función por 'main'.
        En Windows:
                entry WINAPI principal(HINSTANCE a, HINSTANCE b, LPTSTR c,
int d)
                {
                        cuerpo;
                }
        Se cambia por:
                WINAPI WinMain(HINSTANCE a, HINSTANCE b, LPTSTR c, int d)
                {
                        cuerpo;
                }

Cuando ocurre esto en un mismo programa fuente:
        entry void entrada_principal(void)
        {
                cuerpoA;
        }

        void main(void)
        {
                cuerpoB;
        }

el compilador debe hacer:
        void main(void)
        {
                cuerpoA;
        }

        void _main_(void)
        {
                cuerpoB;
```

```
        }
```

En todo programa fuente tiene que haber una única función 'main'.
La palabra reservada 'entry' aparece por primera en "The C programming language",
Apéndice A, página 180, 1º edición de 1978, edición inglesa.
Escrito por Brian W. Kernighan y Dennis M. Ritchie.
Estaba reservada para uso futuro pero nunca se implementó en el compilador C.

[*] Salida del programa principal: exit.
Si hay una entrada (entry) pues también hay una salida (exit).
La palabra reservada 'exit' no va acompañada de paréntesis porque existe la función 'exit' en la biblioteca de funciones estándar de C/C++ <stdlib.h> y <stdlib>

    Uso:

```
        // función de entrada
        entry void main(int argc, char **argv)
        {
                cuerpo;
        }

        // función de salida
        exit int main(int retval)
        {
                cuerpo;
        }
```

    Ejemplo:

```
        // función de entrada
        entry int main(int argc, char **argv)
        {
                hace algo aquí...;

                // Llama a la función de salida con el valor 0
                exit 0;
        }

        // función de salida
        exit int xmain(int retval)
        {
                // Por ejemplo libera espacio de memoria
                hace algo aquí...;

                // Valor de retorno para el S.O.
                return retval;
        }
```

Código de arranque para la nueva implementación (ec++):

```
        int main(int argc, char **argv, char **env)
        {
                int retval_main, retval_exit;

                retval_main = emain(argc, argv, env);
                retval_exit = xmain(retval_main);

                return retval_exit;
        }
```

[*] Vinculación con otros lenguajes de programación mediante la palabra reservada 'extern':

```
        Uso: extern <sequencia de caracteres>

        Conexión a C: extern "c"
        Conexión a C++: extern "c++"
        Conexión a Java: extern "java"
        Conexión a Vala: extern "vala"
```

_____

Ultima actualización: 03:02 p.m. jueves, 14 de enero de 2021