```cpp
#include <iostream>
#include <windows.h>
#include <stdio.h>
#include <time.h>
#include <chrono>

#include "NTL/ZZ.h"

using namespace NTL;

int main()
{
   //Iniciar el timer

   auto start = std::chrono::high_resolution_clock::now();

   SYSTEM_INFO siSysInfo;

   GetSystemInfo(&siSysInfo);

   //Obtener información del hardware de la PC

   printf("Hardware info: \n");

   std::cout << std::endl;

   printf("  OEM ID: %u\n",
      siSysInfo.dwOemId);
   printf("  Number of processors: %u\n",
      siSysInfo.dwNumberOfProcessors);
   printf("  Page size: %u\n",
      siSysInfo.dwPageSize);
   printf("  Processor type: %u\n",
      siSysInfo.dwProcessorType);
   printf("  Active processor mask: %u\n",
      siSysInfo.dwActiveProcessorMask);
   printf("  Processor level: %u\n",
      siSysInfo.wProcessorLevel);
   printf("  Processor Architecture: %u\n",
      siSysInfo.wProcessorArchitecture);
   printf("  Processor Revisison: %u\n",
      siSysInfo.wProcessorRevision);

   //Convertir los datos a NTL

   ZZ a(siSysInfo.dwOemId);
   ZZ b(siSysInfo.dwNumberOfProcessors);
   ZZ c(siSysInfo.dwPageSize);
   ZZ d(siSysInfo.dwProcessorType);
   ZZ e(siSysInfo.dwActiveProcessorMask);
   ZZ f(siSysInfo.wProcessorLevel);
   ZZ g(siSysInfo.wProcessorArchitecture);
   ZZ h(siSysInfo.wProcessorRevision);

   //Detener el timer

      auto finish = std::chrono::high_resolution_clock::now();

   //Mostrar el tiempo obtenido en nanosegundos

      std::cout << "\nTime: " << std::chrono::duration_cast<std::chrono::nanoseconds
>(finish-start).count() << " nanoseconds\n";

   //Convertir el tiempo obtenido a NTL

   ZZ t(std::chrono::duration_cast<std::chrono::nanoseconds>(finish-start).count());
```

```
66
67      //Multiplicar los datos de la PC por el tiempo y añadirlos a un array
68
69      ZZ Array[8] = {a * t, b * t, c * t, d * t, e * t, f * t, g * t, h * t};
70
71      std::cout << std::endl;
72
73      //Mostrar los números aleatorios
74
75      for(int i = 0; i < 8; i++)
76      {
77        std::cout << "Random number " << i+1 << ": " << Array[i] << std::endl;
78      }
79
80      return 0;
81  }
```