```c
#include <stdio.h>

// Global typedef
typedef int MY_INT; // "type" int has been re"def"ined as MY_INT ... Now, "MY_INT" can be treated just like "int"

int main(void)
{
    // Function prototype
    MY_INT Add(MY_INT, MY_INT);

    // Typedefs
    typedef int MY_INT; // "type" int has been re"def"ined as MY_INT ... Now, "MY_INT" can be treated just like "int"
    typedef float PVG_FLOAT; // "type" float has been re"def"ined as PVG_FLOAT ... Now, "MY_FLOAT" can be treated just like "float"
    typedef char CHARACTER; // "type" char has been re"def"ined as CHARACTER ... Now, "CHARACTER" can be treated just like "char"
    typedef double MY_DOUBLE; // "type" double has been re"def"ined as MY_DOUBLE ... Now, "MY_DOUBLE" can be treated just like "double"

    // ****** JUST LIKE IN Win32SDK !!! ******
    typedef unsigned int UINT;
    typedef UINT HANDLE;
    typedef HANDLE HWND;
    typedef HANDLE HINSTANCE;

    // variable declarations
    MY_INT a = 10, i;
    MY_INT iArray[] = { 9, 18, 27, 36, 45, 54, 63, 72, 81, 90 };

    PVG_FLOAT f_pvg = 30.9f;
    const PVG_FLOAT f_pvg_pi = 3.14f;

    CHARACTER ch = '*';
    CHARACTER chArray_01[] = "Hello";
    CHARACTER chArray_02[][10] = { "RTR", "Batch", "2020-2021" };

    MY_DOUBLE d = 8.041997;

    // ****** JUST RANDOM NUMBERS - THEY HAVE NOTHING TO DO WITH ANY WINDOW'S HANDLE OR INSTANCE HANDLE !!! This is just for understanding  ******
    UINT uint = 3456;
    HANDLE handle = 987;
    HWND hwnd = 9876;
    HINSTANCE hInstance = 14466;

    // code
    printf("\n\n");
    printf("Type MY_INT variable a = %d\n", a);

    printf("\n\n");
```

```
47      for (i = 0; i < (sizeof(iArray) / sizeof(int)); i++)
48      {
49          printf("Type MY_INT array variable iArray[%d] = %d\n", i, iArray[i]);
50      }
51      printf("\n\n");
52
53      printf("\n\n");
54      printf("Type PVG_FLOAT variable f = %f\n", f_pvg);
55      printf("Type PVG_FLOAT constanct f_pvg_pi = %f\n", f_pvg_pi);
56
57      printf("\n\n");
58      printf("Type MY_DOUBLE variable d = %lf\n", d);
59
60      printf("\n\n");
61      printf("Type CHARACTER variable ch = %c\n", ch);
62
63      printf("\n\n");
64      printf("Type CHARACTER array variable chArray_01 = %s\n", chArray_01);
65
66      printf("\n\n");
67      for (i = 0; i < (sizeof(chArray_02) / sizeof(chArray_02[0])); i++)
68      {
69          printf("%s\t", chArray_02[i]);
70      }
71      printf("\n\n");
72
73      printf("\n\n");
74      printf("Type UINT variable uint = %u\n\n", uint);
75      printf("Type HANDLE variable handle = %u\n\n", handle);
76      printf("Type HWND variable hwnd = %u\n\n", hwnd);
77      printf("Type HINSTANCE variable hInstance = %u\n\n", hInstance);
78      printf("\n\n");
79
80      MY_INT x = 90;
81      MY_INT y = 30;
82      MY_INT ret;
83
84      ret = Add(x, y);
85      printf("ret = %d\n\n", ret);
86
87      return(0);
88  }
89
90  MY_INT Add(MY_INT a, MY_INT b)
91  {
92      // code
93      MY_INT c;
94      c = a + b;
95      return(c);
96  }
97
```