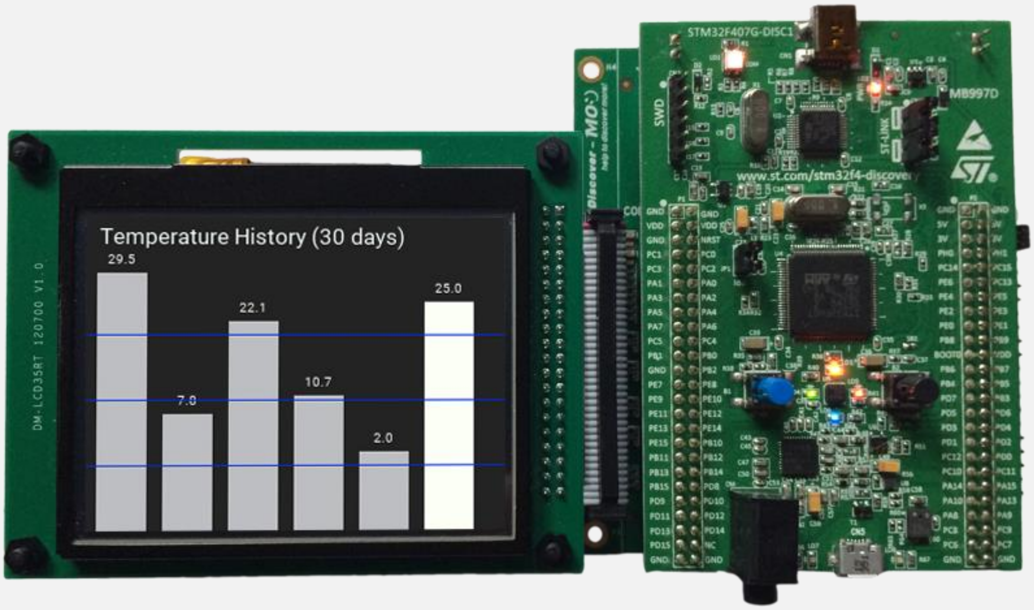
EC-211

DIGITAL SIGNAL PROCESSING LAB

KEIL PROGRAMMING ASSIGNMENT



UTKARSH MAHAJAN 201EC164

ARNAV RAJ 201EC109

<u>Objective:</u> Write a C program to find the convolution of the sequences

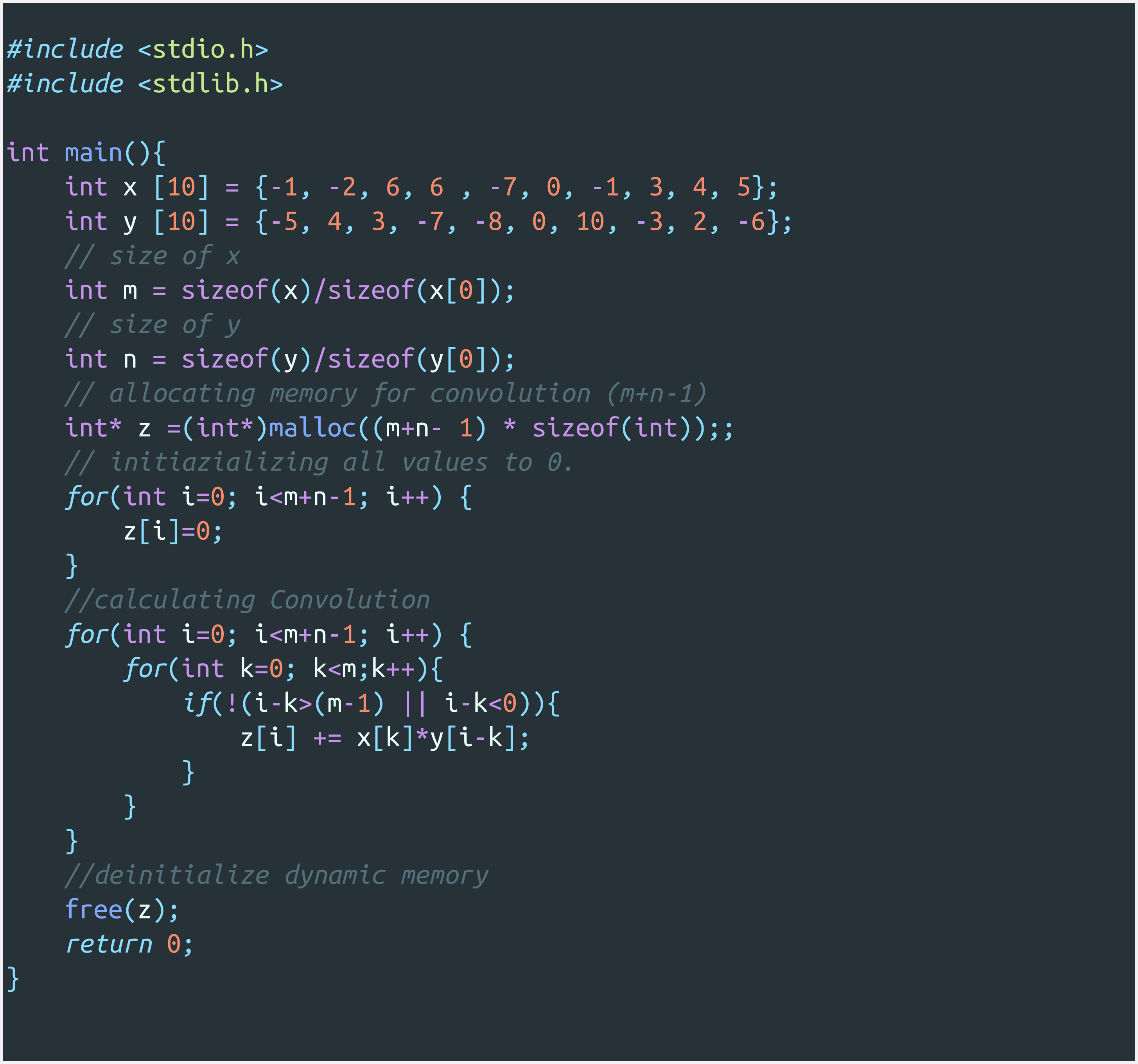
x[n]=[-1, -2, 6, 6 , -7, 0, -1, 3, 4, 5] and

y[n]=[-5, 4, 3, -7, -8, 0, 10, -3, 2, -6].

Show the memory map of the input and output arrays.

-> We will use dynamic allocation for our result array(where our convolution will be stored).

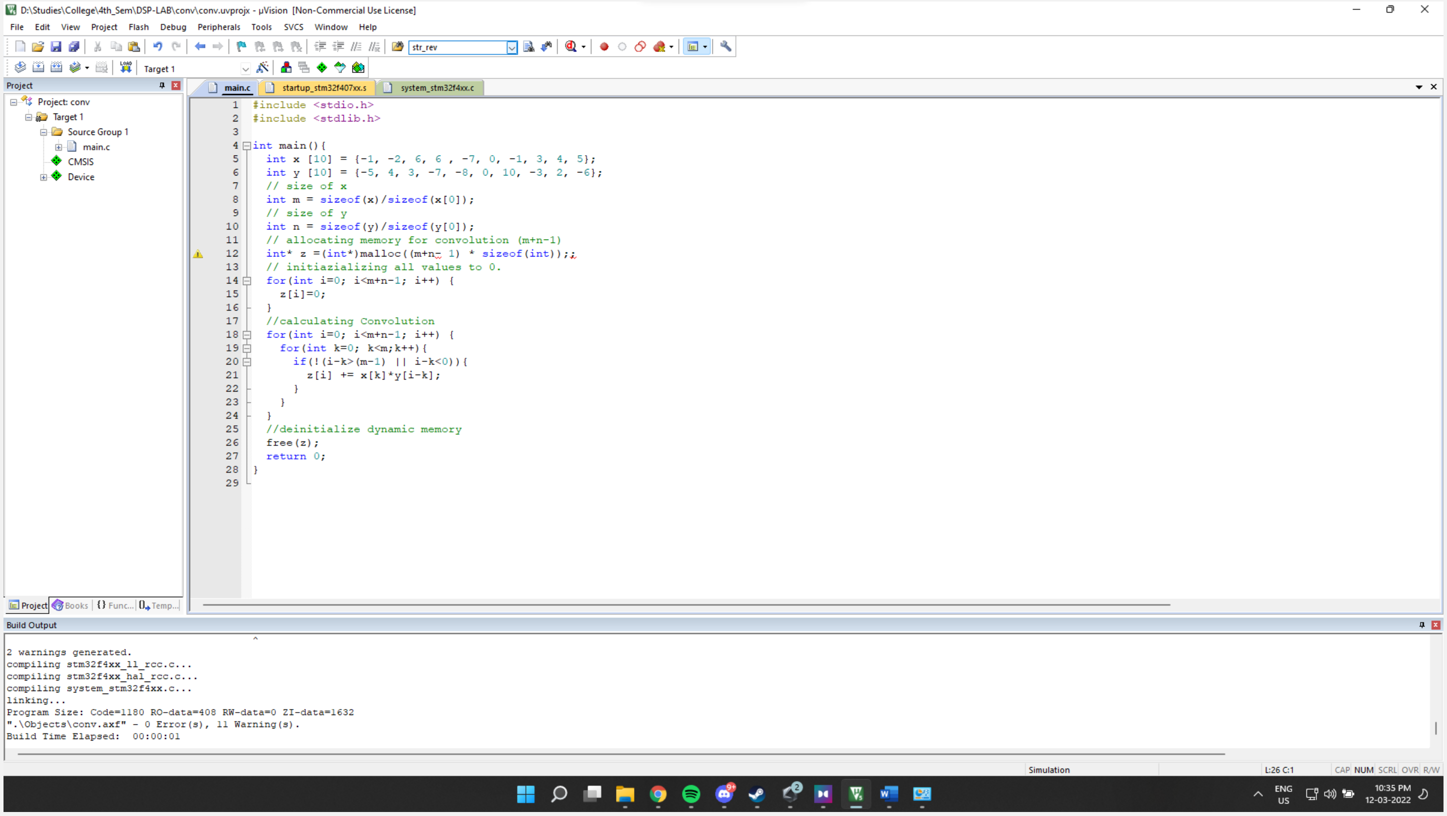
Source Code:



```c
#include <stdio.h>
#include <stdlib.h>

int main(){
    int x [10] = {-1, -2, 6, 6 , -7, 0, -1, 3, 4, 5};
    int y [10] = {-5, 4, 3, -7, -8, 0, 10, -3, 2, -6};
    // size of x
    int m = sizeof(x)/sizeof(x[0]);
    // size of y
    int n = sizeof(y)/sizeof(y[0]);
    // allocating memory for convolution (m+n-1)
    int* z =(int*)malloc((m+n- 1) * sizeof(int));;
    // initiazializing all values to 0.
    for(int i=0; i<m+n-1; i++) {
        z[i]=0;
    }
    //calculating Convolution
    for(int i=0; i<m+n-1; i++) {
        for(int k=0; k<m;k++){
            if(!(i-k>(m-1) || i-k<0)){
                z[i] += x[k]*y[i-k];
            }
        }
    }
    //deinitialize dynamic memory
    free(z);
    return 0;
}
```

Setup in Keil:

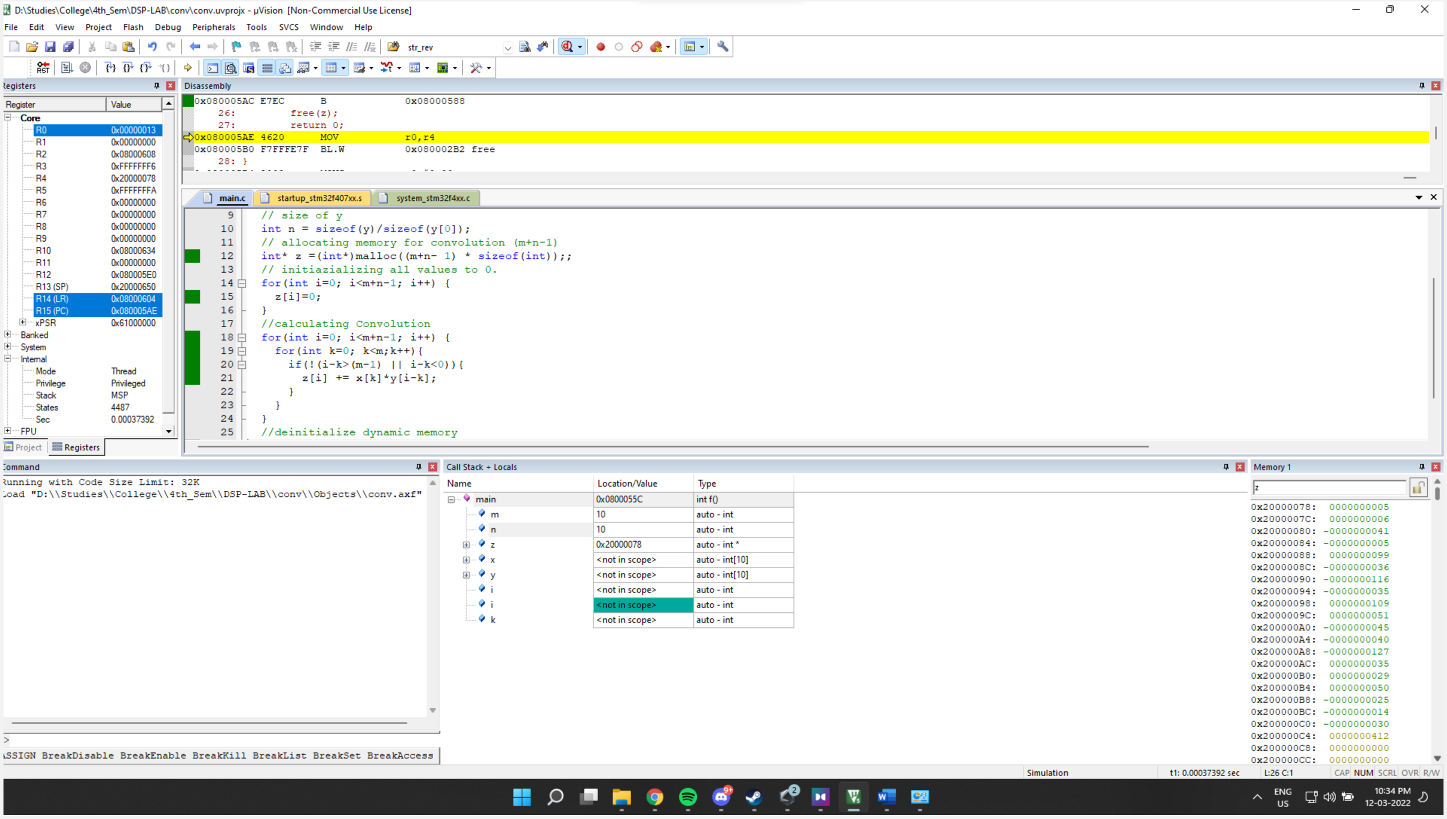


```c
#include <stdio.h>
#include <stdlib.h>

int main(){
  int x [10] = {-1, -2, 6, 6 , -7, 0, -1, 3, 4, 5};
  int y [10] = {-5, 4, 3, -7, -8, 0, 10, -3, 2, -6};
  // size of x
  int m = sizeof(x)/sizeof(x[0]);
  // size of y
  int n = sizeof(y)/sizeof(y[0]);
  // allocating memory for convolution (m+n-1)
  int* z =(int*)malloc((m+n- 1) * sizeof(int));;
  // initiazializing all values to 0.
  for(int i=0; i<m+n-1; i++) {
    z[i]=0;
  }
  //calculating Convolution
  for(int i=0; i<m+n-1; i++) {
    for(int k=0; k<m;k++){
      if(!(i-k>(m-1) || i-k<0)){
        z[i] += x[k]*y[i-k];
      }
    }
  }
  //deinitialize dynamic memory
  free(z);
  return 0;
}
```

```
2 warnings generated.
compiling stm32f4xx_ll_rcc.c...
compiling stm32f4xx_hal_rcc.c...
compiling system_stm32f4xx.c...
linking...
Program Size: Code=1180 RO-data=408 RW-data=0 ZI-data=1632
".\Objects\conv.axf" - 0 Error(s), 11 Warning(s).
Build Time Elapsed:  00:00:01
```

Debugging:



Register	Value
Core	
R0	0x00000013
R1	0x00000000
R2	0x08000608
R3	0xFFFFFFF6
R4	0x20000078
R5	0xFFFFFFFA
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x08000634
R11	0x00000000
R12	0x080005E0
R13 (SP)	0x20000650
R14 (LR)	0x08000604
R15 (PC)	0x080005AE
xPSR	0x61000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	4487
Sec	0.00037392
FPU	

```
0x080005AC E7EC      B           0x08000588
    26:          free(z);
    27:          return 0;
0x080005AE 4620      MOV         r0,r4
0x080005B0 F7FFFE7F  BL.W        0x080002B2 free
    28: }
```

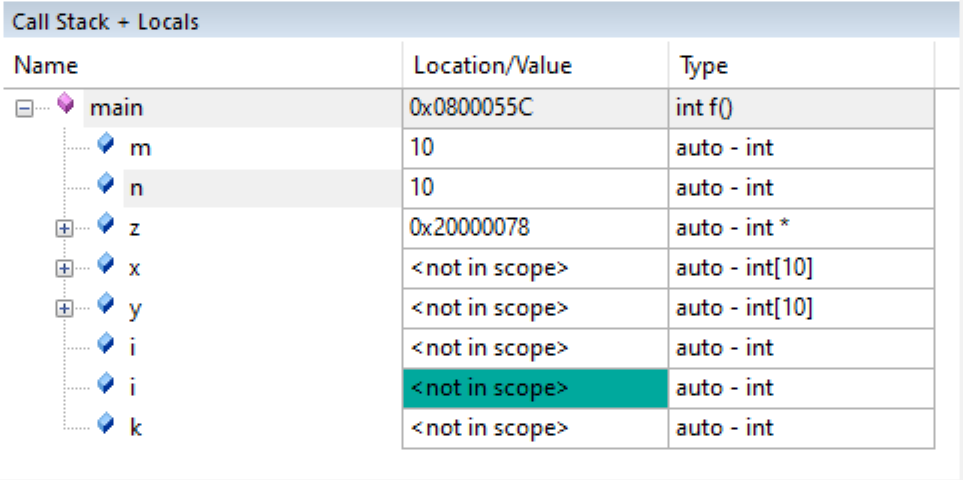
```c
  // size of y
  int n = sizeof(y)/sizeof(y[0]);
  // allocating memory for convolution (m+n-1)
  int* z =(int*)malloc((m+n- 1) * sizeof(int));;
  // initiazializing all values to 0.
  for(int i=0; i<m+n-1; i++) {
    z[i]=0;
  }
  //calculating Convolution
  for(int i=0; i<m+n-1; i++) {
    for(int k=0; k<m;k++){
      if(!(i-k>(m-1) || i-k<0)){
        z[i] += x[k]*y[i-k];
      }
    }
  }
  //deinitialize dynamic memory
```

```
Running with Code Size Limit: 32K
Load "D:\\Studies\\College\\4th_Sem\\DSP-LAB\\conv\\Objects\\conv.axf"
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

Name	Location/Value	Type
main	0x0800055C	int f()
m	10	auto - int
n	10	auto - int
z	0x20000078	auto - int *
x	<not in scope>	auto - int[10]
y	<not in scope>	auto - int[10]
i	<not in scope>	auto - int
i	<not in scope>	auto - int
k	<not in scope>	auto - int

```
z
0x20000078:  0000000005
0x2000007C:  0000000006
0x20000080: -0000000041
0x20000084: -0000000005
0x20000088:  0000000099
0x2000008C: -0000000036
0x20000090: -0000000116
0x20000094: -0000000035
0x20000098:  0000000109
0x2000009C:  0000000051
0x200000A0: -0000000045
0x200000A4: -0000000040
0x200000A8: -0000000127
0x200000AC:  0000000035
0x200000B0:  0000000029
0x200000B4:  0000000050
0x200000B8: -0000000025
0x200000BC: -0000000014
0x200000C0: -0000000030
0x200000C4:  0000000412
0x200000C8:  0000000000
0x200000CC:  0000000000
```

Call stack + Locals

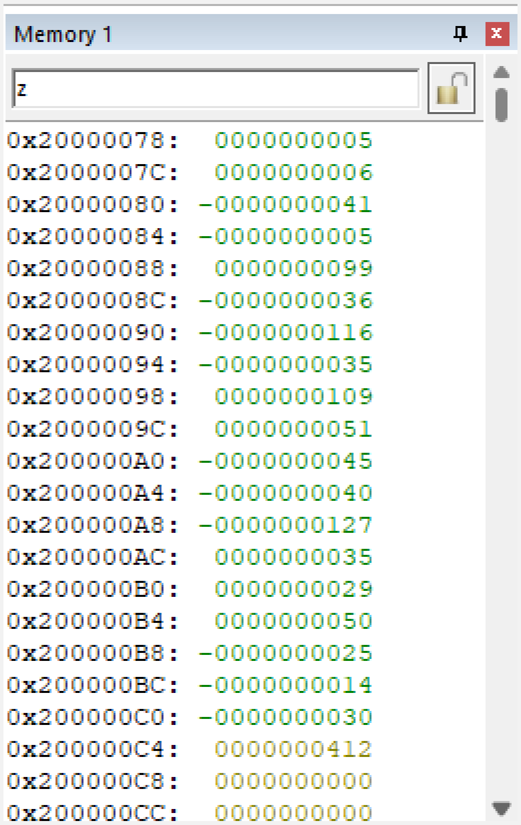


Call Stack + Locals

Name	Location/Value	Type
main	0x0800055C	int f()
m	10	auto - int
n	10	auto - int
z	0x20000078	auto - int *
x	<not in scope>	auto - int[10]
y	<not in scope>	auto - int[10]
i	<not in scope>	auto - int
i	<not in scope>	auto - int
k	<not in scope>	auto - int

Since we have used dynamic allocation for z, we can see only the first element in z. call stack, we will use Memory viewer to see the entire array.

Memory viewer:



Memory 1

z

```
0x20000078:  0000000005
0x2000007C:  0000000006
0x20000080: -0000000041
0x20000084: -0000000005
0x20000088:  0000000099
0x2000008C: -0000000036
0x20000090: -0000000116
0x20000094: -0000000035
0x20000098:  0000000109
0x2000009C:  0000000051
0x200000A0: -0000000045
0x200000A4: -0000000040
0x200000A8: -0000000127
0x200000AC:  0000000035
0x200000B0:  0000000029
0x200000B4:  0000000050
0x200000B8: -0000000025
0x200000BC: -0000000014
0x200000C0: -0000000030
0x200000C4:  0000000412
0x200000C8:  0000000000
0x200000CC:  0000000000
```