*Assembly Language Lecture Series:*

# Introduction to SASM: Constant

Sensei RL Uy, College of Computer Studies,
De La Salle University, Manila, Philippines

# Copyright Notice

# What is SASM?

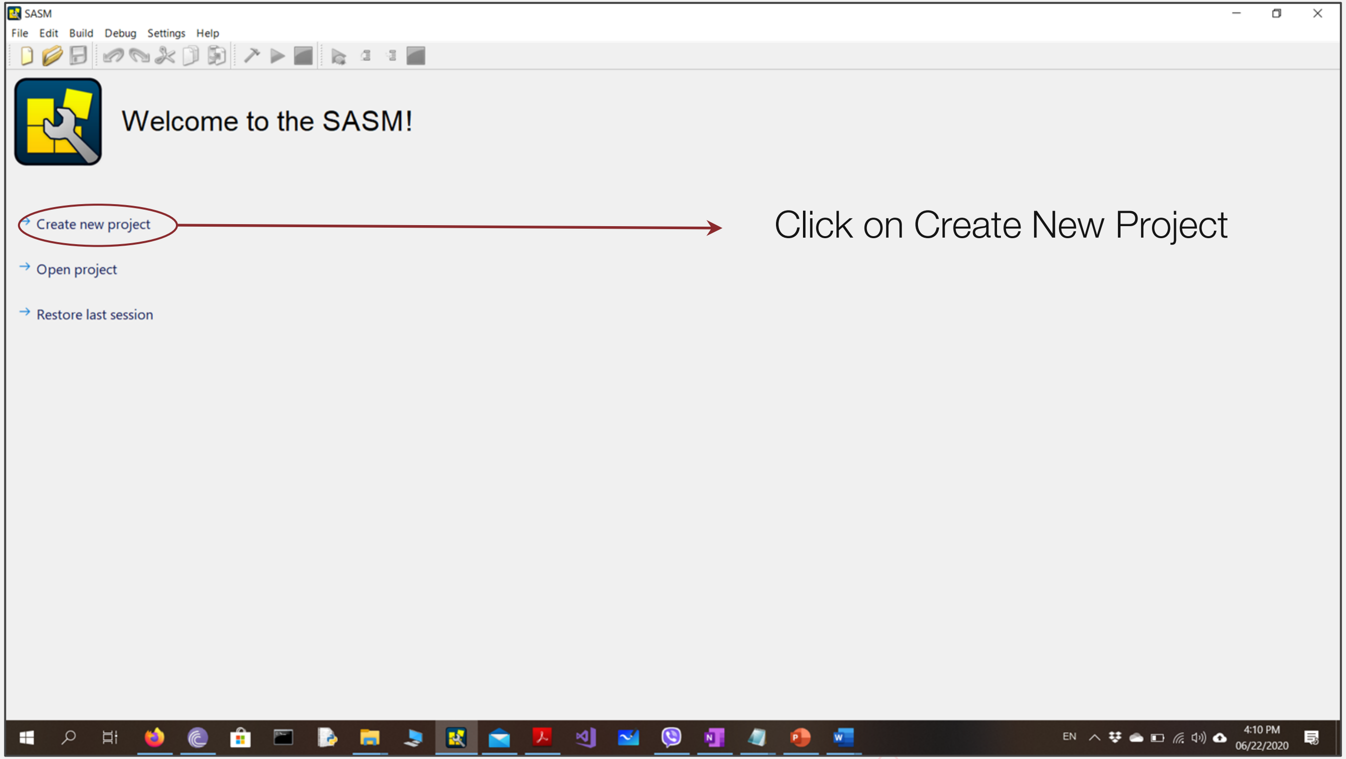
**SASM (SimpleASM)** - simple Open Source cross-platform IDE for NASM assembly languages. SASM has syntax highlighting and debugger.

## How to Download?

**SASM software:** http://dman95.github.io/SASM/english.html
**Installing SASM on Windows:**

- Click on the link “Download for Windows” and follow instructions

# SASM Interface



Welcome to the SASM!

→ Create new project

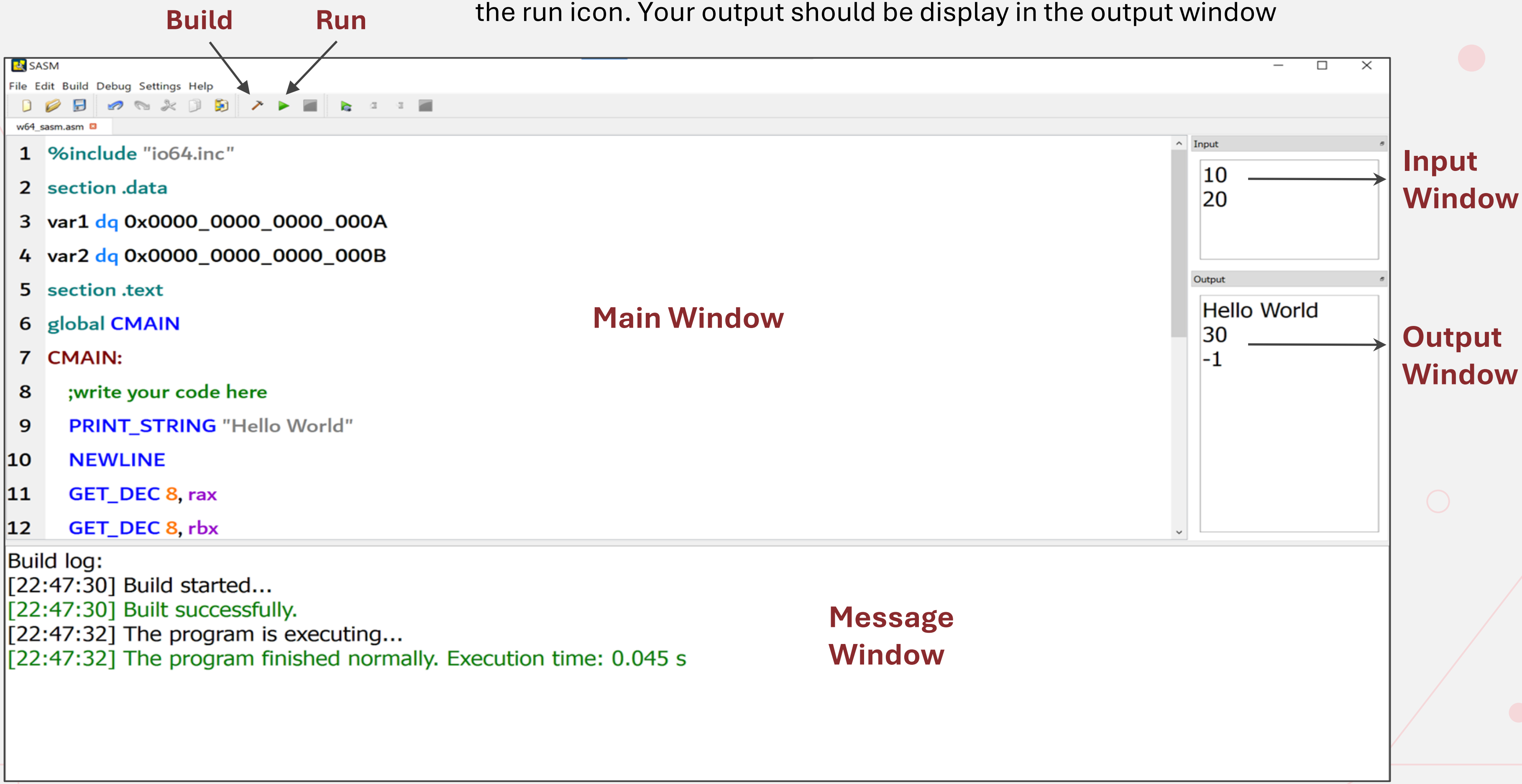
→ Open project

→ Restore last session

Click on Create New Project

**Instruction**:

Type the code below then type in 10 and 20 in the input window. Click the run icon. Your output should be display in the output window



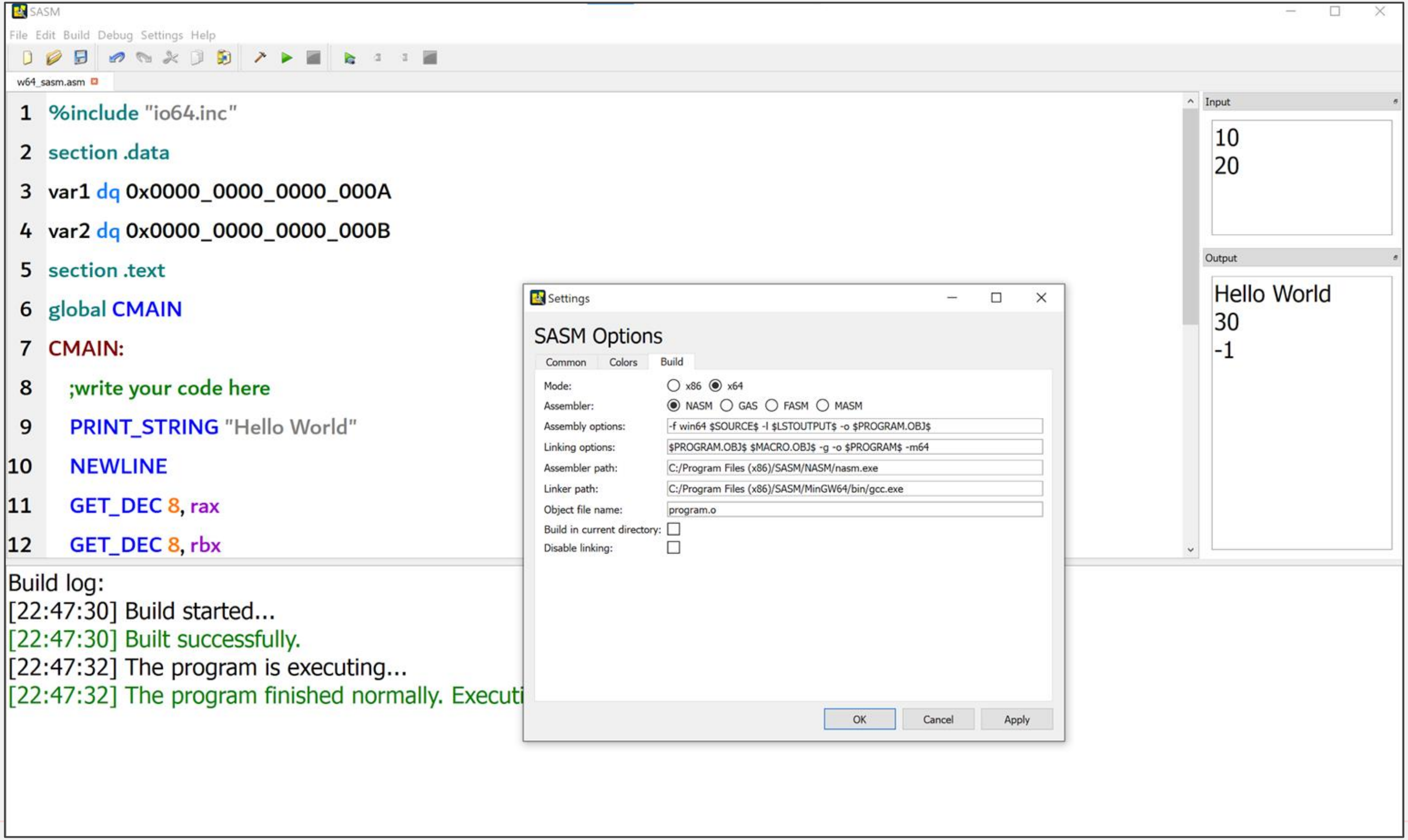
```
%include "io64.inc"
section .data
var1 dq 0x0000_0000_0000_000A
var2 dq 0x0000_0000_0000_000B
section .text
global CMAIN
CMAIN:
    ;write your code here
    PRINT_STRING "Hello World"
    NEWLINE
    GET_DEC 8, rax
    GET_DEC 8, rbx
```

```
10
20
```

```
Hello World
30
-1
```

```
Build log:
[22:47:30] Build started...
[22:47:30] Built successfully.
[22:47:32] The program is executing...
[22:47:32] The program finished normally. Execution time: 0.045 s
```

# SASM Options



```
%include "io64.inc"
section .data
var1 dq 0x0000_0000_0000_000A
var2 dq 0x0000_0000_0000_000B
section .text
global CMAIN
CMAIN:
    ;write your code here
    PRINT_STRING "Hello World"
    NEWLINE
    GET_DEC 8, rax
    GET_DEC 8, rbx
```

SASM Options

Mode: ○ x86 ● x64

Assembler: ● NASM ○ GAS ○ FASM ○ MASM

Assembly options: -f win64 $SOURCE$ -l $LSTOUTPUT$ -o $PROGRAM.OBJ$

Linking options: $PROGRAM.OBJ$ $MACRO.OBJ$ -g -o $PROGRAM$ -m64

Assembler path: C:/Program Files (x86)/SASM/NASM/nasm.exe

Linker path: C:/Program Files (x86)/SASM/MinGW64/bin/gcc.exe

Object file name: program.o

Build in current directory: □

Disable linking: □

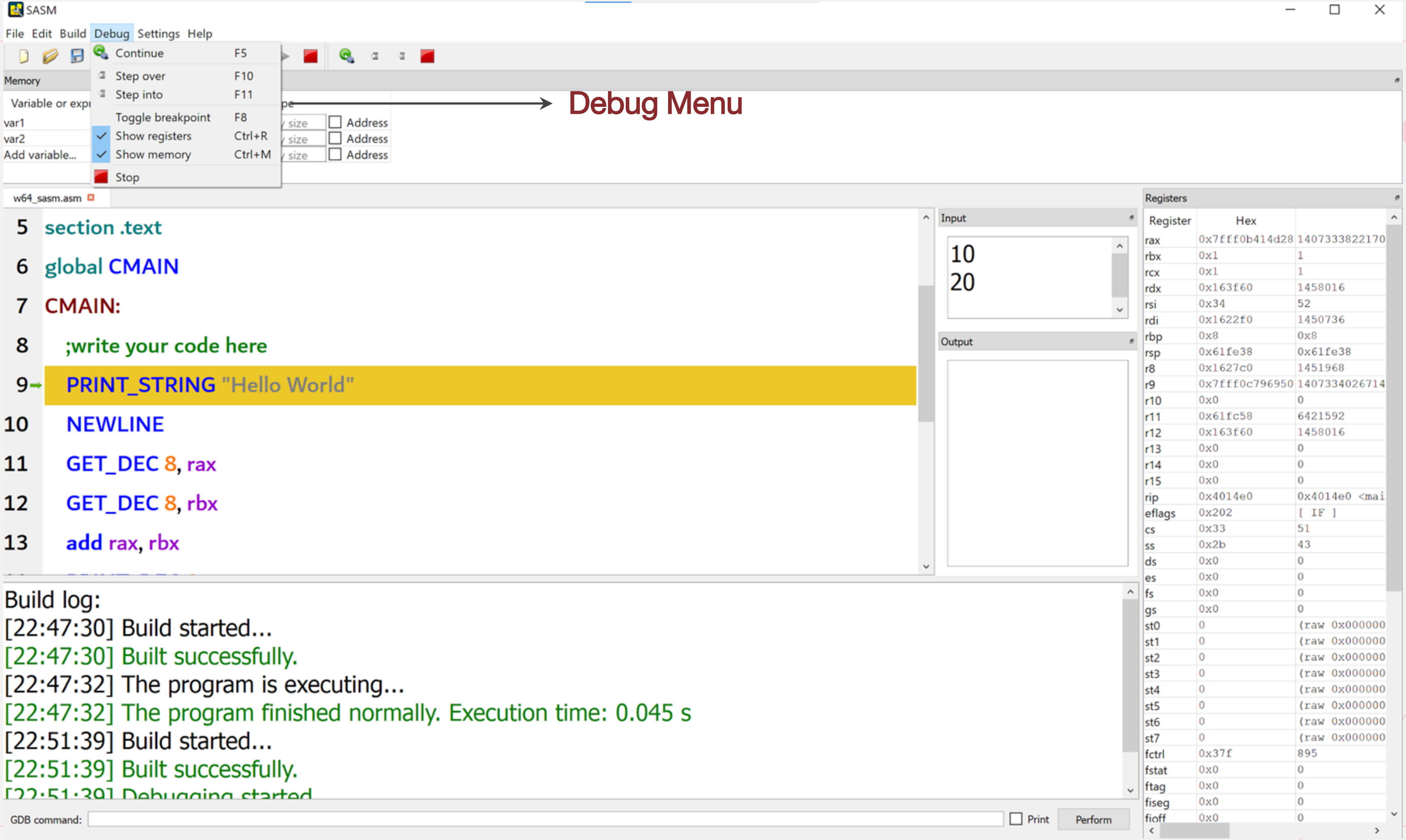
Input

```
10
20
```

Output

```
Hello World
30
-1
```

```
Build log:
[22:47:30] Build started...
[22:47:30] Built successfully.
[22:47:32] The program is executing...
[22:47:32] The program finished normally. Executi
```



Debug Menu

```
section .text
global CMAIN
CMAIN:
    ;write your code here
    PRINT_STRING "Hello World"
    NEWLINE
    GET_DEC 8, rax
    GET_DEC 8, rbx
    add rax, rbx
```

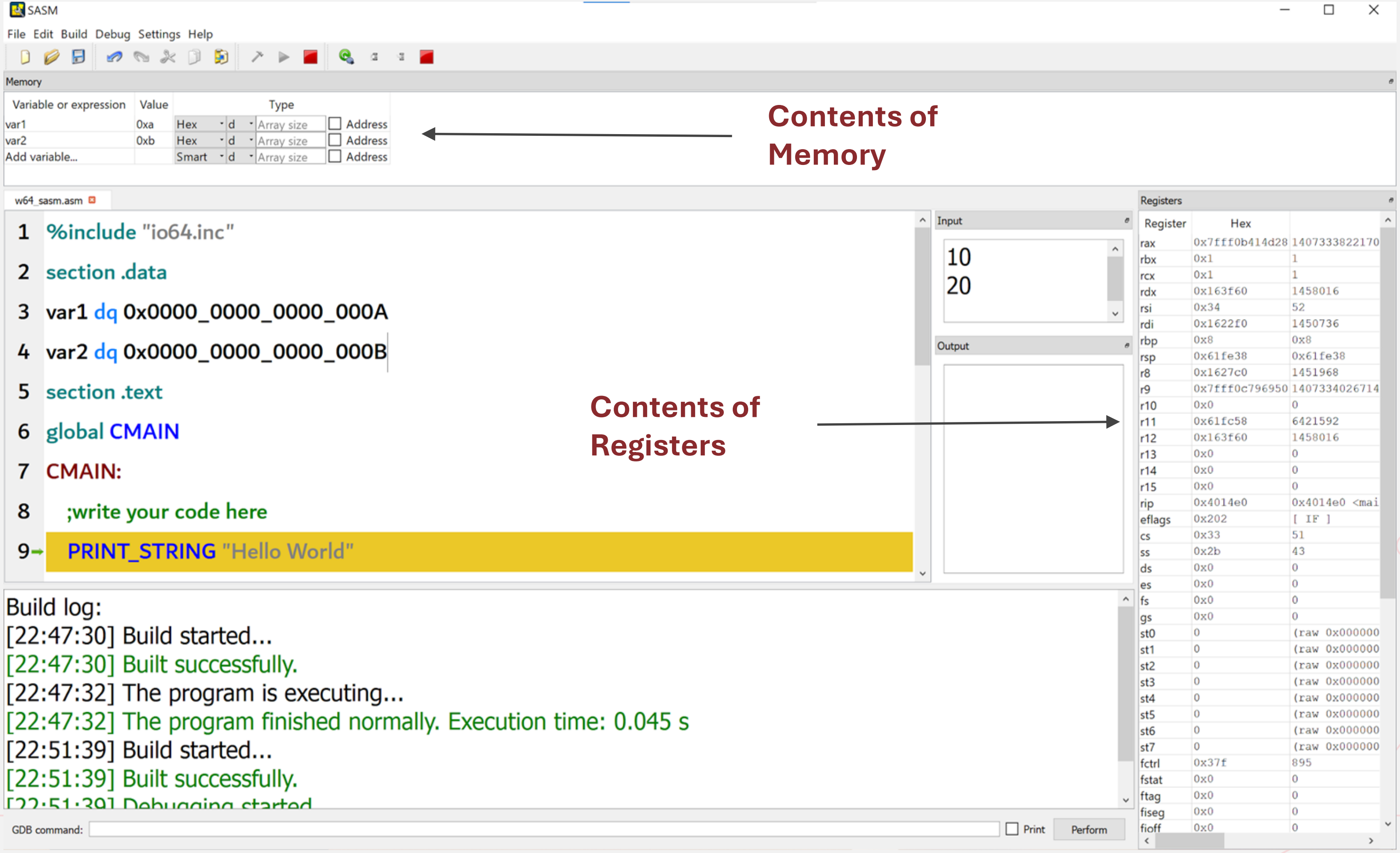
Input:

```
10
20
```

```
Build log:
[22:47:30] Build started...
[22:47:30] Built successfully.
[22:47:32] The program is executing...
[22:47:32] The program finished normally. Execution time: 0.045 s
[22:51:39] Build started...
[22:51:39] Built successfully.
[22:51:39] Debugging started
```

| Register | Hex | |
|---|---|---|
| rax | 0x7fff0b414d28 | 1407333822170 |
| rbx | 0x1 | 1 |
| rcx | 0x1 | 1 |
| rdx | 0x163f60 | 1458016 |
| rsi | 0x34 | 52 |
| rdi | 0x1622f0 | 1450736 |
| rbp | 0x8 | 0x8 |
| rsp | 0x61fe38 | 0x61fe38 |
| r8 | 0x1627c0 | 1451968 |
| r9 | 0x7fff0c796950 | 1407334026714 |
| r10 | 0x0 | 0 |
| r11 | 0x61fc58 | 6421592 |
| r12 | 0x163f60 | 1458016 |
| r13 | 0x0 | 0 |
| r14 | 0x0 | 0 |
| r15 | 0x0 | 0 |
| rip | 0x4014e0 | 0x4014e0 <mai |
| eflags | 0x202 | [ IF ] |
| cs | 0x33 | 51 |
| ss | 0x2b | 43 |
| ds | 0x0 | 0 |
| es | 0x0 | 0 |
| fs | 0x0 | 0 |
| gs | 0x0 | 0 |
| st0 | 0 | (raw 0x000000 |
| st1 | 0 | (raw 0x000000 |
| st2 | 0 | (raw 0x000000 |
| st3 | 0 | (raw 0x000000 |
| st4 | 0 | (raw 0x000000 |
| st5 | 0 | (raw 0x000000 |
| st6 | 0 | (raw 0x000000 |
| st7 | 0 | (raw 0x000000 |
| fctrl | 0x37f | 895 |
| fstat | 0x0 | 0 |
| ftag | 0x0 | 0 |
| fiseg | 0x0 | 0 |
| fioff | 0x0 | 0 |

# Debug



| Variable or expression | Value | Type | | | |
|---|---|---|---|---|---|
| var1 | 0xa | Hex | d | Array size | □ Address |
| var2 | 0xb | Hex | d | Array size | □ Address |
| Add variable... | | Smart | d | Array size | □ Address |

Contents of Memory

```
%include "io64.inc"
section .data
var1 dq 0x0000_0000_0000_000A
var2 dq 0x0000_0000_0000_000B
section .text
global CMAIN
CMAIN:
    ;write your code here
    PRINT_STRING "Hello World"
```

Input:

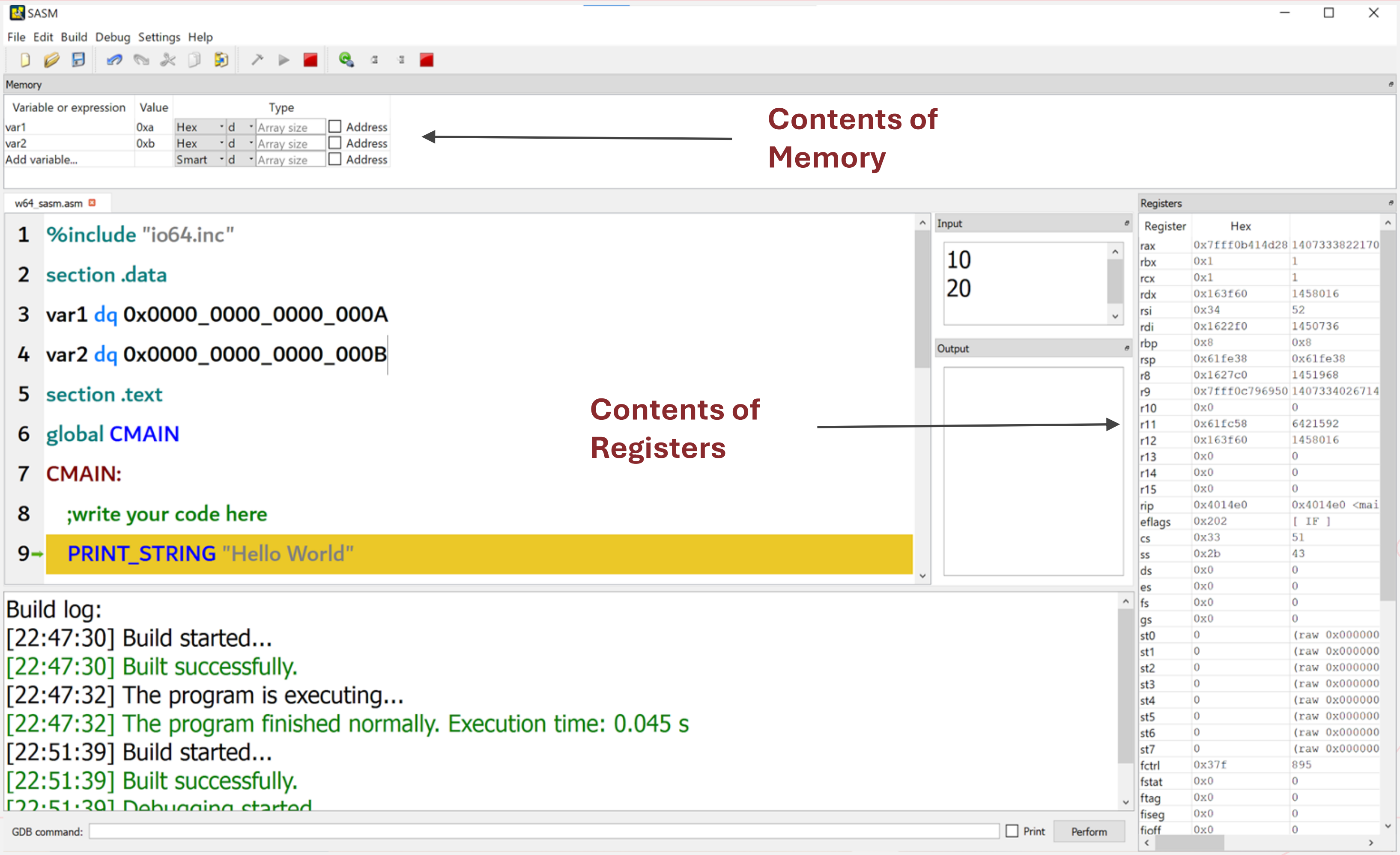
```
10
20
```

Contents of Registers

| Register | Hex | |
|---|---|---|
| rax | 0x7fff0b414d28 | 1407333822170 |
| rbx | 0x1 | 1 |
| rcx | 0x1 | 1 |
| rdx | 0x163f60 | 1458016 |
| rsi | 0x34 | 52 |
| rdi | 0x1622f0 | 1450736 |
| rbp | 0x8 | 0x8 |
| rsp | 0x61fe38 | 0x61fe38 |
| r8 | 0x1627c0 | 1451968 |
| r9 | 0x7fff0c796950 | 1407334026714 |
| r10 | 0x0 | 0 |
| r11 | 0x61fc58 | 6421592 |
| r12 | 0x163f60 | 1458016 |
| r13 | 0x0 | 0 |
| r14 | 0x0 | 0 |
| r15 | 0x0 | 0 |
| rip | 0x4014e0 | 0x4014e0 <mai |
| eflags | 0x202 | [ IF ] |
| cs | 0x33 | 51 |
| ss | 0x2b | 43 |
| ds | 0x0 | 0 |
| es | 0x0 | 0 |
| fs | 0x0 | 0 |
| gs | 0x0 | 0 |
| st0 | 0 | (raw 0x000000 |
| st1 | 0 | (raw 0x000000 |
| st2 | 0 | (raw 0x000000 |
| st3 | 0 | (raw 0x000000 |
| st4 | 0 | (raw 0x000000 |
| st5 | 0 | (raw 0x000000 |
| st6 | 0 | (raw 0x000000 |
| st7 | 0 | (raw 0x000000 |
| fctrl | 0x37f | 895 |
| fstat | 0x0 | 0 |
| ftag | 0x0 | 0 |
| fiseg | 0x0 | 0 |
| fioff | 0x0 | 0 |

```
Build log:
[22:47:30] Build started...
[22:47:30] Built successfully.
[22:47:32] The program is executing...
[22:47:32] The program finished normally. Execution time: 0.045 s
[22:51:39] Build started...
[22:51:39] Built successfully.
[22:51:39] Debugging started
```

# Debug



| Variable or expression | Value | Type | | | |
|---|---|---|---|---|---|
| var1 | 0xa | Hex | d | Array size | □ Address |
| var2 | 0xb | Hex | d | Array size | □ Address |
| Add variable... | | Smart | d | Array size | □ Address |

Contents of Memory

```
%include "io64.inc"
section .data
var1 dq 0x0000_0000_0000_000A
var2 dq 0x0000_0000_0000_000B
section .text
global CMAIN
CMAIN:
    ;write your code here
    PRINT_STRING "Hello World"
```

Input:

```
10
20
```

Contents of Registers

```
Build log:
[22:47:30] Build started...
[22:47:30] Built successfully.
[22:47:32] The program is executing...
[22:47:32] The program finished normally. Execution time: 0.045 s
[22:51:39] Build started...
[22:51:39] Built successfully.
[22:51:39] Debugging started
```

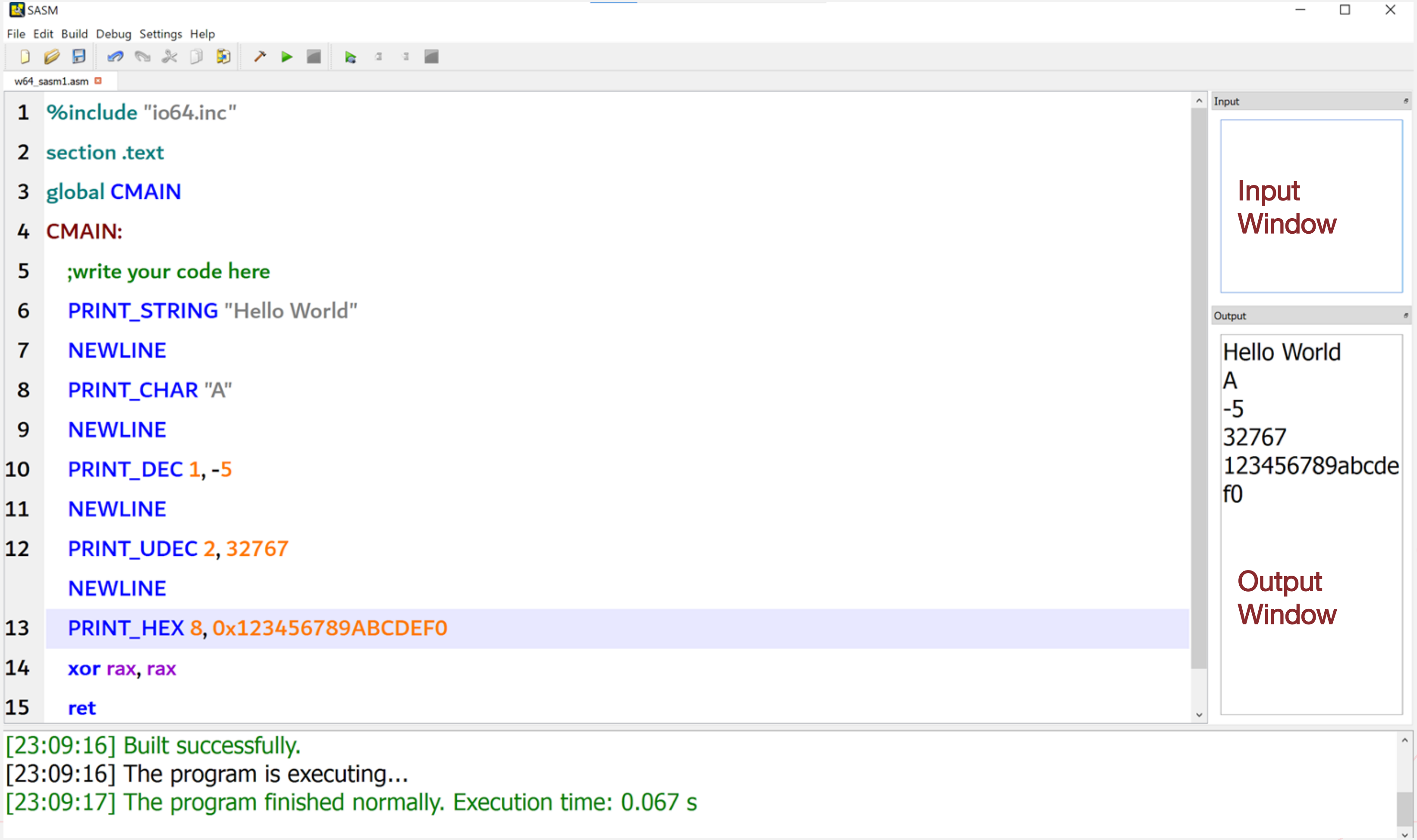
| Register | Hex | |
|---|---|---|
| rax | 0x7fff0b414d28 | 1407333822170 |
| rbx | 0x1 | 1 |
| rcx | 0x1 | 1 |
| rdx | 0x163f60 | 1458016 |
| rsi | 0x34 | 52 |
| rdi | 0x1622f0 | 1450736 |
| rbp | 0x8 | 0x8 |
| rsp | 0x61fe38 | 0x61fe38 |
| r8 | 0x1627c0 | 1451968 |
| r9 | 0x7fff0c796950 | 1407334026714 |
| r10 | 0x0 | 0 |
| r11 | 0x61fc58 | 6421592 |
| r12 | 0x163f60 | 1458016 |
| r13 | 0x0 | 0 |
| r14 | 0x0 | 0 |
| r15 | 0x0 | 0 |
| rip | 0x4014e0 | 0x4014e0 <mai |
| eflags | 0x202 | [ IF ] |
| cs | 0x33 | 51 |
| ss | 0x2b | 43 |
| ds | 0x0 | 0 |
| es | 0x0 | 0 |
| fs | 0x0 | 0 |
| gs | 0x0 | 0 |
| st0 | 0 | (raw 0x000000 |
| st1 | 0 | (raw 0x000000 |
| st2 | 0 | (raw 0x000000 |
| st3 | 0 | (raw 0x000000 |
| st4 | 0 | (raw 0x000000 |
| st5 | 0 | (raw 0x000000 |
| st6 | 0 | (raw 0x000000 |
| st7 | 0 | (raw 0x000000 |
| fctrl | 0x37f | 895 |
| fstat | 0x0 | 0 |
| ftag | 0x0 | 0 |
| fiseg | 0x0 | 0 |
| fioff | 0x0 | 0 |

# SASM using constant I/O macro (output)

| **SASM output macro** | **Description** | **Example** |
|---|---|---|
| PRINT_STRING *string* | Print null-terminated text *string* | PRINT_STRING “Hello” |
| PRINT_CHAR *ch* | Print symbol *ch* | PRINT_CHAR “A” |
| PRINT_DEC size, *data* | Print number *data* in signed decimal representation | PRINT_DEC 2, -5 |
| PRINT_UDEC size, *data* | Print number *data* in unsigned decimal representation | PRINT_UDEC 2, 10000 |
| PRINT_HEX size, *data* | Print number *data* in hexadecimal representation | PRINT_HEX 4, 0x9ABCDEFO |

- **ch, data** - a constant, variable name, register or address
- **string** – string constant, variable name or address
- **size** – the size of data in bytes (1,2,4,8)
- ***Note: The byte size of “1” is not working correctly in SASM.  It is the same as the byte size “2.”***
    - Check: PRINT_DEC 1, 32767 [It printed 32767.  But what do you think should be the correct output?]

# Constant Output using SASM macro



```
%include "io64.inc"
section .text
global CMAIN
CMAIN:
    ;write your code here
    PRINT_STRING "Hello World"
    NEWLINE
    PRINT_CHAR "A"
    NEWLINE
    PRINT_DEC 1, -5
    NEWLINE
    PRINT_UDEC 2, 32767
    NEWLINE
    PRINT_HEX 8, 0x123456789ABCDEF0
    xor rax, rax
    ret
```

Input Window

```
Hello World
A
-5
32767
123456789abcde
f0
```

Output Window

```
[23:09:16] Built successfully.
[23:09:16] The program is executing...
[23:09:17] The program finished normally. Execution time: 0.067 s
```