**Memory Map Practice**

* **Fill in the memory maps and the expressions as they would be just before main ends.**
* **Note that the addresses are written highest address at the top.**
* **Assume that ints are 4 bytes and shorts are 2 bytes**
* **Write all answers in hexadecimal, showing all bytes referenced by the expression (for instance, an int would reference 4 bytes)**

void main( )

{

int iArr [ ] = {0,1,2,3};

int \* iPtr = iArr;

short \*sPtr = (short \*) &iArr[1];

sPtr++;

}

|  |  |  |
| --- | --- | --- |
| Byte’s Address | **Alias** | **Byte’s Value**  \*(sPtr)  \*(iPtr + 1)  \*(sPtr + 3)  (sPtr - 2)  (iPtr + 4)  &iPtr |
| 0x12FF95 |  |  |
| 0x12FF94 |  |  |
| 0x12FF93 |  |  |
| 0x12FF92 | sPtr |  |
| 0x12FF91 |  |  |
| 0x12FF90 |  |  |
| 0x12FF8F |  |  |
| 0x12FF8E | iPtr |  |
| 0x12FF8D |  | 00 |
| 0x12FF8C |  | 00 |
| 0x12FF8B |  | 00 |
| 0x12FF8A |  | 03 |
| 0x12FF89 |  |  |
| 0x12FF88 |  |  |
| 0x12FF87 |  |  |
| 0x12FF86 |  |  |
| 0x12FF85 |  | 00 |
| 0x12FF84 |  | 00 |
| 0x12FF83 |  | 00 |
| 0x12FF82 |  | 01 |
| 0x12FF81 |  |  |
| 0x12FF80 |  |  |
| 0x12FF7F |  |  |
| 0x12FF7E | iArr |  |

#include <math.h>

#pragma pack(1)

typedef struct

{

short Students;

short iRoom;

} stClass;

void main(void)

{

int iNdx;

short int\* sShortPtr = NULL;

stClass staClass[2];

char \*cBytePtr = NULL;

for(iNdx = 0; iNdx < 2; iNdx++)

{

staClass[iNdx].Students = pow(2,iNdx); // pow(2,iNdx) is 2iNdx

staClass[iNdx].iRoom = iNdx ;

}

cBytePtr = (char \*) staClass;

sShortPtr = (short \*) cBytePtr;

}

**Expressions**

\*(cBytePtr) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*(cBytePtr + 10) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*(sShortPtr + 5) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*cBytePtr + 6 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(cBytePtr + 1) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(sShortPtr + 1) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| **Alias** | **Address (in Hex)** | **Value**  **(in Hex)** |
|  | 0x0012FF7F | 0x00 |
|  | 0x0012FF7E | 0x00 |
|  | 0x0012FF7D | 0x00 |
| iNdx | 0x0012FF7C |  |
|  | 0x0012FF7B | 0x00 |
|  | 0x0012FF7A | 0x12 |
|  | 0x0012FF79 | 0xFF |
| sShortPtr | 0x0012FF78 |  |
|  | 0x0012FF77 | 0x00 |
|  | 0x0012FF76 | 0x01 |
|  | 0x0012FF75 |  |
|  | 0x0012FF74 |  |
|  | 0x0012FF73 |  |
|  | 0x0012FF72 |  |
|  | 0x0012FF71 | 0x00 |
| staClass | 0x0012FF70 | 0x01 |
|  | 0x0012FF6F | 0x00 |
|  | 0x0012FF6E | 0x12 |
|  | 0x0012FF6D | 0xFF |
| cBytePtr | 0x0012FF6C | 0x70 |