

## 1. 2016-2

```
001 /* (c) 2016 Rahmat M. Samik-Ibrahim -- This is free software
005  * Assume (&ptrchr is 0x7FFFEEDDCCBB, order of bytes: little-endian) */
009 #define LINES 3
010 #include <stdio.h>
012 void printeq(int lines) {
013     while (lines-- > 0 ) printf("= = ");
014     printf("\n");
015 }
017 void main(void) {
018     int          ii;
019     unsigned char dummy = 'a';
020     unsigned char* ptrchr = &dummy;
022     printeq(LINES);
023     printf(" dummy: %c\n", dummy);
024     printf("*ptrchr: %c\n", *ptrchr);
025     printeq(LINES);
026     printf("%p\n", &ptrchr);
027     printeq(LINES);
028     ptrchr = (char*) &ptrchr;
029     for (ii=0; ii<6; ii++) {
030         printf("%X ", *ptrchr);
031         ptrchr++;
032     }
033     putchar('\n');
034     printeq(LINES);
035 }
```

(a) Write down the output of this program

## 2. 2017-1

### C Programing

<pre>001 /* 002  * (c) 2017 Rahmat M. Samik-Ibrahim 003     -- This is free software 004  * REV00 Thu Mar 30 18:27:30 WIB 2017 005  * START Thu Mar 30 18:27:30 WIB 2017 006  * INT is 32 bit little endian 007  * 41H='A'; 42H='B'; 43H='C'; 44H='D' 008  */ 009 #include &lt;stdio.h&gt; 010 char chrary[]="ZZZZ ZZZZ ";</pre>	<pre>011 void main(void) { 012     char chrvar = 'M'; 013     int  intvar = 0x41424344; 014     int* intptr = (int*) chrary; 015     printf("YY.   chrary=%p\n",  chrary); 016     printf("ZZ.   intptr=%p\n",  intptr); 017     printf("01.   chrvar=%c\n",  chrvar); 018     printf("02.   *chrary=%c\n", *chrary); 019     printf("03. str chrary=%s\n", chrary); 020     *intptr = intvar; 021     printf("04. str chrary=%s\n", chrary); 022 }</pre>
--	---

Program Output (Line: 015, 016, 017, 018, 019, 021):

YY. chrary=0x600a08





## 7. 2019-2 (45%)

```

001 // (c) 2019 This is Free Software
002 // Rahmat M. Samik-Ibrahim 20191021
003 /*
004 These are Clue #1 - Clue #5:
005 =====
006 1:All strings end with 0x00.
007 2:A "string size" includes that 0x00.
008 3:All arrays start with index 0.
009 4:Address=64 bit Little ENDIAN.
010 5:ASCII '0' is 0x30.
011 The program output (lines 27-29):
012 =====
013 1. &string1[0]=0x556677889910
014 2. &string2[0]=0x556677889918
015 3. &stringPtr =0x556677889928
016 */
018 #include <stdio.h>
019 #include <string.h>
020 char string1[]="0123456";
021 char string2[]="0123456";
022 char* stringPtr;
023
024 void main(void) {
025     int size1=sizeof(string1);
026     stringPtr=&string1[size1-1];
027     printf("1. &string1[0]=%p\n", &string1[0]);
028     printf("2. &string2[0]=%p\n", &string2[0]);
029     printf("3. &stringPtr =%p\n", &stringPtr);
030     printf("4. stringPtr =%p\n", stringPtr);
031     *stringPtr = '7';
032     printf("5. STRING: %s\n", &string1[0]);
033 }

```

(a) Program Output (line 30) (46%): .....

(b) Program Output (line 32) (31%): .....

(c) What will be in these following addresses after executing the program (in hexadecimal) (49%)?

Addresses (HEX)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000 5566 7788 991X																
0000 5566 7788 992X																

## 8. 2020-1

```

001 // (c) 2020 This is Free Software
002 // Rahmat M. Samik-Ibrahim 2020
003 // R03 0310Tue1715
004 /*
005 This Clue #1 - Clue #5:
006 =====
007 1: All strings end with 0x00.
008 2: All arrays start with index 0.
009 3: Address=64 bit Little ENDIAN.
010 4: ASCII '0' is 0x30.
011 5: ASCII 'A' is 0x41.
012 The first 3 lines of program output:
013 =====
014 1. 0X0000556677665520
015 2. 0X0000556677889918
016 3. 0X0000556677889910
017 */
019 #include <stdio.h>
020 #include <string.h>
021 typedef unsigned long UL;
022 char* stringptr="0123456";
023 char string1[]="89ABCDE";
024
025 void main(void) {
026     printf("1. %#16.16lX\n", (UL) stringptr);
027     printf("2. %#16.16lX\n", (UL) &stringptr);
028     printf("3. %#16.16lX\n", (UL) &string1[0]);
029
030     printf("4. %#16.16lX\n", (UL) &string1[6]);
031     printf("5. %#X %c\n",string1[6], string1[6]);
032     printf("6. %#X %c\n",*stringptr, *stringptr);
033     stringptr++;
034     printf("7. %#16.16lX\n", (UL) stringptr);
035     printf("8. %#X %c\n",*stringptr, *stringptr);
036 }

```

Program Output:

