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

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

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

```
YY. chrary=0x600a08
```

3. **2017-2**

C Programing ADDR

```
001 /*                                        014 void main (void) {
002  * (c) 2017 Rahmat M. Samik-Ibrahim        015     stringPTR=stringChar;
003  * http://rahmatm.samik-ibrahim.vlsm.org/  016     printf ("ADDR1: %p VAL: %p STR: %s\n", &stringChar,
004  * This is free software.                                                  stringChar, stringChar);
005  * REV00 Mon Oct 16 21:15:03 WIB 2017      017     printf ("ADDR2: %p VAL: %p STR: %s\n", &stringPTR,
006  * START Mon Oct 16 21:15:03 WIB 2017                                      stringPTR, stringPTR);
007  */                                        018     while (*(++stringPTR) != 0 ) {
008                                            019         printf ("ADDR3: %p VAL: %p STR: %s\n", &stringPTR,
009 #include <stdio.h>                                                         stringPTR, stringPTR);
010                                            020     }
011 char* stringChar="HALLO";                 021     printf ("End of String = %p\n", stringPTR);
012 char* stringPTR;                          022 }
```

Program Output:

```
ADDR1:   0x601038 VAL: 0x400674 STR: HALLO
ADDR2:   0x601048 VAL: 0x400674 STR: HALLO
```

4. **2018-1**

What is the output of this following program:

```
001 /* (c) 2018 This is a free program  */    018 /* Clue#1: All strings end with 0x00 */
002 /* Rahmat M. Samik-Ibrahim          */    019 /* Clue#2: Address=64 bit BIG ENDIAN */
003 /* The "array"   starts  at 0x601040 */    020 /* Clue#3: ASCII '0' (Zero) is 0x30   */
004 /* The "pointer" address is 0x601050 */    021 /* Clue#4: ASCII 'A'        is 0x41   */
005
006 #include <stdio.h>
007                                            _____
008 char  array[]="0123456789ABCDE";
009 char* pointer=array;                       _____
010 void  main(void) {
011     printf("START\n");                     _____
012     printf("%p\n", &pointer);
013     printf("%p\n",  pointer);              _____
014     printf("%s\n",  pointer);
015     printf("%d\n",  pointer[15]);          _____
016     printf("STOP\n");
017 }                                          _____
```

Initially, addresses 0x601040 - 0x60105F = 0x00. What will be in those addresses after executing the program?

Addresses (HEX)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000 0000 0060 1040																
0000 0000 0060 1050																

5. **2018-2 (44%)**

What is the output of this following program:

```
001 /* (c) 2018 This is a free program   */      019 /* This Clue #1 - Clue #5            */
002 /* Rahmat M. Samik-Ibrahim           */      020
003                                               021 /* 1: All strings end with 0x00     */
004 #include <stdio.h>                            022 /* 2: Address=64 bit Little ENDIAN */
005                                               023 /* 3: ASCII '0' (Zero)  is 0x30     */
006 unsigned char  string[]="0123456789ABCDE";    024 /* 4: ASCII 'A'         is 0x41     */
007 unsigned long  longst=(unsigned long) string; 025 /* 5: Long Integer Size = 64bit    */
008 unsigned char* achar;                         026
009
010 void main(void) {                             PROGRAM OUTPUT
011    achar=(unsigned char*) &longst;
012    printf("(0) %p\n",  string);               (0) 0x601040
013    printf("(1) %p\n", &longst);               (1) 0x601050
014    printf("(2) %p\n",  &achar);               (2) 0x601060
015    printf("(3) %p\n",   achar);               (3) 0x601050
016    printf("(4) %#X\n", *achar);               (4) 0X40
017    printf("(5) %#16.16lx\n",longst);          (5) 0x0000000000601040
018 }
```

Initially, addresses 0x601040 - 0x60106F = 0x00. What will be in those addresses after executing the program?

Addresses (HEX)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000 0000 0060 1040																
0000 0000 0060 1050																
0000 0000 0060 1060																

6. **2019-1 (53.9%)**

```
Clue #1 - Clue #5:                     001 // (c) 2019 This is Free Software
1: All strings end with 0x00.          002 // Rahmat M. Samik-Ibrahim 20190315-1352 R01
2: Address=64 bit Little ENDIAN.       003
3: ASCII A is 0x41.                    004 #include <stdio.h>
4: ASCII a is 0x61.                    005 char  ch_array[]="abcdefg";
5: Initially Addresses:                006 char* ch_pointer="ABCDEFG";
   0x0000 5566 7788 9900 - 991F = 0x00. 007

Program Output:                        008 void main(void) {
 &ch_array[0]=0x556677889910           009    printf(" &ch_array[0]=%p\n", &ch_array[0]);
   ch_array[0]=a                       010    printf("  ch_array[0]=%c\n",  ch_array[0]);
                                        011    ch_pointer = ch_pointer + 5;
 &ch_pointer =0x556677889918           012    printf(" &ch_pointer =%p\n", &ch_pointer);
   ch_pointer =0x556677889909          013    printf("  ch_pointer =%p\n",  ch_pointer);
 *ch_pointer =F                        014    printf("  *ch_pointer =%c\n", *ch_pointer);
                                        015 }
```

What will be in these following addresses after executing the program (in **hexadecimal**)?

Addresses (HEX)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000 5566 7788 990X																
0000 5566 7788 991X																