

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 <stdio.h> 010 char chrarry[]="ZZZZ ZZZZ "; </pre>	<pre> 011 void main(void) { 012 char chrvar = 'M'; 013 int intvar = 0x41424344; 014 int* intptr = (int*) chrarry; 015 printf("YY. chrarry=%p\n", chrarry); 016 printf("ZZ. intptr=%p\n", intptr); 017 printf("01. chrvar=%c\n", chrvar); 018 printf("02. *chrarry=%c\n", *chrarry); 019 printf("03. str chrarry=%s\n", chrarry); 020 *intptr = intvar; 021 printf("04. str chrarry=%s\n", chrarry); 022 } </pre>
Program Output (Line: 015, 016, 017, 018, 019, 021):	
YY. chrarry=0x600a08	

3. 2017-2

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

<pre> 001 /* (c) 2018 This is a free program */ 002 /* Rahmat M. Samik-Ibrahim */ 003 /* The "array" starts at 0x601040 */ 004 /* The "pointer" address is 0x601050 */ 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 } </pre>	<pre> 018 /* Clue#1: All strings end with 0x00 */ 019 /* Clue#2: Address=64 bit BIG ENDIAN */ 020 /* Clue#3: ASCII '0' (Zero) is 0x30 */ 021 /* Clue#4: ASCII 'A' is 0x41 */ </pre> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
---	---

[illegible]

- (0) 0x601040
- (1) 0x601050
- (2) 0x601060
- (3) 0x601050
- (4) 0X40
- (5) 0x0000000000601040

[illegible]

[illegible]

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:

(a) (line 30) -----

(b) (line 31) -----

(c) (line 32) -----

(d) (line 34) -----

(e) (line 35) -----

(f) **INITIALLY**, addresses 0x5566 7766 5520 - 0x5566 7766 552F, and 0x5566 7788 9910 - 0x5566 7788 991F = 0; What will be in those 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 7766 552X																
0000 5566 7788 991X																

9. 2022-2 (47.3%)

Program "mymemory3.c" a shortened version of the "mymemory2.c" program in the last **WEEK 05** assignment. See next page for output program "mymemory3" and source code "mymemory3.c" (Line numbers are added).

(a) (45%) Based on the output of the "mymemory3" program, what is the "**Total usable main memory size**" of that system? Answer:

Total Memory: ----- MB.

```

##### OUTPUT + LINES of mymemory3 #####
01 ZCZC RAM 71 MB
02 ZCZC FREE 26 MB
03 ZCZC BUFFER 1 MB
04 ZCZC SWAP 975 MB
05 ZCZC FREESW 940 MB
06 ZCZC FREE1 26 MB
07 ZCZC FREESW1 940 MB
08 ZCZC FREE2 7 MB
09 ZCZC FREESW2 442 MB
10 ZCZC ADDR 01 0X000055E3A0079155 printMyAddress()
11 ZCZC ADDR 02 0X000055E3A0079192 main()
12 ZCZC ADDR 03 0X000055E3A007C040 &pcounter
13 ZCZC ADDR 04 0X00007FCC6D324010 intArray
14 ZCZC ADDR 05 0X00007FCC8D378CF0 printf()
15 ZCZC ADDR 06 0X00007FFFCCEC2F260 &intArray
16 ZCZC ADDR 07 0X00007FFFCCEC2F26C &localdummy
17 ZCZC ADDR 08 0X00007FFFCCEC2F270 &guestInfo

### mymemory3.c #####
001 // Copyright (C) 2022 C. BinKadal
002 // This program is free script/software.
003 // REV01: Tue 11 Oct 2022 19:00
004 // START: Tue 11 Oct 2022 18:00
005
006 #include <stdio.h>
007 #include <stdlib.h>
008 #include <unistd.h>
009 #include <sys/sysinfo.h>
010
011 typedef char* String;
012 typedef int* IntPtr;
013 typedef unsigned long UL;
014 typedef void* AnyAddrPtr;
015 typedef struct sysinfo SYSINFO;

017 int pcounter=1;
018 void printMyAddress (AnyAddrPtr address, String message) {
019     printf("ZCZC ADDR %2.2d %#16.16IX %s\n",
020         pcounter++, (UL) address, message);
021 }

023 #define ArraySize 128*1024*1024
024 int main(void) {
025     SYSINFO guestInfo;
026     int localdummy=0;

    sysinfo(&guestInfo);
029     printf("ZCZC RAM %5lu MB\n", guestInfo.totalram/1024/1024);
030     printf("ZCZC FREE %5lu MB\n", guestInfo.freeram/1024/1024);
031     printf("ZCZC BUFFER %5lu MB\n", guestInfo.bufferram/1024/1024);
032     printf("ZCZC SWAP %5lu MB\n", guestInfo.totalswap/1024/1024);
033     printf("ZCZC FREESW %5lu MB\n", guestInfo.freeswap/1024/1024);
034     IntPtr intArray=malloc((ArraySize+1) * sizeof(int));
035     sysinfo(&guestInfo);
036     printf("ZCZC FREE1 %5lu MB\n", guestInfo.freeram/1024/1024);
037     printf("ZCZC FREESW1 %5lu MB\n", guestInfo.freeswap/1024/1024);
038     for (int ii=0; ii<ArraySize; ii++) intArray[ii]=255;
039     sysinfo(&guestInfo);
040     printf("ZCZC FREE2 %5lu MB\n", guestInfo.freeram/1024/1024);
041     printf("ZCZC FREESW2 %5lu MB\n", guestInfo.freeswap/1024/1024);
042     printMyAddress( printMyAddress, "printMyAddress()");
043     printMyAddress( main, "main()");
044     printMyAddress(&pcounter, "&pcounter");
045     printMyAddress( intArray, "intArray");
046     printMyAddress( printf, "printf()");
047     printMyAddress(&intArray, "&intArray");
048     printMyAddress(&localdummy, "&localdummy");
049     printMyAddress(&guestInfo, "&guestInfo");
050     sleep(1);
051 }

```

[illegible]

10. 2023-2 (CB:46%)

```

01 // Copyright (C) 2023 BinKadal,Sdn.Bhd.
02 // This program is free script/software.
03 // This program is distributed in the
04 // hope that it will be useful, but
05 // WITHOUT ANY WARRANTY;without even
06 // implied warranty of MERCHANTABILITY
07 /* or FITNESS FOR A PARTICULAR PURPOSE.
08 # INFO: UTS 2023-2 (midterm) */
09 // REV01: Mon 09 Oct 2023 19:00
11 // START: Mon 09 Oct 2023 17:00
12
13 /* This Clue #1 - Clue #6:
14 =====
17 1: All strings end with 0x00.
18 2: All arrays start with index 0.
19 3: Address=64 bit Little ENDIAN.
20 4: ASCII '0' is 0x30.
21 5: ASCII 'A' is 0x41.
22 6: ASCII 'a' is 0x61.
23 */
24
25 #include <stdio.h>
26 #include <string.h>
27 typedef unsigned long UL;
28 typedef char*      STRING;
29 char  array[]="0123456789ABCDE";
30 STRING dummy="ZZZZZZZZZZZ";
31 STRING string="abcdefghijklmno";
32
33 void main(void) {
34     printf("X1. %#16.16lX\n", (UL)  dummy);
35     printf("X2. %#16.16lX\n", (UL)  string);
36     printf("\n");
37     printf("X3. %#16.16lX\n", (UL) &array[0]);
38     printf("X4. %#16.16lX\n", (UL) &dummy);
39     printf("X5. %#16.16lX\n", (UL) &string);
40     printf("\n");
41     printf("01. %s\n",                string);
42     printf("02. %c\n",                *string);
43     string+=2;
44     printf("03. %#16.16lX\n", (UL)  string);
45     printf("04. %c\n",                *string);
46     printf("\n");
47     printf("05. %s\n",                array);
48     string=&array[2];
49     printf("06. %c\n",                *string);
50     printf("07. %#16.16lX\n", (UL)  string);
51     string+=2;
52     printf("08. %c\n",                *string);
53     printf("09. %#16.16lX\n", (UL)  string);
54 }

```

Program Output:

X1. 0X0000556677889904
X2. 0X0000556677889910

X3. 0X000055667788B010
X4. 0X000055667788B020
X5. 0X000055667788B028

(a) (73% line 41) _____

(b) (27% line 42) _____

(c) (91% line 44) _____

(d) (36% line 45) _____

(e) (82% line 47) _____

[illegible]