

04 ADR-PTR - Addressing & Pointers Vaulsworg Of Adr. Par. - Addressing & Foliners Aneka Soal Ujian Sistem Operasi Rahmat M. Samik-Ibrahim et.al. Rahmat M. Samik-Ibrahim et.al.

© 2016 - 2021 — Rev: 39 - 15-Sep-2021. URL: https://rms46.vlsm.org/2/199.pdf. Kumpulan soal ujian lainnya dapat diakses melalui URL: https://os.vlsm.org/. Silakan mengubah, memperbanyak, serta mendistribusikan dokumen ini selama tidak menghapus ketentuan ini!

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 printeg(int lines) {
       while (lines-- > 0 ) printf("= = ");
013
       printf("\n");
014
015 }
017 void main(void) {
018
       int
                        ii;
                       dummy = 'a';
019
       unsigned char
020
       unsigned char* ptrchr = &dummy;
022
       printeq(LINES);
023
       printf(" dummy: %c\n", dummy);
       printf("*ptrchr: %c\n", *ptrchr);
024
025
       printeq(LINES);
026
       printf("%p\n", &ptrchr);
027
       printeq(LINES);
028
       ptrchr = (char*) &ptrchr;
029
       for (ii=0; ii<6; ii++) {
          printf("%X ", *ptrchr);
030
031
          ptrchr++;
       }
032
033
       putchar('\n');
034
       printeq(LINES);
035 }
```

(a) Write down the output of this program

2. **2017-1**

```
C Programing
                                                  011 void main(void) {
001 /*
002 * (c) 2017 Rahmat M. Samik-Ibrahim
                                                  012 char chrvar = 'M';
          -- This is free software
                                                  013
                                                        int intvar = 0x41424344;
                                                        int* intptr = (int*) chrary;
003 * REV00 Thu Mar 30 18:27:30 WIB 2017
                                                  014
004 * START Thu Mar 30 18:27:30 WIB 2017
                                                  015
                                                        printf("YY. chrary=%p\n", chrary);
005 * INT is 32 bit little endian
                                                                     intptr=%p\n", intptr);
                                                  016
                                                        printf("ZZ.
006 * 41H='A'; 42H='B'; 43H='C"; 44H='D'
                                                  017
                                                        printf("01. chrvar=%c\n", chrvar);
                                                        printf("02. *chrary=%c\n", *chrary);
007 */
                                                  018
800
                                                  019
                                                        printf("03. str chrary=%s\n", chrary);
                                                  020
009 #include <stdio.h>
                                                        *intptr = intvar;
                                                        printf("04. str chrary=%s\n", chrary);
010 char chrary[]="ZZZZ ZZZZ ";
                                                  021
                                                  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 ) {
800
                                                 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);
                                                 022 }
012 char* stringPTR;
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) {
      printf("START\n");
011
012
      printf("%p\n", &pointer);
013
      printf("%p\n", pointer);
      printf("%s\n", pointer);
014
015
      printf("%d\n", pointer[15]);
      printf("STOP\n");
016
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	В	С	D	Ε	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
       achar=(unsigned char*) &longst;
011
012
      printf("(0) %p\n", string);
                                                     (0) 0x601040
      printf("(1) %p\n", &longst);
013
                                                     (1) 0x601050
      printf("(2) %p\n", &achar);
014
                                                     (2) 0x601060
      printf("(3) \%p\n",
015
                            achar);
                                                     (3) 0x601050
      printf("(4) %#X\n", *achar);
016
                                                     (4) 0X40
017
      printf("(5) %#16.16lx\n",longst);
                                                     (5) 0x000000000601040
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	В	С	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
                                          002 // Rahmat M. Samik-Ibrahim 20190315-1352 R01
1: All strings end with 0x00.
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
                                          008 void main(void) {
Program Output:
                                                 printf(" &ch_array[0]=%p\n", &ch_array[0]);
 &ch_array[0]=0x556677889910
                                          009
  ch_array[0]=a
                                          010
                                                 printf(" ch_array[0]=%c\n", ch_array[0]);
                                                 ch_pointer = ch_pointer + 5;
                                          011
 &ch_pointer =0x556677889918
                                          012
                                                 printf(" &ch_pointer =%p\n", &ch_pointer);
                                                 printf(" ch_pointer =%p\n", ch_pointer);
  ch_pointer =0x556677889909
                                          013
                                                 printf(" *ch_pointer =%c\n", *ch_pointer);
 *ch_pointer =F
                                          014
                                          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	В	С	D	E	F
0000 5566 7788 990X																
0000 5566 7788 991X																

```
7. 2019-2 (45%)
```

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

- (a) Program Output (line 30) (46%): ______
- (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	В	С	D	Ε	F
0000 5566 7788 991X																
0000 5566 7788 992X																

8. **2020-1**

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

Aneka Sc	al Hiian	Sistem	Operaci	
Aneka 50	oai Ujian	Sistem	Operasi	

04 ADR-PTR

Halaman 5 dari 5

(a)	(line 30)	
(b)	(line 31)	
(c)	(line 32)	
(d)	(line 34)	
(e)	(line 35)	

(f) **INITIALLY**, addresses 0x556677665520 - 0x55667766552F, and 0x556677889910 - 0x55667788991F = 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	В	С	D	E	F
0000 5566 7766 552X																
0000 5566 7788 991X																